Recherche par image robuste par apprentissage profond

Elias Ramzi

HESAM UNIVERSITÉ

le cnam

ÉCOLE DOCTORALE Sciences des Métiers de l'Ingénieur
Centre d'études et de recherche en informatique et communications

THÈSE

présentée par : **Elias RAMZI**
soutenue le : **20 mars 2024**

pour obtenir le grade de : **Docteur d'HESAM Université**

préparée au : **Conservatoire national des arts et métiers**
Discipline : **Mathématiques, Informatique et Systèmes**
Spécialité : **Informatique**

Robust image retrieval with deep learning

THÈSE dirigée par :
M. THOME Nicolas Professeur des universités, Sorbonne Université

et co-encadrée par :
M. AUDEBERT Nicolas Maître de conférences, Cnam
M. RAMBOUR Clément Maître de conférences, Cnam
M. BITOT Xavier Project leader, Coexya

Jury

M. Michel CRUCIANU	Professeur des universités, Cnam	Président
Mme. Diane LARLUS	Principal Research Scientist, Naver Labs Europe	Rapporteuse
M. Yannis AVRITHIS	Principal Investigator, IARAI	Rapporteur
M. Matthieu CORD	Professeur des universités, Sorbonne Université	Examinateur
M. Hervé JÉGOU	Chief Scientific Officer, Kyutai	Examinateur
M. Nicolas AUDEBERT	Maître de conférences, Cnam	Co-encadrant
M. Clément RAMBOUR	Maître de conférences, Cnam	Co-encadrant
M. Nicolas THOME	Professeur, Sorbonne Université	Directeur de thèse
M. Xavier BITOT	Project leader, Coexya	Invité

THÈSE

Remerciements

Je tenais à remercier mon jury de thèse. En commençant par Diane et Yannis qui m'ont suivi au long de ma thèse et qui ont ensuite accepté d'être rapporteureuse pour mon manuscrit. Vos conseils m'ont toujours aidé, et j'ai eu plaisir à échanger avec vous. Merci Michel, Matthieu et Hervé d'avoir accepté d'être examinateurs et de faire le déplacement pour ma thèse. J'ai apprécié vos questions et nos échanges. Je tiens à remercier grandement mes encadrants de thèse. Merci à Nicolas Thome, de m'avoir d'abord accepté en stage sur un coup de téléphone durant la période de covid. Puis de m'avoir fait confiance pour continuer en thèse. J'ai beaucoup apprécié travailler avec toi, pour tes conseils, tes idées, et la façon dont tu t'es beaucoup impliqué dans mes travaux. Merci Nicolas Audebert, de m'avoir suivi tout au long de cette thèse et de m'avoir aidé en t'impliquant autant dans mes projets. J'ai beaucoup apprécié nos échanges et tes suggestions toujours pertinentes. Merci Clément, j'ai également beaucoup apprécié travailler avec toi, merci pour l'aide précieuse que tu as apporté lors de mes différents projets. Enfin, merci Xavier de m'avoir fait confiance pour effectuer ma thèse avec Coexya. J'ai apprécié nos échanges, avec notamment tes nombreuses questions et ta volonté de comprendre mon travail. Merci de m'avoir laissé la liberté de mener ma recherche académique comme je le souhaitais. Merci beaucoup à vous pour m'avoir autant aider pour cette thèse. Je tenais ensuite à remercier mes collègues doctorants du Cnam. En commençant par Loïc avec qui j'ai partagé plus de trois ans en stage d'abord et dans un bureau ensuite. Merci à Marc, avec qui j'ai collaboré pendant plus d'un an. Merci à Yannis avec qui j'ai également collaboré. Merci à Perla et à tous mes autres collègues et co-doctorants George, Laura, Charles, Armand, Léo, Wafa, et les nouveaux, avec qui j'ai partagé de nombreux bons moments, et qui ont participé à rendre cette d'autant meilleure. Merci également à mes collègues de Coexya, Mostefa, Hugo, Narek, Fadil, Hemza, avec qui j'ai eu de nombreux échanges et partagé de bons moments. Je vous souhaite une très bonne continuation. Merci à mes amis qui ont été là pendant cette longue épreuve. En particulier merci à Nicolas et Alex avec qui j'ai pu échanger sur les bons et aussi les difficiles moments de ma thèse. Merci à Élodie de m'avoir soutenu durant ma thèse. Merci à mes amis de la Septette, du weekend, mes amis de Supélec ainsi qu'aux autres qui ne s'identifient pas

dans ces catégories. Enfin, je tenais à remercier grandement ma famille, sans qui je n'aurais surement pas pu en arriver là. Merci Zaccharie de m'avoir soutenu moralement durant ma thèse et de m'avoir autant aidé. Merci à Maxime, qui a voulu faire comme ses deux grands frères. Et enfin un grand merci à mes parents pour leur soutien tout au long de ma longue scolarité, merci de m'avoir poussé et aider autant.

Encore une fois un grand merci à tout le monde.

Résumé

Cette thèse aborde la problématique de la recherche robuste d'images par apprentissage profond. La recherche par le contenu d'images consiste à trouver des images visuellement similaires à une image "requête" dans de grandes bases de données. Les approches par apprentissage profond sont basées sur l'apprentissage de représentations des images afin de mesurer leur similarité, par exemple, avec la distance euclidienne. La recherche d'images est notamment utilisée dans les moteurs de recherche, tels qu'Acsepto, le moteur de recherche de logos de marques déposées développé par Coexya. Cette thèse vise à améliorer les performances et la fiabilité des systèmes de recherche d'images. À cette fin, nous explorons la robustesse dans l'apprentissage profond selon trois perspectives.

Nous exposons d'abord les difficultés qui se présentent lors de l'optimisation des métriques d'évaluation utilisées en recherche d'images, telles que la Précision Moyenne (AP) et le rappel à k, à savoir la non-différentiabilité et la non-décomposabilité. Elles rendent ces métriques difficilement optimisables par descente de gradient stochastique. Il est ainsi nécessaire d'utiliser des fonctions de coût de substitution pour entraîner les réseaux de neurones profonds (DNN), ce qui induit une disparité entre l'objectif d'entraînement et les métriques d'évaluation. Pour réduire cet écart, nous introduisons une famille de fonctions de coût différentiables qui sont des bornes supérieures des métriques d'évaluation usuelles et incluent un objectif explicite de décomposabilité. Cette famille permet d'optimiser plusieurs métriques d'évaluation, telles que l'AP, le rappel à k et le NDCG. Cette approche, appelée ROADMAP, surpasse les fonctions de coût de l'état de l'art sur plusieurs bases de données de recherche d'images.

Ensuite, nous cherchons à réduire la gravité des erreurs commises par les systèmes de recherche d'images basés apprentissage profonds. En effet, les réseaux de neurones, lorsqu'ils ne sont pas contraints, ont tendance à commettre des erreurs sévères, qui sont difficilement compréhensibles par les humains. Ces erreurs graves peuvent réduire la confiance des utilisateurs dans les moteurs de recherche. Nous proposons une solution en exploitant les relations hiérarchiques entre les catégories d'images. En effet, les relations sémantiques peuvent servir de proxy pour la façon dont les humains jugent la gravité d'une erreur. Ces relations sont

intégrées dans une nouvelle extension de l'AP au cadre hiérarchique, $\mathcal{H}$-AP. Nous définissons ensuite HAPPIER, une fonction de coût différentiable optimisant $\mathcal{H}$-AP, construite similairement à ROADMAP. Nous montrons quantitativement et qualitativement que les réseaux de neurones entraînés avec HAPPIER produisent des classements avec des erreurs moins sévères et se rapprochent davantage de la sémantique des ensembles de données.

Enfin, nous abordons les capacités de détection d'exemples hors distribution (OOD) des DNN. Il s'agit de détecter des données qui ne devraient pas être traitées par les DNN, par exemple, des images de catégories qui n'ont pas été vues pendant l'entraînement. Nous introduisons HEAT, une nouvelle méthode de détection d'OOD. HEAT est une méthode post-hoc, ce qui la rend applicable à potentiellement toutes les architectures pré-entraînées, sans nécessité de les affiner. Nous proposons d'utiliser les modèles à énergie pour raffiner les méthodes de la littérature, en apprenant un terme résiduel pour améliorer leur expressivité. Nous exploitons ensuite leurs différents biais de modélisation complémentaires en utilisant la composition de fonctions d'énergies pour améliorer les capacités de détection d'OOD des DNN. Nous démontrons quantitativement l'intérêt de ces deux composantes sur trois jeux de données, pour lesquels HEAT surpasse les méthodes de l'état de l'art en détection d'OOD.

Mots-clés : Apprentissage profond, Vision par ordinateur, Recherche par image, Robustesse.

Abstract

This thesis tackles robust image retrieval with deep learning. Image retrieval consists in querying large databases to find images visually similar to a query image. Deep learning approaches involve learning representations to measure how are similar images are, *e.g.* using the Euclidean distance. Image retrieval is notably used in search engines, for instance trademarked logo retrieval with Coexya's Acsepto. The motivation of this thesis is to improve image retrieval systems performances and reliability. To this end, we explore in this thesis the robustness of deep learning from three perspectives.

We first expose the shortcomings that arise when optimizing for the evaluation metrics typically used in image retrieval, *e.g.* Average Precision (AP) and recall at k, namely non-differentiability and non-decomposability. These shortcomings make these metrics not directly amenable to stochastic gradient descent. This forces the use of surrogate losses to train deep neural networks (DNNs), which leads to a discrepancy between the training objective and the evaluation metrics of image retrieval systems. To reduce this gap, we introduce a family of differentiable rank-based losses that are upper bounds of the evaluation metrics and include an objective to explicitly reduce non-decomposability. We show that this framework works for several evaluation metrics, *e.g.* AP, recall and NDCG. Using ROADMAP to optimize AP compares favorably to other state-of-the-art surrogate losses on several image retrieval benchmarks.

Then, we aim to reduce the severity of the mistakes from deep image retrieval systems. Indeed, DNNs, when not controlled, tend to make severe mistakes that do not align well with human understanding. These severe errors can reduce the trust of users in search engines. We address this issue by leveraging hierarchical relations between categories. Indeed, hierarchical relations can serve as a proxy for how humans would judge the severity of a mistake. These relations are integrated in a novel extension of the AP to the hierarchical setting, $\mathcal{H}$-AP. From this metric we derive HAPPIER, a differentiable surrogate to $\mathcal{H}$-AP built upon the ROADMAP framework. We show quantitatively and qualitatively that DNNs trained with HAPPIER produce ranking with less severe mistakes and closer align to the semantics of the

datasets.

Finally, we address the out-of-distribution (OOD) detection capabilities of DNNs. It consists in detecting inputs that should not be processed by DNNs, *e.g.* images from categories that were not seen during training. We introduce HEAT, a new OOD detection method. HEAT is a post-hoc method, which makes it applicable to virtually any pre-trained backbones, without the need to fine-tune them. We propose to use the principled energy-based model framework to correct methods from the literature, by learning a residual term to improve their expressiveness. We then leverage their different modeling biases using energy function composition to improve OOD detection capabilities of DNNs. We show in experiments the interest of the two components of HEAT. Furthermore, we show that HEAT outperforms state-of-the-art OOD detection methods.

Keywords: Deep learning, Computer vision, Image retrieval, Robustness.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Content

1.1 AI summer.

In recent years, the field of artificial intelligence (AI) has witnessed a remarkable transformation, propelled by the renewal of deep learning (DL) [1]. It has revolutionized numerous field. Generative AI has found tremendous success with Large Language Models (LLM), such as ChatGPT 3 & 4 [2], [3] (Fig. 1.1f), Llama 1 & 2 [4], [5], and image generation, *e.g.* text-to-image generation with Stable Diffusion [6] (Fig. 1.1e) or DALL-E [7]. It has changed how we represent multimedia content, such as images with DINOv2 [8], MAE [9], SAM [10] (Fig. 1.1c), audio with Whisper [11] (Fig. 1.1b), and more recently multi-modal data with CLIP [12] (Fig. 1.1a), ImageBind [13] and Gemini [14]. Its application in reinforcement learning has allowed to master the game of Go with AlphaGo [15] subsequently defeating its world champion (Fig. 1.1d), and has allowed in robotics to beat racing drones champions [16], or for autonomous transportation such as driving [17]. Its application are diverse, and it was successfully applied to physics research, *e.g.* weather forecasting with GraphCast [18] or the stabilization of plasma in nuclear fusion [19]. It has had a tremendous impact on biology with the release of AlphaFold [20], and is at the core of ventures such as Altos lab that uses AI to rejuvenate cells with the Shinya Yamanaka reaction [21]. Another prolific area is the application of AI to medicine, *e.g.* faster IRM reconstruction [22], CT scan organ segmentation [23], or drug discovery [24].

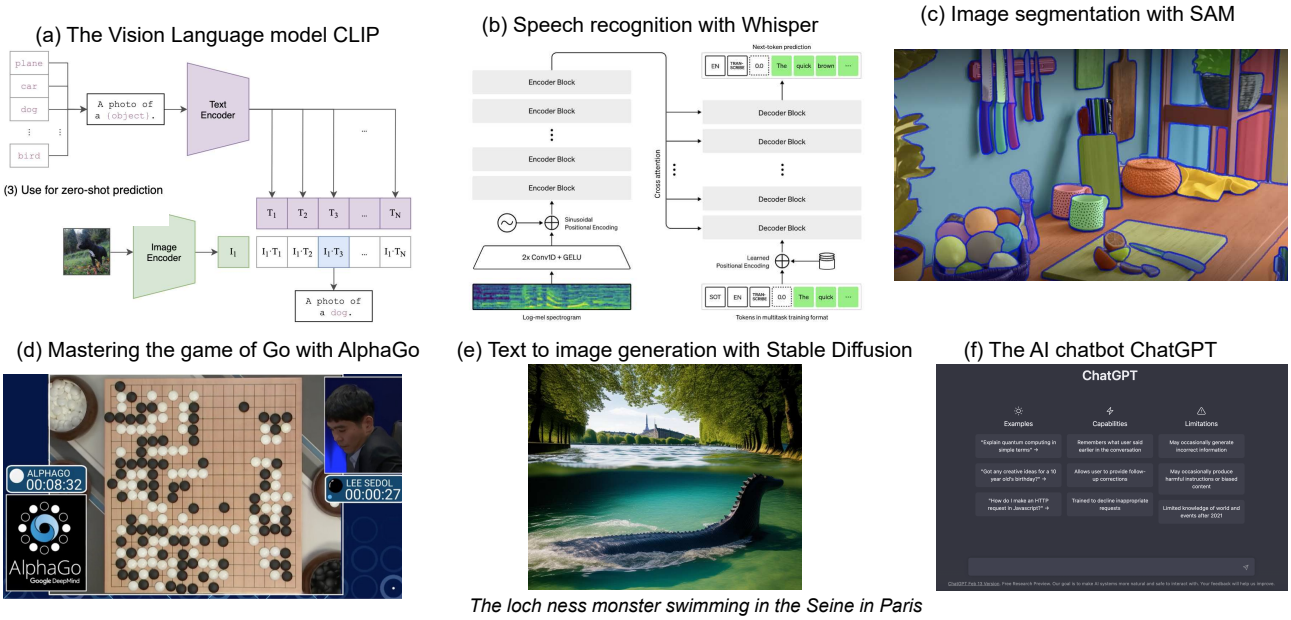


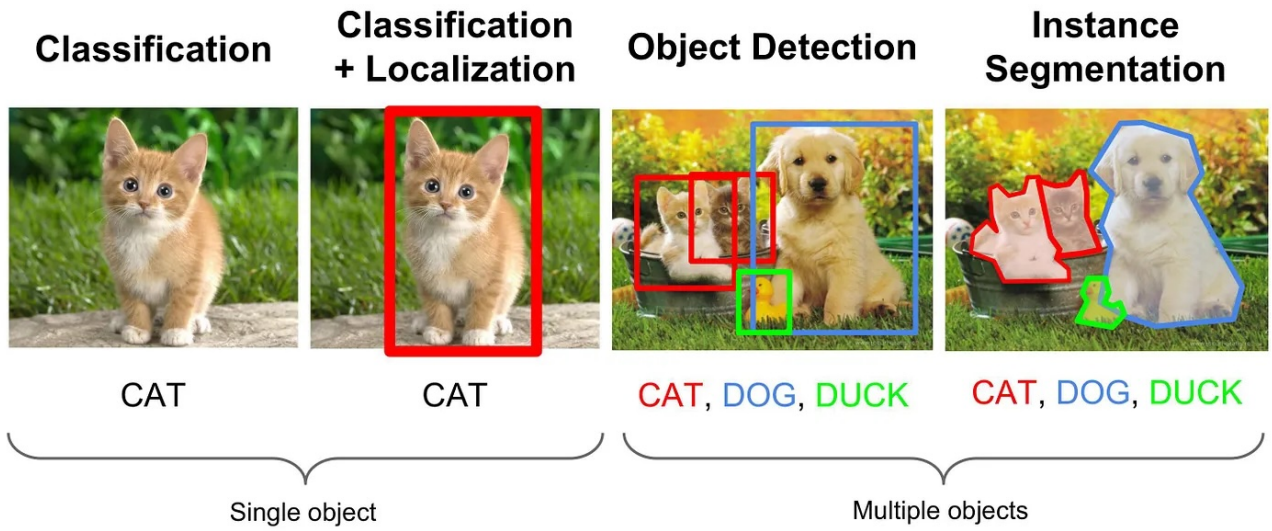
Figure 1.1: Examples of deep learning research with high impact on the AI community and on a general audience. ChatGPT (f) has had 1.7 *billion* visit in October 2023*.

* source: https://explodingtopics.com/blog/chatgpt-users

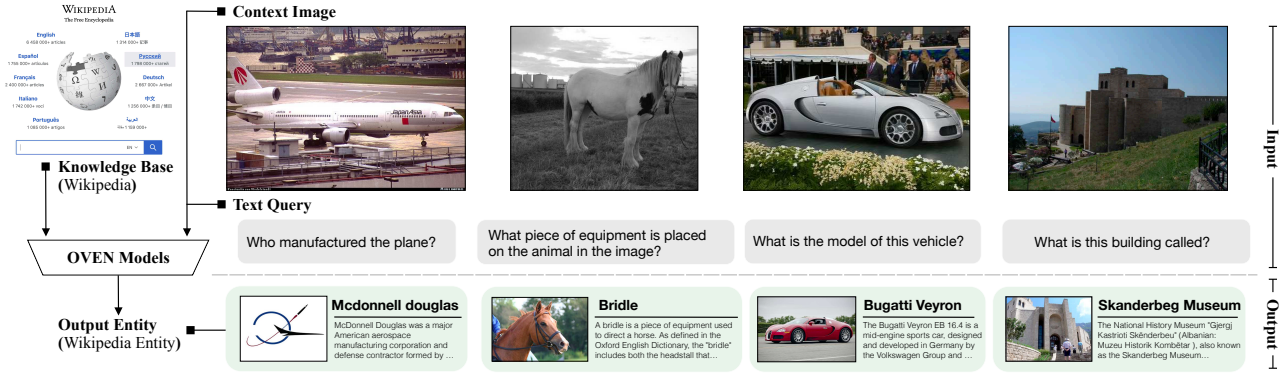
In major fields of AI, such as computer vision, natural language processing or speech recognition, deep learning has become the dominant, if not the only, paradigm. This dominance of deep learning stems from two major factors 1) the wide availability of large datasets such as ImageNet [25], COCO [26], ADE20K [27], Google Landmark dataset v2 [28] or iNaturalist [29] and in recent years the collection of massive datasets orders of magnitude bigger than previous ones, such as JFM [30] (private), LAION-5B [31], LVD-142M [8] 2) the ever-increasing amount of compute available to industrial labs and researchers, with dedicated research on developing the best chips [32], [33]. Both these factors allow training bigger models for longer and on very diverse data.

By continually increasing datasets size and compute, deep learning enters the era of foundation models [34]. These models that have been trained on large amounts of data, *e.g.* 2 *billion* text-image pairs for OpenClip [35] or 2 *trillion* tokens for Llama 2 [5]. Because of the magnitude of their training set, these models are flaunted to have "seen the world", *i.e.* they have seen diverse data and have a broad comprehension of the world. These generalist models can be used in a "zero-shot" manner, *i.e.* without necessitating training, on a wide range of tasks. They are build to have a general understanding of the data without being expert on specific tasks. Thus, because of their generality, they can under-perform *vs.* expert models on specific tasks or datasets, for instance CLIP in image retrieval [36], or SAM on medical images [37]. Although these models are promised to be the base of numerous AI systems, adapting them remains the most effective for specific tasks.

Modern computer vision. In computer vision, the use of deep learning redefines how images are processed and represented. First successful applications of deep learning for computer vision were based on modern convolutional neural networks [40], before becoming virtually the base of all computer vision methods after its success in large scale settings with MCDNN [41] and the famous AlexNet [42] that won the ILSVRC 2012 challenge [43]. Architectures then evolved with the well-known VGG [44] and ResNets [45]. Recently vision transformers, ViTs [46], [47], have been developed in computer vision by adapting the Transformer [48] from natural language processing. The success of deep learning for computer vision notably comes from the fact that it learns representation, the "embeddings" or "deep features", rather than relying on hand-crafted "expert features" such as SIFT [49]. Indeed, expert features have been designed by researchers using notions of signal processing and our understanding of important aspects of images, *e.g.* color gradients. These hand-crafted features mostly grasp low level cues, and thus they may lack some expressiveness for semantic content, where deep features are data-driven and can represent different level of abstraction to tackle the tasks at hand. Embeddings are



(a) Image taken from [38]. Example of different computer vision tasks solved using deep neural networks.



(b) Image taken from [39]. Visual question answering system based on image retrieval.

Figure 1.2: Illustration of the applications of DNN for computer vision.

high dimensional vectors that can represent complex images in a compact manner and allow comparison using simple tools such as the Euclidean distance. These complex representations have enabled deep learning systems to perform numerous tasks such as image classification [44], [45], [50], object detection [51]–[53], image segmentation [10], [54], [55], which are illustrated on Fig. 1.2a, or visual question answering [39], [56], illustrated on Fig. 1.2b, *etc.*

1.2 Context and motivations.

This thesis stems from the collaboration between Cnam and Coexya. Coexya[1] is an industrial group that, among other, edits software solutions dedicated to intellectual property (IP) management. One of them being Acsepto[2]: a software suite for trademark search and watch, used in 16 Intellectual Property Offices worldwide. One of the Acsepto search engines is dedicated trademark logos (TMs) search. Indeed, when a person, organization or company wants to protect a trademark logo, it has to submit a trademark application to the country's IP office, such as the INPI[3] in France, to ensure that the candidate TM is not confusingly similar to an already existing logo. Because of the magnitude of their database, *e.g.* 3.2M registered TMs in France, IP offices have had to rely on the automation of the search process. This search process is illustrated by Fig. 1.3. Given a logo, referred to as a "query", that an applicant would like to register, *e.g.* the ICML logo here, Acsepto retrieves a list of logos in a client's database that are the most similar to the query.

Figure 1.3: Example of a query in Coexya's Acsepto with the ICML logo. Contrarily to the standard image retrieval academic setting, for most trademark logo applications, there are not necessarily positive image results. Some logos are more relevant than others.

[1]Coexya's website: https://www.coexya.eu/
[2]Acsepto's description: Fiches-produits-Acsepto.pdf
[3]INPI's website: https://www.inpi.fr/

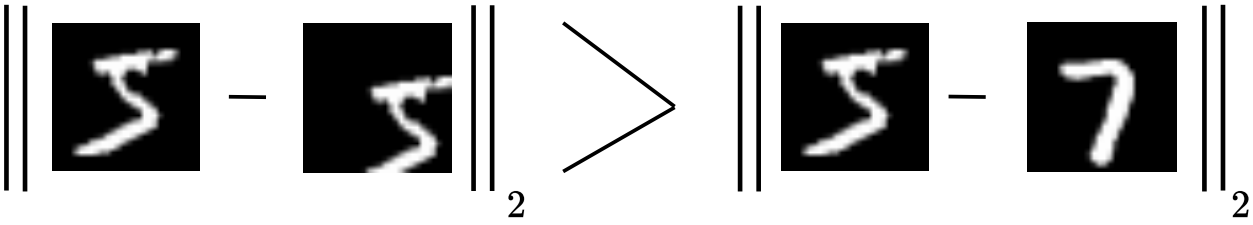


Figure 1.4: The pixel-wise L_2 distance between an image and a translated version of itself is greater than with a completely different image. This illustrates the necessity to design more powerful representations to compare images. (Note that for the content, the inequality also holds true: $5 - 5 = 0 > 5 - 7 = -2$).

Querying a database with an image is a task of computer vision referred to as content-based image retrieval (CBIR). CBIR is based on *image representation*. It consists in building representations of images that enable their comparisons. Indeed, comparing two images based on their raw pixels, *e.g.* doing a L_2 distance pixel-wise, is not accurate and is very sensitive to small variations of an image. This is illustrated on Fig. 1.4, by translating an image of the MNIST dataset [57] by few pixels the L_2 distance becomes higher than with another image. To this end, Acsepto has been based in previous versions on handcrafted and engineered representations of images; its final version before deep learning was based on a mixture of several algorithms, including 2D-Fourier transform [58], 2D-Zernike polynomials [59], HOG features [60], SURF features [61], FCTH features [62] and an in-house algorithm. The representations derived from these different algorithms allowed to focus on different aspect of images, *e.g.* the shapes or color gradients.

Coexya has since adopted deep features following the rise of deep learning in computer vision. One of the strength of DNN is that they are able to learn representation of the images based on data: we say that DNN are "data driven". This allows creating deep representations spaces where distances are perceptual. Meaning that two images that are visually or semantically similar will be close in the Euclidean distance sense. Coexya's first deep models rely on fine-tuning ImageNet pre-trained DNN, *e.g.* the ResNet-50 [45], on their internal database annotated with the Vienna classification[4], a standardized multi-label classification of trademarked logos established by the World Intellectual Property Organization[5]. The deep features extracted from these fine-tuned DNN are subsequently used to compare trademark logos with one another. Fine-tuning is important as it allows adapting the model to a different domain, *e.g.* trademarked logos for Coexya. It also allows learning representations that can distinguish subtle differences in

[4]The Vienna classification: https://www.wipo.int/classifications/vienna/en/index.html
[5]WIPO: https://www.wipo.int/portal/en/index.html

images, indeed datasets in image retrieval are "fine-grained", which is not the case for generalist datasets such as ImageNet. With this collaboration, Coexya sought to improve their models used in their Acsepto software, by improving their predictive performances and making them more reliable.

This context leads use to address the notion of robustness of DNN under three different perspectives:

1. Challenge 1: Robustness in optimization, where we design theoretically sound training loss, leading to better performances on the evaluation metrics.

2. Challenge 2: Robustness of the learned rankings, to mitigate *mistake severity* and ensure alignment of the ranking with human preferences by relying on hierarchical annotations.

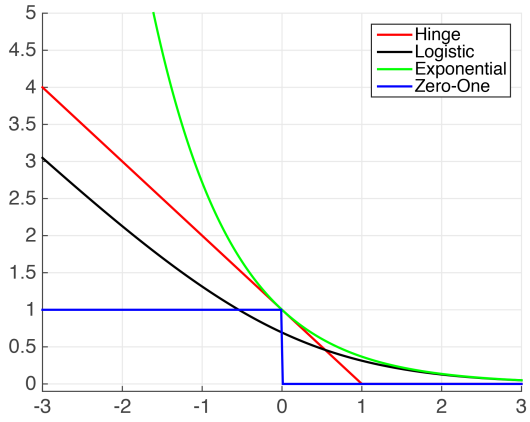
3. Challenge 3: Robustness of the models, by detecting out-of-distribution images using data-driven models to estimate the density of the training images.

1.2.1 Challenge 1: Optimization of non-smooth and non-decomposable metrics.

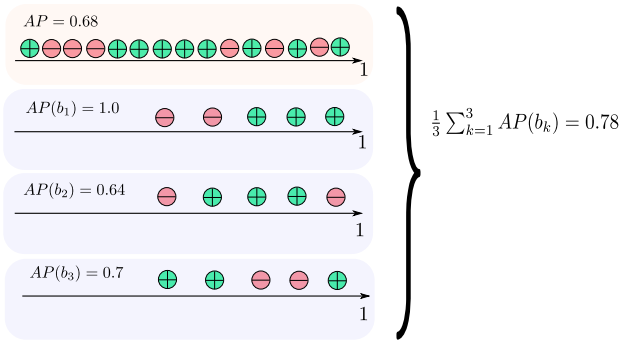
In order to learn representations, DNN are trained on a dataset by minimizing a loss function. The gradients are computed from outputs of DNN. Using the back-propagation algorithm [63], a gradient is computed for each layer of the DNN. The weights are then updated using stochastic gradient descent (SGD). This training paradigm relies on loss functions that are differentiable, *i.e.* the gradient of the loss with respect to the output of the DNN can be computed and is informative. In order to have a DNN that is trained for a specific task, the best case scenario is to be able to optimize the evaluation metrics during training. For instance, this is possible for the standard regression metrics: the mean squared error (MSE). However, for several losses this is not possible, *e.g.* for the well known 0/1 loss used in classification, illustrated in black on Fig. 1.5a. Indeed, it is a step function and its gradients are either 0 or undefined, making them uninformative for SGD. It thus requires the use of a "surrogate" loss that is differentiable, such as the hinge loss or the cross-entropy, which is the loss used for classification in practice. These losses are illustrated on Fig. 1.5a.

Image retrieval systems are evaluated with ranking-based metrics, *e.g.* average precision (AP), recall at k (R@k) or normalize discounted cumulative gain (NDCG). These metrics are used because image retrieval is a strongly unbalanced task, *i.e.* there are many more negatives than positives. Indeed, given a query image, most of the images in the database will be

irrelevant. For instance, on Fig. 1.2b when querying the image of the airplane most images in the database are not a "McDonnell Douglas", thus are irrelevant. These metrics are based on the ranking operator, which can be derived from step functions, as will be detailed in Sec. 2.2. As they are based on step functions, they suffer from the same issues as the 0/1 loss: they are not differentiable, thus fine-tuning image retrieval models requires designing appropriate surrogate losses. This issue has been long studied, and has been addressed by either using coarse upper bounds, *e.g.* the contrastive loss [64], triplet losses [65], proxy losses [66] or using approximations of the rank which allows fine approximation of the target metrics [67]–[70]. For instance, Coexya relied on classification based training which mismatches image retrieval evaluation metrics, leading to suboptimal performances. Designing surrogate losses that correctly approximate the evaluation metrics, while keeping important robustness properties such as being upper bounds, is a challenging problem.



(a) In order to optimize the 0/1 loss (in blue), a surrogate loss is needed, *e.g.* the logistic or cross-entropy loss in black.*



(b) Image retrieval evaluation metrics, *e.g.* AP, are not decomposable. The average AP estimated on the blue batches is 0.78, whereas the true global values in yellow is 0.68.

Figure 1.5: Stochastic gradient descent relies on loss functions that are differentiable Fig. 1.5a, and that are decomposable, which is not the case for AP Fig. 1.5b.

*Image taken from `www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote10.html`

Furthermore, these metrics are "list-wise", *i.e.* the value of the metric for a given query depends on other examples. Therefore, they are not linearly separable between examples. This makes them "non-decomposable". Their values estimated on subset or mini-batches of data are biased. This is illustrated on Fig. 1.5b, where the average of the AP on each batch (from second to last rows) is greater than the global AP (top row). As mentioned previously DNN are optimized using SGD which is used in practice for both computational and performance reasons. Other losses, *e.g.* the cross entropy, do not face this issue and can be estimated using

mini-batches of data. Non-decomposability is also an issue for other metrics, such as the Dice score [71], [72]. While non-decomposability is a known issue, it has been less studied than the non-differentiability issue. Approaches that tackle non-decomposability using *ad hoc* and brute force methods, *e.g.* increasing the batch size at the expense of computational efficiency in [67] or storing previous batches [68], [73].

1.2.2 Challenge 2: Brittleness of DNN outputs and mistake severity.

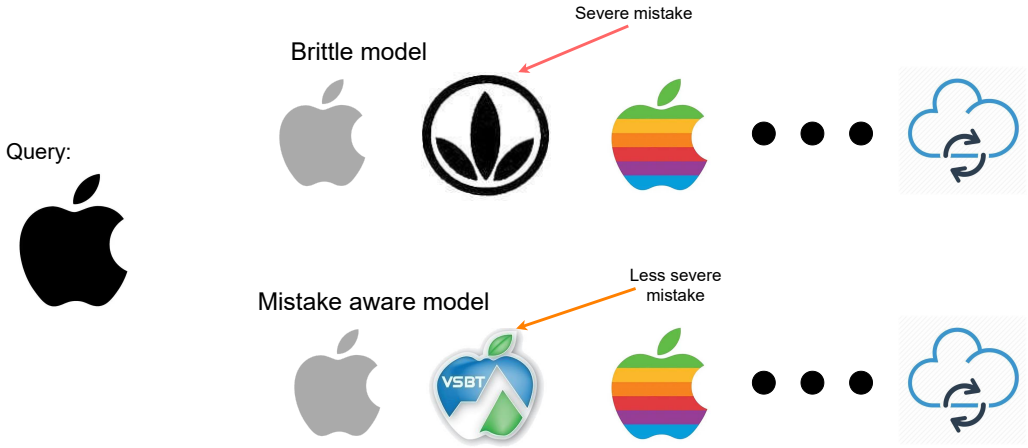


Figure 1.6: For a given query, we illustrate two retrieved results. The top makes an error that is more severe than the bottom one.

Whereas DNN are very powerful to represent images and perform specific tasks, they can be surprisingly brittle to different factors and can have unstable outputs. One notorious aspect of their brittleness and instability is the so-called *adversarial attacks* [74], where the output of DNN can change drastically while the input changes slightly. Another instability is that DNN have little control over the severity of the errors they commit in terms of human understanding. This was notably observed in [75], where they show on ImageNet classification [43] that while predictive performances from AlexNet [42] to the more evolved ResNet-50 [45] have increased, the severity of the errors have not lowered. This can be in part explained by "shortcut learning" of DNN [76]. Indeed, DNN tend to learn tasks by learning shortcuts, *e.g.* looking at the background in image classification rather than the primary object. This can implicate that rather than learning a semantic of images to recognize, they rely on features unrecognizable to humans, leading to severe mistakes when they commit some. Similarly, CBIR systems can exhibit failure cases, where they commit really severe mistakes when they mistakenly retrieve some false positives. This is illustrated on Fig. 1.6, where, given an "Apple" logo, two models can make errors that are more or less severe. Contrarily to previous handcrafted features

used to represent images, *e.g.* SIFT [49], HOG [60] or VLAD [77], DNN representations lack interpretability. This issue make the comprehension of these instabilities harder to understand and fix in practice.

Figure 1.7: For a given query (*e.g.* the Apple logo) there are several groups that more or less relevant: old Apple logos, logos depicting apples, fruits, and finally logos that do not share any similarity. Note that Apple has engaged lawsuits for logos in groups 2 and 3*.

* sources: www.huffpost.com/entry/apple-sues-woolworths-ove_n_309450
www.techspot.com/news/99131-apple-wants-trademark-images-apples.html
www.wired.com/2008/10/apple-takes-on/

The definition of *mistake severity* is challenging. It is connected with the human preferences and understanding of the tasks. Because "human preferences" are difficult to define in practice, an interesting area of research uses hierarchical relations between image labels as a proxy. For the well-known ImageNet dataset, the mistake severity can be derived from hierarchical relations of the WordNet [78] syntactical database. In information retrieval researchers use "graded relevances" that modelizes the importance of the retrieved instances for a given query. Subsequently, using them in graded metrics such as NDCG [79] or a graded AP [80]. Similarly, different levels of relevance can be created for CBIR of trademark logos, this is illustrated on Fig. 1.7, for a given query retrieved logos can be more or less relevant. This task was also addressed using surrogate losses in [81], [82]. Reducing the mistake severity is important for search engines, including Coexya's Acsepto, in order to convince the users to use them and build trust.

1.2.3 Challenge 3: Out-of-distribution sample detection.

Although DNN predictive performances have increased, as discussed previously, detecting when samples are out-of-distribution (OOD) remains a challenging task. The task of OOD detection is a challenging direction and has been extensively studied. OOD detection is another important challenge to make DNN more robust in their predictions. It gives the ability to detect whether they should process an image or not. In critical applications it is important that a DNN knows when it does not know, *e.g.* in medicine, defense or in autonomous driving, a system should give back the control to a human decider if it does not know what to do. OOD detection is also a challenging research direction for CBIR. It is for instance interesting for Coexya. Indeed, one industrial advantage of Coexya is its private databases of trademark logos. In order to remain competitive, one direction might be to scrape the web or create subset of very large dataset [31] for new logos or different training sets. This requires being able to detect whether an image is of a logo or not. This is illustrated on Fig. 1.8, a model has been trained on the ID training set, and must detect at inference time whether an image is "in-distribution", *i.e.* it comes from the ID test set, or "out-of-distribution", *i.e.* it is an OOD data.

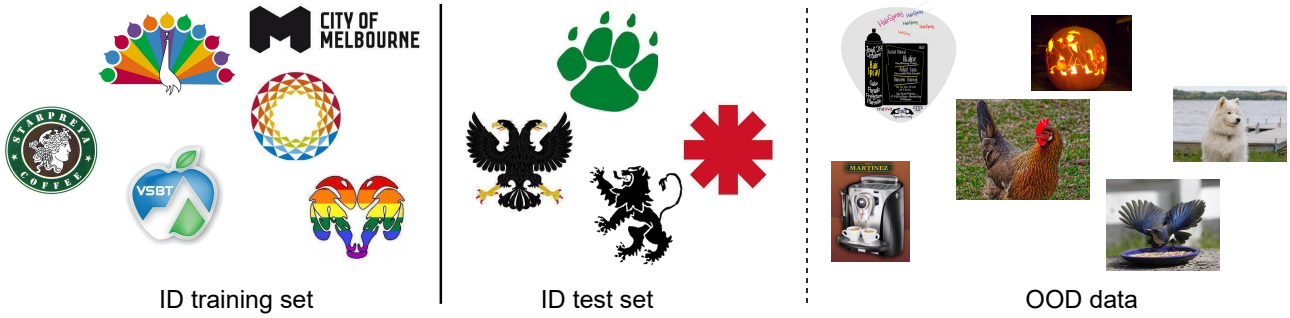


Figure 1.8: Trademark logo detection. A model must decide whether images are in-distribution (ID), *e.g.* from the METU dataset [83], or out-of-distribution (OOD), *e.g.* from the ImageNet dataset [43].

The difficulty of OOD detection notably comes from the overconfidence of deep models. This was identified in [84], where the authors show how deep models suffer from overconfidence. For instance, in classification, this means that DNN will give a strong probability to a wrong class. This makes naïve approach, such as the well-known maximum softmax probability (MSP) of [85], fail in some cases. Indeed, MSP uses the maximum probability of a DNN as a confidence measurement, which is not sufficient in practice. There have been several attempts at solving OOD detection. Authors of [86], [87] tried to enforce OOD detection by integrating OOD samples in the training dataset. Other methods relied on autoencoders to access a likelihood

on an image [88]. State-of-the art methods try to estimate the density of the ID training set of DNN, *e.g.* [89] approximate the ID density using a Gaussian Mixture Model or, more recently, authors of [90] approximate it using the k-nearest neighbors density. There has been a shift of paradigm in the OOD detection literature, following the recent advent of very large off-the-shelves models that have strong predictive performances. Thus, recent methods for OOD detection follow the *post-hoc* paradigm, where they leverage pre-trained neural networks [89], [91], [92].

1.3 Summary and contributions.

In this thesis, we address several aspects of the robustness of deep neural network. Specifically, we introduce a method for the robust optimization of ranking metrics that are used in image retrieval, by tackling both the issues of non-differentiability and non-decomposability (Chapter 3). We show that using the hierarchical relations between labels, we can train more robust neural networks with respect to their errors in Chapter 4. We also investigate post-hoc robustness of DNN with their out-of-distribution detection performances and how to boost them using energy-based models in Chapter 5.

Outline. In regards with the challenges mentioned above, our contributions are the following:

- Chapter 3: Optimization of Ranking Losses for Image retrieval.

In this chapter, we address the two limitations of optimizing ranking-based metrics identified in Challenge 1: non-differentiability and non-decomposability. We define a new training framework that tackles both issues. It uses an approximation of the ranking function, SupRank, to provide a smooth and upper-bound surrogate losses. SupRank is an accurate approximation of the rank, and has sound mathematical properties and experimental performances. We also show the theoretical advantages that SupRank has compared to the smooth approximations of [69], [70]. We optimize a second loss function during training, to enforce the decomposability of ranking losses during mini-batch training. It has a small computational overhead, and make ranking optimization feasible in small batch settings. We show in a theoretical analysis how this additional objective helps the decomposability of ranking loss optimization. This framework is general and can be applied to numerous ranking losses. In this first chapter we concentrate on the standard image retrieval setting and apply this framework to two ranking-based metrics: average precision and recall at k to optimize DNN for image retrieval. We show in extensive

experimental validations the interest of our framework. We first show that it compares favorably against recent methods from the literature that optimize ranking-based metrics. We show that our framework allows the optimization of ranking-based metrics in small batch settings. We then show that our framework is robust to hyperparameters. Finally, we compare our method against state-of-the-art methods from the literature and show it outperforms competitions on several datasets, including small to large scale, and validate our method on one of Coexya's internal datasets.

- Chapter 4: Hierarchical Image Retrieval for Robust Ranking.

In this chapter, we question the definition of the similarity used in image retrieval in order to address the brittleness of DNN with respect to the severity of their errors. We expose the limitations of the standard binary similarity commonly used in image retrieval by looking at the model's robustness when making mistakes. To mitigate this brittleness of mistake severity, we propose to use hierarchical relations between labels to define a more rich definition of the similarity between two images. To integrate this similarity during training and for evaluation, we introduce an extension of the average precision, the hierarchical average precision or $\mathcal{H}$-AP. To showcase the interest of using hierarchical relations, we optimize two different hierarchical metrics using the framework of Chapter 3: $\mathcal{H}$-AP with HAPPIER, and NDCG with ROD-NDCG. Using this framework allows us to have more robust training than approximations used in information retrieval [79], [80]. Furthermore, optimizing surrogate of evaluation metrics lead to better performances than other surrogate losses used in hierarchical image retrieval such as [81], [82]. We then discuss the assumption we made of having access to hierarchical labels. We show how to annotate in practice image retrieval datasets with hierarchical labels. We use a semi-automatic pipeline to extend a well-known landmarks retrieval dataset, Google-Landmarks v2 [28], with hierarchical labels. We show in experimental validation that both HAPPIER and ROD-NDCG i) are on par with state-of-the-art methods for standard image retrieval ii) outperform by a large margin standard image retrieval methods on hierarchical metrics, iii) outperform other hierarchical methods on hierarchical metrics and standard image retrieval. Our results hold for six hierarchical datasets of the literature and our hierarchical GLDv2. We also show the interest of HAPPIER for logo retrieval on two of Coexya's internal datasets. We conduct ablation studies of our framework to show its robustness to hyperparameters. Finally, we qualitatively show that HAPPIER creates an embedding space that is better organized than non-hierarchical methods, and qualitatively show the lower mistake severity of HAPPIER *vs.* non-hierarchical methods.

- Chapter 5: Post-hoc out-of-distribution detection.

In this chapter, we study another aspect of DNN robustness: post-hoc out-of-distribution (OOD) detection, as described in Challenge 3. We leverage the energy-based models (EBM) framework [93] to introduce a new method for post-hoc OOD detection: HEAT. It is based on two components: residual learning and composition of energy functions. We first use EBMs to learn a residual function for different methods of the OOD detection literature. Indeed, several methods of the literature are based on approximation of the density of the training dataset, *e.g.* [91] uses a Gaussian mixture model to approximate the ID density or [94] uses an energy score derived from the output logits of a DNN. However, because of their strong prior biases, these methods lack expressiveness to correctly approximate the ID distribution. Learning a residual term with an EBM allows more expressiveness. Another aspect of the different modeling biases of these methods is that they will be able to detect different type of OOD samples. We show that using the energy function composition, we are able to combine effectively several types of corrected prior OOD scorers to improve overall OOD detection performances. Finally, HEAT is a post-hoc methods, which allows it to be used on virtually any off-the-shelf deep model with strong predictive performances. We focus on image classification as it is a standard benchmark in the OOD detection literature. In our experiments, we show how both the components of HEAT improve OOD detection performances. We compare HEAT to state-of-the-art post-hoc OOD detection on two standard benchmarks CIFAR-10 and CIFAR-100 and on the large scale ImageNet. We also show that HEAT works with several architectures, including CNNs and Vision Transformers. Finally, we show that HEAT is robust to low data regimes, and with respect to its hyperparameters.

1.4 Related publications.

This thesis is based on the material of the following papers:

Publication	Chapter
[95] **Elias Ramzi**, Nicolas Thome, Clément Rambour, Nicolas Audebert, and Xavier Bitot. "Robust and Decomposable Average Precision for Image Retrieval." Advances in Neural Information Processing Systems, 34$^{\text{th}}$ (NeurIPS, 2021). online: https://arxiv.org/abs/2110.01445	3
[96] **Elias Ramzi**, Nicolas Audebert, Nicolas Thome, Clément Rambour, and Xavier Bitot. "Hierarchical Average Precision Training for Pertinent Image Retrieval." in Proceedings of the 17$^{\text{th}}$ European Conference on Computer Vision (ECCV, 2022). online: https://arxiv.org/abs/2207.04873	4
[97] Marc Lafon, **Elias Ramzi**, Clément Rambour, Nicolas Thome. "Hybrid Energy Based Model in the Feature Space for Out-of-Distribution Detection." in Proceedings of the 40$^{\text{th}}$ International Conference on Machine Learning (ICML, 2023). online: https://arxiv.org/abs/2305.16966	5
[98] **Elias Ramzi**, Nicolas Audebert, Clément Rambour, André Araujo, Xavier Bitot and Nicolas Thome. "Optimization of Rank Losses for Image Retrieval." Under review, IEEE Transactions on Pattern Analysis and Machine Intelligence (under-review – TPAMI). online: https://arxiv.org/pdf/2309.08250.pdf	3,4

Chapter 2

Related work

In this chapter we first discuss major trends in image retrieval before giving a general overview of the three directions identified in Chapter 1, i) optimization in image retrieval, ii) hierarchical learning for mistake severity, and iii) post-hoc out-of-distribution detection. In Sec. 2.1 we give an overview of the evolution of image retrieval, from handcrafted features to the use of deep neural networks and advanced pipelines. We will then discuss in Sec. 2.2 modern training schemes with proxy losses developed for image retrieval and issues arising from mini-batch optimization in deep learning. In Sec. 2.3 we will present the evolution of hierarchical classification in computer vision, draw inspiration from the information retrieval community, before reviewing the hierarchical image retrieval literature. Finally, in Sec. 2.4 we will discuss out-of-distribution (OOD) detection and its current post-hoc paradigm, before discussing energy-based models and our motivation for their use in OOD detection.

Content

2.1 Trends in image retrieval.

Image retrieval is a subdomain or task of computer vision. It refers to the process of retrieving relevant images from a large collection based on a query image or a textual description [99]. In this thesis we concentrate on image queries, *i.e.* content based image retrieval, and do not discuss text to image retrieval [100], [101] or composed image retrieval [102], [103]. Image-to-image retrieval is illustrated on Fig. 2.1, given a query image of the "Moulin Rouge", the image retrieval systems should find in database images that are similar, *i.e.* other photos of the "Moulin Rouge". Image retrieval can be used in *e.g.* search engines such as Coexya's Acsepto, in re-identification of vehicles [104] or in visual question answering systems [39], [56]. Among the difficulties of image retrieval is that the datasets are *fine-grained*, this means that the visual difference between similar and dissimilar images are hard to distinguish. Furthermore, the representation of images in image retrieval needs to be invariant to the angle, scale and lighting settings at which the pictures are taken. This is illustrated on Fig. 2.1, where images of the "Moulin Rouge" are taken from the right or left side, from up-close or far away, and during night and day.

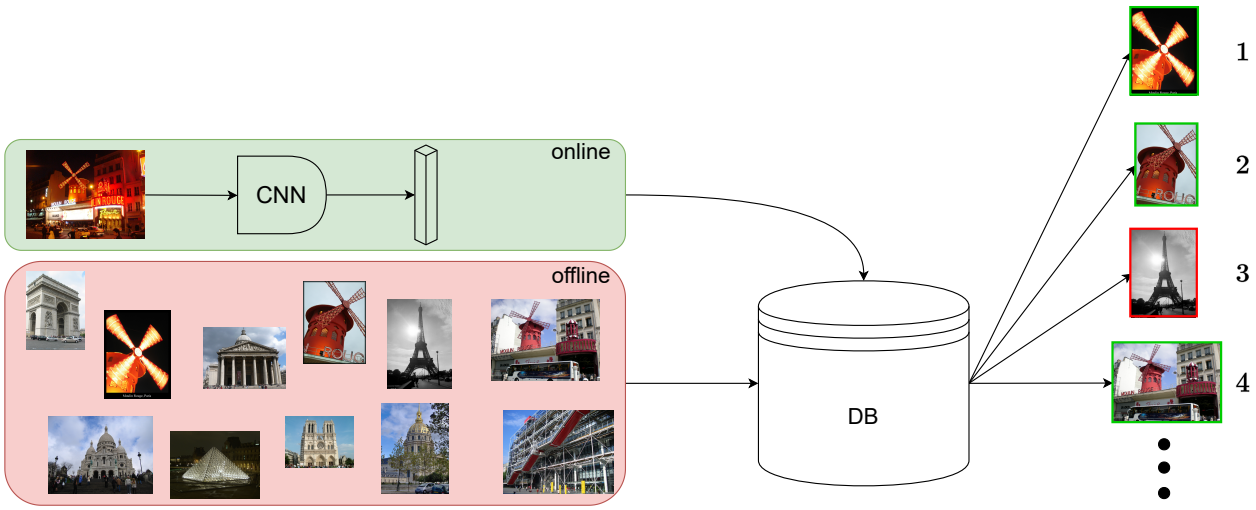


Figure 2.1: In green, the query is an image of the "Moulin Rouge". The goal of image retrieval is to find images similar to the query, *i.e.* other images of the Moulin Rouge, in the database. Images are then ordered from most to least similar, *e.g.* by decreasing cosine similarity between the vectorial representations of the query image and images in the database.

2.1.1 Background in image retrieval.

Historically, image retrieval systems heavily relied on handcrafted features, such as SIFT (Scale-Invariant Feature Transform) [49], [105] or HOG (Histogram of Oriented Gradients) [60]. First pipelines typically involved several steps.

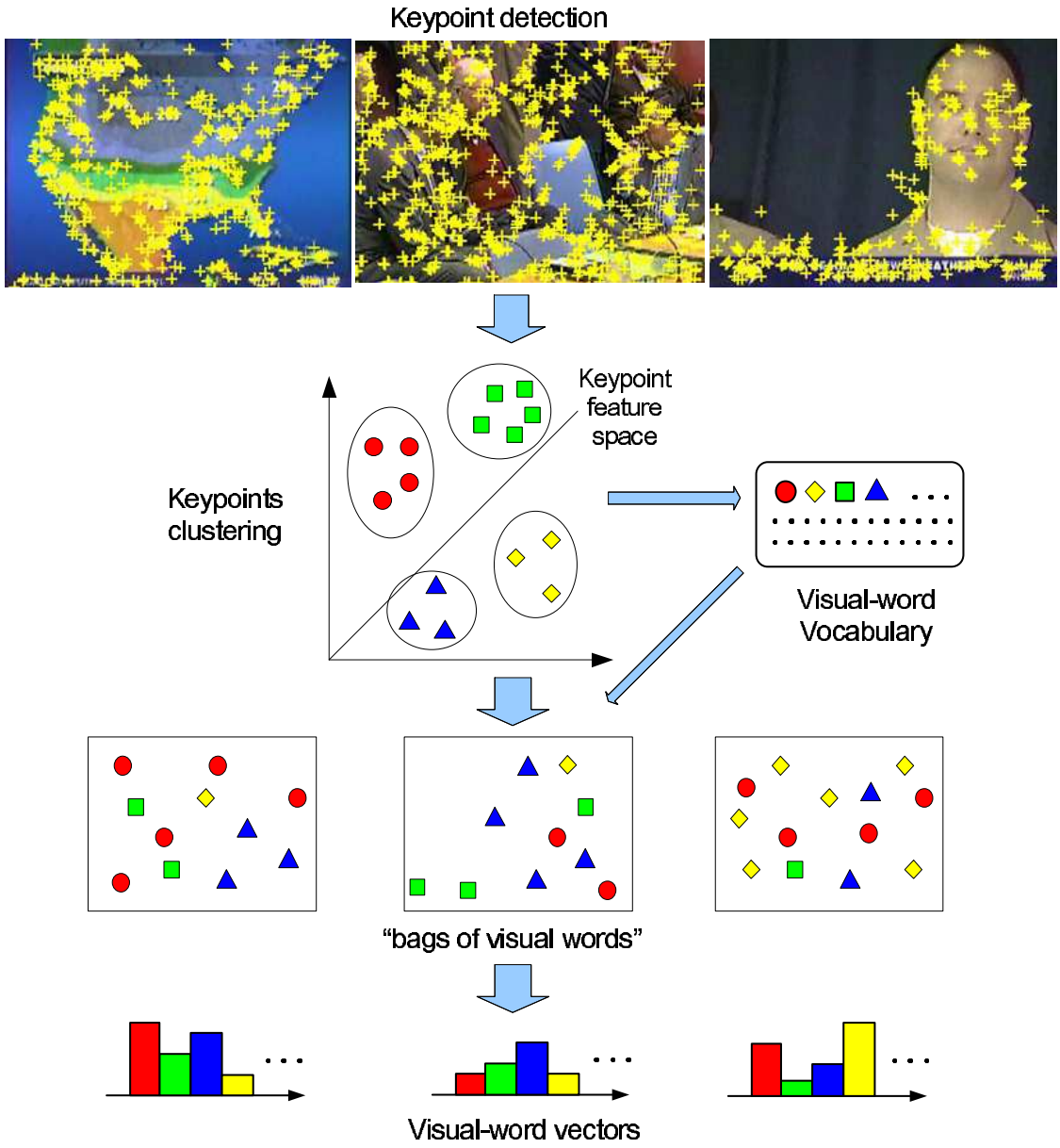


Figure 2.2: Image taken from [106]. Image retrieval systems used to rely on several steps to process images: keypoint detection, feature extraction, the creation of visual words by clustering keypoint representations and indexation of images.

1. Finding region of interests, or key-points, on images, *e.g.* using affine-invariant Hessian regions [107].
2. Computing descriptors, *e.g.* 128D SIFT descriptors in [108].
3. Cluster the descriptors, or "quantized", using K-means [109], [110] or hierarchical K-means [111], to create a vocabulary of visual words.
4. Finally, index the images with their visual words and similarity is computed using L_2 distance between visual words of two images [112]. A TF-IDF weighting scheme can also be applied to weigh down recurring words [112].

This pipeline is illustrated on Fig. 2.2. Other methods, *e.g.* VLAD (Vector of Locally Aggregated Descriptor) [77], worked on reducing the computational performance of image retrieval systems by aggregating local descriptors into a single global representation, which can be subsequently used for nearest neighbor search in a database.

However, handcrafted features captured basic visual cues, *e.g.* color gradients, but lacked the ability to capture high-level semantic information. By learning data-driven features that can represent high level of semantic content, deep learning has dramatically changed the landscape of image retrieval.

2.1.2 Advent of deep learning in image retrieval.

The advent of deep learning and convolutional neural networks (CNNs) [40], [45], and more recently Transformers [46], has revolutionized the field of image retrieval. Deep neural networks, have demonstrated remarkable prowess in learning complex and semantic representations of images, enabling the automatic extraction of discriminative features directly from raw pixel data. DNN learn global representation of images, thus removing the need for key-point detection and directly creating descriptors representing the semantic content of the images. Because they learn global representation, comparing two images can be simply done with cosine similarity or Euclidean distance.

First attempts at using deep learning for image retrieval were based on neural networks that have been trained on large datasets, *e.g.* AlexNet [42] trained for image classification on ImageNet. Researchers tried to enforce some ideas of the previous pipelines, such as regions of interest and geometric information, by changing the global aggregation from maximum or average aggregation to more sophisticated aggregation [113]–[115]. For instance [113] proposed R-MAC, *i.e.* regional maximum activation of convolutions, where instead of aggregating all the

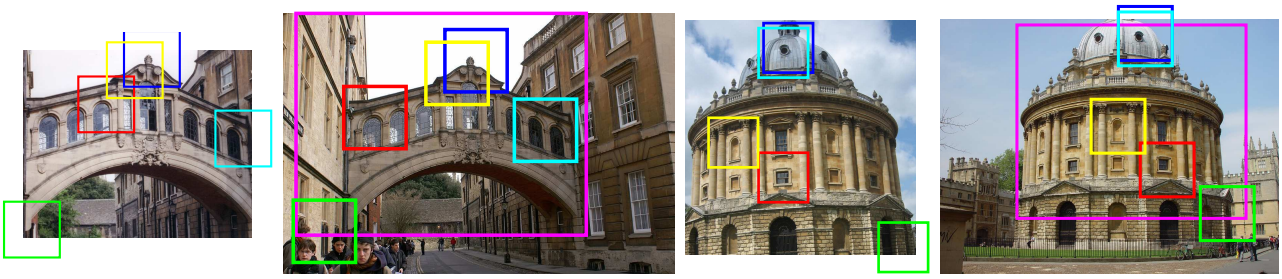


Figure 2.3: Image taken from [113]. Query objects (left) and the corresponding localization in another image (right) are shown. We visualize the patches that contribute the highest to the image similarity score. Displayed patches correspond to the receptive field of CNN activations. Object localization is displayed in magenta, while different colors are used for patches in correspondence.

outputs of the convolution with the max-pooling (as done in AlexNet) they encode sub-regions of the image using regional max-pooling. This is illustrated on Fig. 2.3 where authors of [113] display for two queries which regions participate the most to the final similarity.

It was quickly shown that fine-tuning models worked best for image retrieval in [116], as it allows to i) adapt to potentially a different domain and classes ii) learn discriminative features as image retrieval datasets are fine-grained. Ensued an active research direction dedicated to end-to-end learning [117] for image retrieval, with the use of contrastive loss [118], *e.g.* in [64] that fine-tune neural networks on a dataset created using a structure-from-motion (SfM) pipeline [119], [120]; triplet losses [121]; proxy-based losses [66], ranking losses [70], continuing in some of the latest big conferences, *e.g.* ICML 2023 [122], ICCV 2023 [123]. The research of a good training loss notably comes from drawbacks discussed in Chapter 1 of image retrieval evaluation metrics, resulting in a difference between the training loss and the evaluation metric, which is further discussed in Sec. 2.2.

Another direction to improve image retrieval systems is by building dedicated architectures. For instance, authors of [124]–[126] take advantage of hyperbolic embeddings [127]. Authors of [128] introduce regional & scale GeM extending the well-known GeM-pooling [64] to create global features that extract carefully the local features.

Other methods that perform well are based on the use of local features. R2D2 [129] learns local descriptors and uses measures of repeatability and reliability to select the best local descriptors. Authors of [130] use "super features" that are created using an attention module on top of CNNs local features, that are an intermediate between global representations and local features. Recently, authors of [131] adapted the transformer architectures to best take advantage of their local features.

2.1.3 Post-processing pipelines.

Other research directions are dedicated to improving image retrieval systems' i) predictive and ii) computational performances using post-processing strategies.

To improve predictive performances, there exists three standard post-processing strategies [132]. i) (α-weighted) Query expansion [64], [133] refines the query embeddings using the most similar images in the database. It is also included in losses to stabilize training [122], [134]. ii) Re-ranking [108] is a two stage pipeline. With the first stage aiming at high recall, in the top-k a lot of images should be similar to query image, the second stage aims at precision, the first images should be similar to the query. The first stage often relies on global features for efficiency. The second stage is more compute intensive and often rely on local features or learned key-point descriptors [135] and matching algorithms, *e.g.* RANSAC [136] or more recently dedicated re-ranking architectures, *e.g.* 4D convolutions [137] or transformers [138], [139]. Recent methods [128] perform re-ranking using global descriptors derived from query expansion, which significantly lower the computational requirements of the re-ranking stage. iii) Database-side feature augmentation [140], where database embeddings are refined using the graph of positive images.

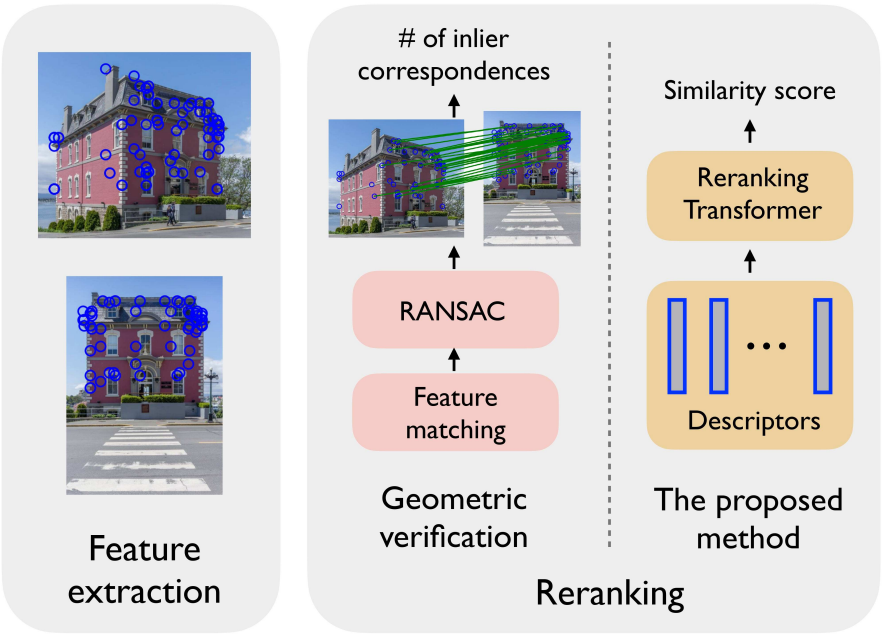


Figure 2.4: Image taken from [138]. Re-ranking is based on local features, [108] matches local features using RANSAC [136]. The authors of [138] introduce a transformer [48] that learns this matching.

Another issue faced by image retrieval systems in practical settings is the computational requirements of querying large databases. One of the most popular methods to lower the computational cost of querying a database is using Product Quantization (PQ) [141]. PQ works by dividing embeddings into blocks. Each block is quantized using a codebook, for instance by clustering the blocks using K-means [109], [110]. PQ allows efficient approximate nearest neighbor search while boosting performances *vs.* standard quantization, as it reduces quantization noise by quantizing smaller vectors. Another direction is using deep hashing [142] where images are encoded in binary vectors. The Hamming distance is then used to compute the similarity between vectors. These methods drastically reduce the cost of computing the distance, but also the memory footprint of storing and loading the images representations.

In this thesis, we concentrate on the fine-tuning aspect of the image retrieval pipeline. We stress that the research towards post-processing the rankings is orthogonal to ours. Our work focuses on the global representation that are use for initial ranking, and could be used in a more complex pipeline involving one or more components presented in this section. Specifically, we build upon a long line of work that design the best proxy-loss for image retrieval, which is discussed in the next section.

2.2 Losses in image retrieval.

Image retrieval is a very unbalanced task: most of the examples in a database are negative wrt. a query image. In order to evaluate image retrieval systems, researchers use ranking-based metrics that take into account this unbalance: *e.g.* average precision (AP), recall rate at k (R@k), as discussed in Chapter 1.

The average precision is based on the recall and precision, that are defined below for a query image q:

$$\text{Recall}(k) = \frac{\#\text{ number of positive before k}}{|\Omega^+|} \tag{2.1}$$

$$\text{Precision}(k) = \frac{\#\text{ number of positive before k}}{k} \tag{2.2}$$

with Ω the database and Ω^+ the set of images similar to the query.

Note that in image retrieval, the metric often referred to as "Recall" is the recall rate and is different from the usual recall:

$$R@K = \frac{1}{|\Omega|}\sum_{k\in\Omega}\mathbb{1}^+(k), \quad \text{where } \mathbb{1}^+(k) = \begin{cases} 1 \text{ if a positive instance has a ranking smaller than k} \\ 0 \quad \text{otherwise} \end{cases} \tag{2.3}$$

The average precision computes the precision at each recall step. It can also be written as a function of the rank and rank$^+$, which is the rank among the positive images.

$$\text{AP} = \sum_{k\in\Omega}(\text{Recall}(k) - \text{Recall}(k-1)) \cdot \text{Precision}(k) = \frac{1}{|\Omega^+|}\sum_{k\in\Omega^+}\frac{\text{rank}^+(k)}{\text{rank}(k)} \tag{2.4}$$

These metrics evaluate the quality of a ranking and are suited for image retrieval. They are based on the ranking operator. Which can be defined as a sum of Heaviside step function [79], H Fig. 2.8a:

$$\text{rank}(k) = 1 + \sum_{j\in\Omega} H(s_j - s_k) \tag{2.5}$$

This definition can be interpreted as counting the number of samples j that have a similarity s_j with the query image greater than the similarity s_k of instance k.

These metrics have two main issues when dealing with stochastic gradient descent (SGD) as in deep learning. 1) They are not optimizable directly through gradient descent. Indeed, because of the Heaviside function this operator has gradient that are either null or undefined (see Fig. 2.5), which we will sometimes refer to as "non-differentiable". This is illustrated on Fig. 2.5. 2) they are not linearly decomposable with respect to training samples, we say that they are "non-decomposable". Indeed, for a non-decomposable metric $\mathcal{M}$, its value on a dataset Ω can not be expressed as a sum of metrics on individual examples, $m(k)$, as in Eq. (2.6).

$$\mathcal{M}(\Omega) \neq \frac{1}{|\Omega|}\sum_{k\in\Omega} m(k) \tag{2.6}$$

These drawbacks limit the use of ranking-based metrics in gradient based optimization framework such as deep learning. In this section, we discuss methods to optimize such metrics for image retrieval. Because these metrics are used in other domains, *e.g.* the average precision is used in multi-label classification and object detection [51]. Furthermore, since the NDCG [143], [144] used in information retrieval and the Dice score [71], [72] used in segmentation are also not decomposable, addressing the issue of non-decomposability may also be applicable in other contexts.

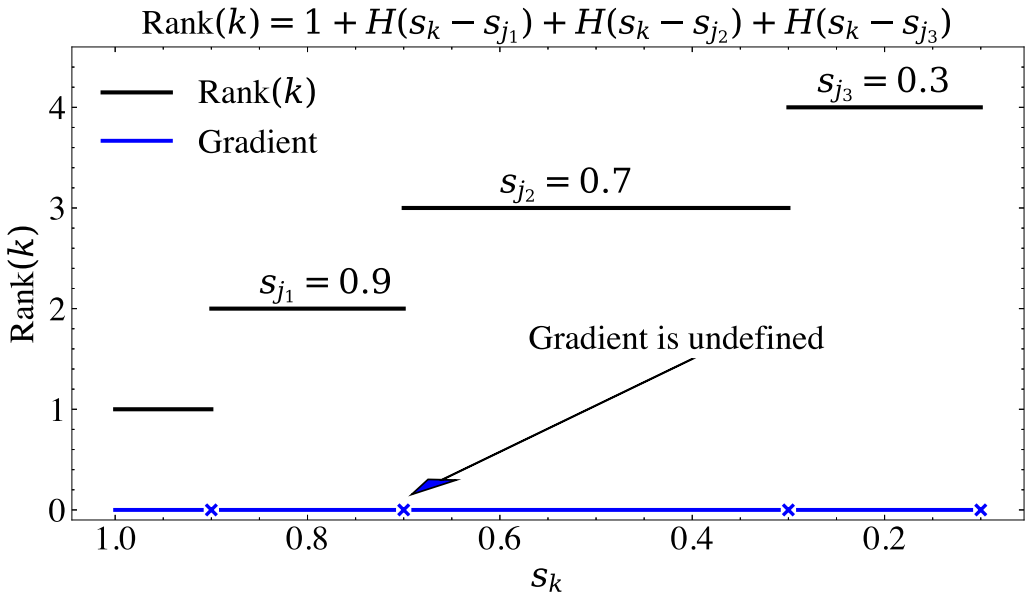


Figure 2.5: When considering instance k, rank(k) is a piecewise constant function of s_k, the cosine similarity between k and the query. The gradient of rank(k) is thus 0, and undefined where $s_k = s_j$.

Discussion on unsupervised image retrieval. In this section, we discussed losses in a supervised setting. Note that an emergent field is that of unsupervised fine-tuning of models for image retrieval datasets. It allows adapting large models to specific domains of image retrieval without the need for the costly annotation process of fine-grained datasets. STML [134] follows recent trends of self-supervised literature [145]–[147] and uses a teacher-student framework to optimize the relaxed contrastive loss [148] a soft-version of the contrastive loss used in *e.g.* [146], [149]. On the other hand, [124], [150] take inspiration from the deep clustering papers [151], [152] and train the models to classify correctly according to pseudo-labels. There have also been some work that use both supervised losses and unsupervised ones, *e.g.* [153] that uses an unsupervised loss to address the granularity difference between training and evaluation.

2.2.1 Smooth Surrogate losses.

As ranking-based metrics used in image retrieval are not differentiable, there has been a focus on defining smooth surrogate losses to optimize DNN for image retrieval.

The image retrieval community has designed several families of proxy-losses to optimize metrics such as AP and R@k. Losses based on tuplets, like pair losses [118], [155], [156], triplet losses (TL) [65], [117], [157] (defined in Eq. (2.7)), or larger tuplets [158]–[160] learn local

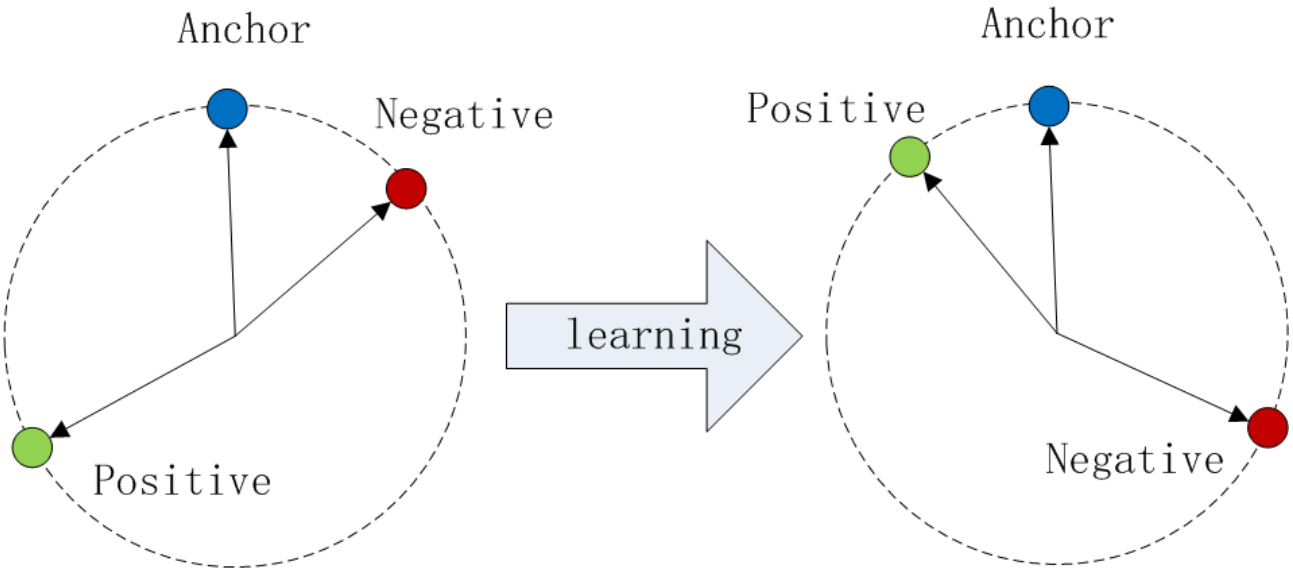


Figure 2.6: Image taken from [154]. The triplet loss optimize local ranking on triplets of example (anchor, positive, negative) such that the distance between that anchor and negative is superior to the one between anchor and positive.

comparison relations between instances (see Fig. 2.6).

$$\mathrm{TL}(q,p,n) = \max(\cos(q,n) - \cos(q,p) - m; 0), \tag{2.7}$$

$$\text{where } \begin{cases} q \text{ is a query image} \\ p \text{ is an image similar to } q \\ n \text{ is a negative image} \\ m \in \mathbb{R} \text{ is a "margin"} \end{cases}$$

These metric learning methods optimize a coarse upper bound on AP, *i.e.* it is an upper bound, however their values strongly exaggerate the values of AP, and need complex post-processing and tricks to be effective [65]. Methods using proxies [66], [161]–[164] have been subsequently introduced to lower the computational complexity of tuplet based training. For instance, for triplet losses, the number of triplets grows cubicly with the number of training samples. These methods learn jointly a deep model and weight matrix that represent proxies using a cross-entropy based loss. Proxies approximates the original data points by minimizing the cross-entropy of data points belonging to the same category. However, these losses do not directly optimize the target metrics.

Another family of losses that has been extensively studied is list-wise or ranking losses, that

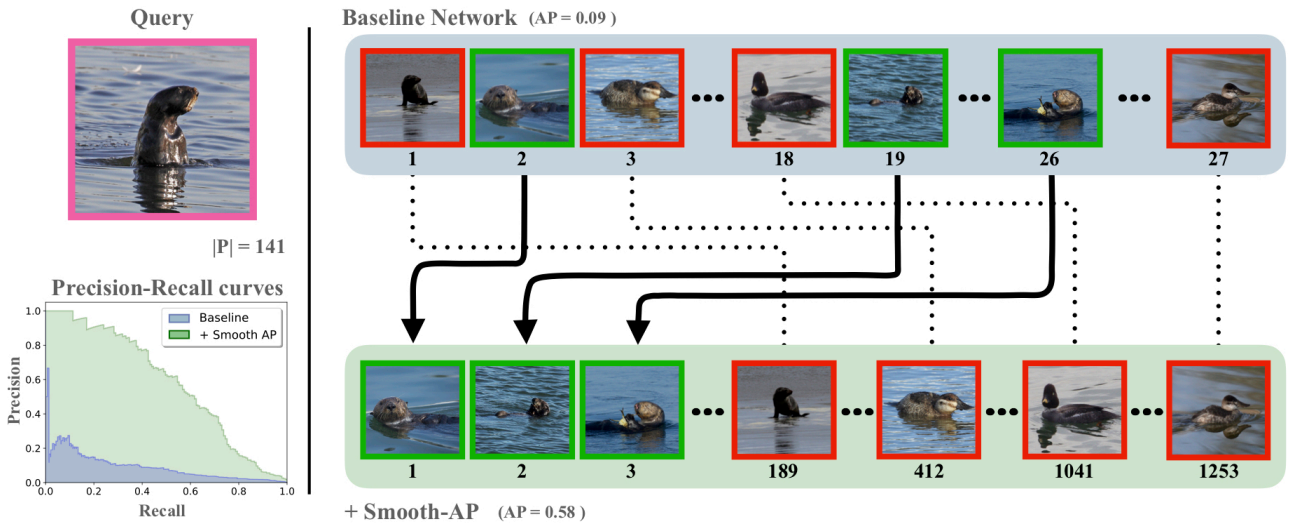
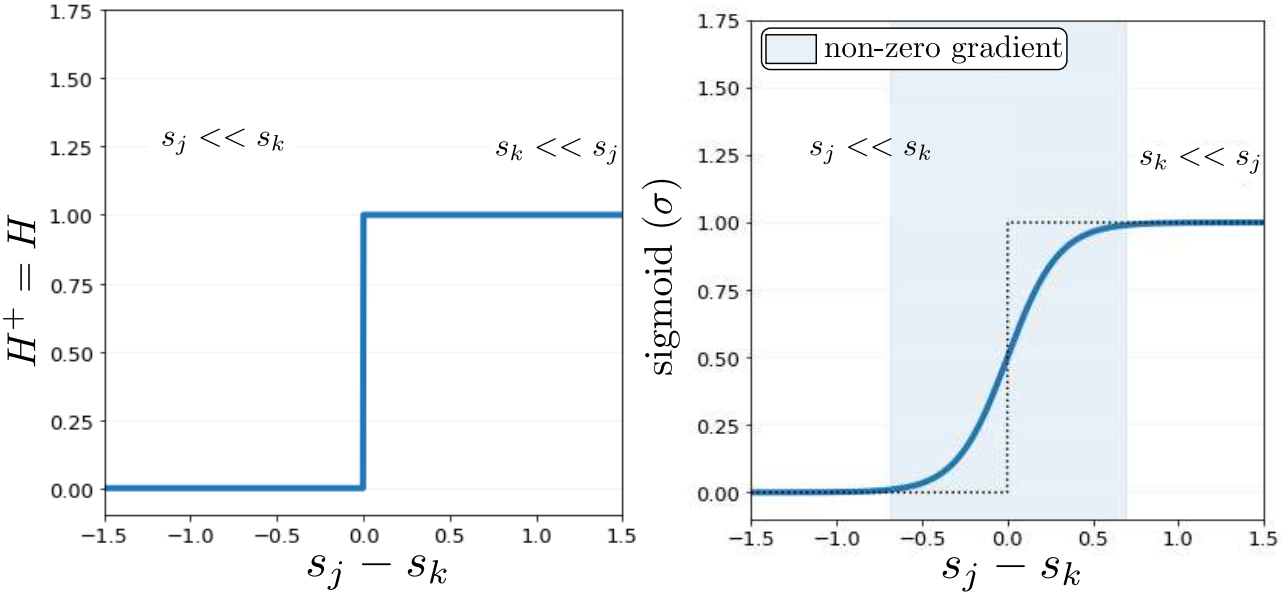


Figure 2.7: Image taken from [69]. Illustration of a ranking loss. It takes a list of positive and negative and optimize such that the correct ranking is achieved on the whole list.

aims at optimizing directly the target metrics. One option for training with AP is to design smooth upper bounds on the AP loss. First attempts were based on structural SVMs [165], [166]. Followed by extensions to speed up the "loss-augmented inference" [167] or to adapt to weak supervision [168]. Recently, a generic black box combinatorial solver has been introduced [169] and applied to AP optimization [68]. To overcome the brittleness of AP with respect to small cosine similarities variations, an *ad hoc* perturbation is applied to positive and negative scores during training. These methods provide elegant AP upper bounds, but are generally coarse AP approximations.

Other approaches rely on designing smooth approximations of the rank function. This is done in soft-binning techniques [67], [170]–[173] by using a smoothed discretization of similarity scores. [174] relies on explicitly approximating the non-differentiable rank functions using neural networks. They thus require using synthetic data to learn a DNN and do not have any guarantees on the learned function. Other methods, use a sum of sigmoid functions to approximate the Heaviside function of Eq. (2.5) in the Smooth-AP approach [69] or the more recent Smooth-Recall loss [70]. These approaches enable accurate surrogates by providing tight and smooth approximations of the rank function. However, they lose some theoretical properties, such as being upper bounds *e.g.* the rank approximation from [69] (illustrated on Fig. 2.7) uses the sigmoid function σ Fig. 2.8b to approximate the Heaviside function:



(a) Heaviside (step) function

(b) Sigmoid function used in [69]

Figure 2.8: Illustration of the Heaviside function and the sigmoid used in [69] to approximate it. The Heaviside function has gradient that are null or undefined. The sigmoid function has non-zero gradients in the neighbour of 0, and null outside.

$$\text{Smooth-rank}(k) = 1 + \sum_{j \in \Omega} \sigma\left(\frac{s_j - s_k}{\tau}\right), \quad \tau \in \mathbb{R} \tag{2.8}$$

$$\text{Smooth-AP} = \frac{1}{|\Omega^+|} \cdot \sum_{k \in \Omega^+} \frac{\text{Smooth-rank}^+(k)}{\text{Smooth-rank}(k)} \tag{2.9}$$

Using the sigmoid to approximate the Heaviside function and the rank had been introduced in [79]. However, it has several drawbacks, the resulting loss is not an upper bound on the true loss, and they can suffer from ill-behaved or vanishing gradients.

2.2.2 Non-decomposable losses.

Another difficulty when optimizing ranking metrics is that they are not decomposable, *i.e.* their value can not be computed over mini-batches and then averaged, as it would be the case for other metrics, *e.g.* the accuracy. This is illustrated on the toy example from Fig. 2.9, on each batch the average precision is 1 as the perfect ranking is achieved. However, the global average precision is not one, as the global ranking is not correct.

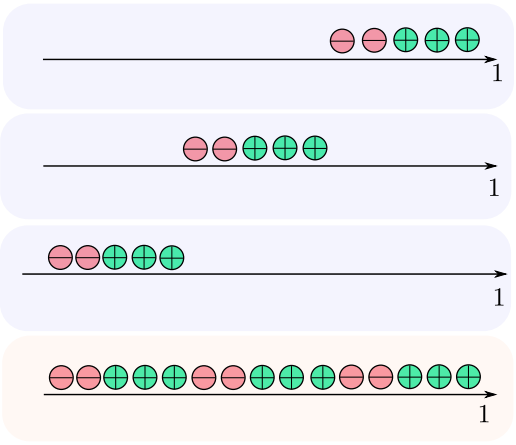


Figure 2.9: Illustration of the non-decomposability of ranking metrics. Each row represents retrieved instances in a batch ranked by their cosine similarity with a query, the last row represents the global ranking. On each batch the local ranking is perfect, *i.e.* the local ranking is optimal. However on the bottom the global ranking is not perfect, thus the global ranking is not optimal

Mini-batch training is mandatory in deep learning for computational efficiency and performances, although there has been works that try to use deep learning with non-stochastic gradient descent [175]. However, SGD assumes that the loss can be linearly decomposed between batches. Mini-batch training leads ranking metrics computed on batches to often over-estimate the global metric, leading to optimization issues. There have been several research directions in image retrieval to address this issue.

Non-decomposability is related to sampling informative constraints in simple ranking metrics surrogates, *e.g.* triplet losses, since the constraints' cardinality on the whole training set is prohibitive. This has been addressed by effective batch sampling [176]–[178] or selecting informative constraints within mini-batches [158], [178]–[180]. The difference between different "informative constraints" is illustrated on Fig. 2.10. Most of the triplets are "easy", *i.e.* the triplet loss constraint is already satisfied, thus they will not bring any information and will reduce gradient from informative triplets. Researcher have found it useful to select either "hard" or "semi-hard" triplets that do not respect the constraint and will be informative. In cross-batch memory technique [73], the authors assume a slow drift in learned representations to store them and compute global mining in pair-based deep metric learning.

In AP optimization, the non-decomposability has essentially been addressed by a brute force increase of the batch size [67], [70], [169], [173], which is also a hard problem [181]. This includes an important overhead in computation and memory, generally involving a two-step approach for first computing the AP loss and subsequently re-computing activations and back-propagating

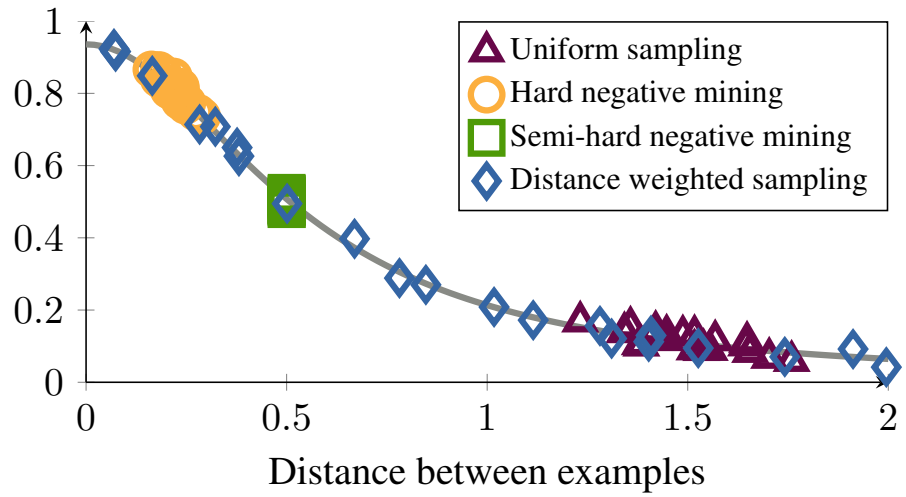


Figure 2.10: Image taken from [65]. Hard batch mining consist in finding tuplets (*e.g.* triplets) that are not respecting the constraints of the loss (*e.g.* triplet loss). It helps convergence and achieving better performances. Authors of [65] define several types of hard negatives based on their distance with the query and the margin used in the triplet losses, see Eq. (2.7). They then bias triplet sampling towards "semi-hard" negatives, *i.e.* negatives that do not respect the triplet constraints but have a lower similarity with the query than the positive.

gradients.

2.3 Hierarchical learning for robust retrieval.

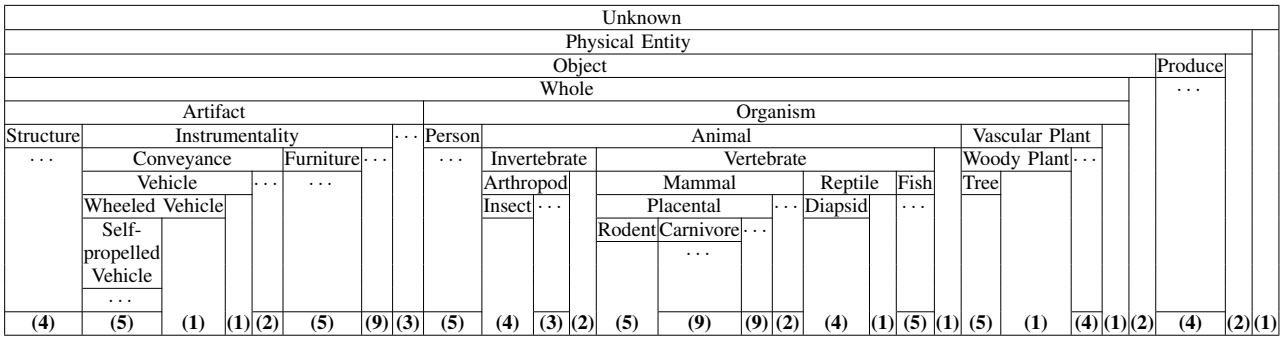


Figure 2.11: Image taken from [182]. Hierarchical labels of the CIFAR-100 [50] dataset.

In standard task in deep learning such as classification, object detection or image retrieval, the goal is to correctly predict the fine-grained classes of images. For instance, on CIFAR-100 [50] the goal is to classify an image among the 100 fine-grained categories, *e.g. dolphin, cloud, motorcycle etc.*. These tasks in their standard definition do not take into account what

are the predictions of the models when they fail to recognize the fine-grained classes. We refer to this as the mistake severity, *i.e.* how bad an error is. The notion of "how bad" is hard to quantify. An interesting direction is to consider human preferences and measure how severe humans will consider a particular error. Because human studies take time and would be impossible to scale in order to evaluate the ever-growing number of deep learning research papers, an interesting proxy for human preferences that has emerged is using class hierarchies, illustrated for CIFAR-100 on Fig. 2.11. We will also discuss in Sec. 2.3.2 the optimization in information retrieval based on relevance that can be based on human ratings. There also has been progress in the field of NLP for "human alignment" based on the recent method *Reinforcement Learning with Human Feedbacks* (RLHF) introduced in [183] and subsequently used in [2], [3], [5] that is based human preferences and optimize using reinforcement learning (RL), for instance with PPO [184]. Some works are also paving the way in computer vision, by using RL to optimize target metrics in [185] or using RLHF to fine-tune Stable Diffusion [186].

2.3.1 Hierarchical classification.

The first application of hierarchical modeling stem from NLP research [187], [188] and was mostly motivated for efficiency at training and test time. Hierarchical classification also has a long history in computer vision. Originally, researchers aspired to enhance classification performance by hierarchically organizing classes to enable more efficient and accurate predictions [189]–[192]. The hope was that such an approach would leverage the inherent structure and relationships within the data, leading to improved results. However, modern deep neural networks have mostly used the standard cross-entropy [44]–[46] as classifying at the fine-grained level is the ultimate task of classification. Involving hierarchical information can confuse the models and prevent them from learning sufficient discriminative features [193].

There has been a recent regain of interest in hierarchical classification [75], [194]–[196], with the motivation of learning robust models by lowering their mistake severity Eq. (2.10). As the authors of [75] phrase it, to make "better mistakes". Optimizing the mistake severity is based on lowering the LCA, "lowest common ancestor". It is the distance between two labels in the hierarchical tree. A lower LCA means that the labels are closely related, a greater LCA means that the labels are further away in the hierarchical tree. A robust model, *i.e.* that minimizes the mistake severity (MS) Eq. (2.10), minimizes the LCA between predictions and the ground truths (gt) when it misclassifies instances.

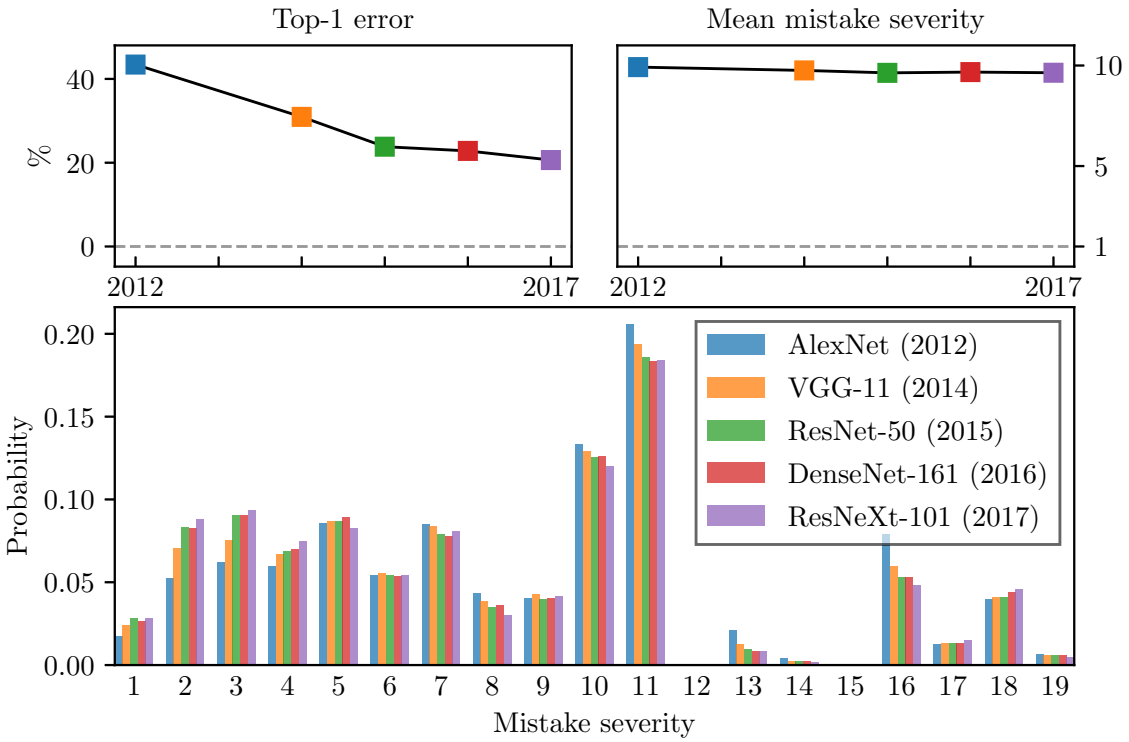


Figure 2.12: Image taken from [75]. Although predictive performance of models keep rising, their mistake severity has stagnated over the years.

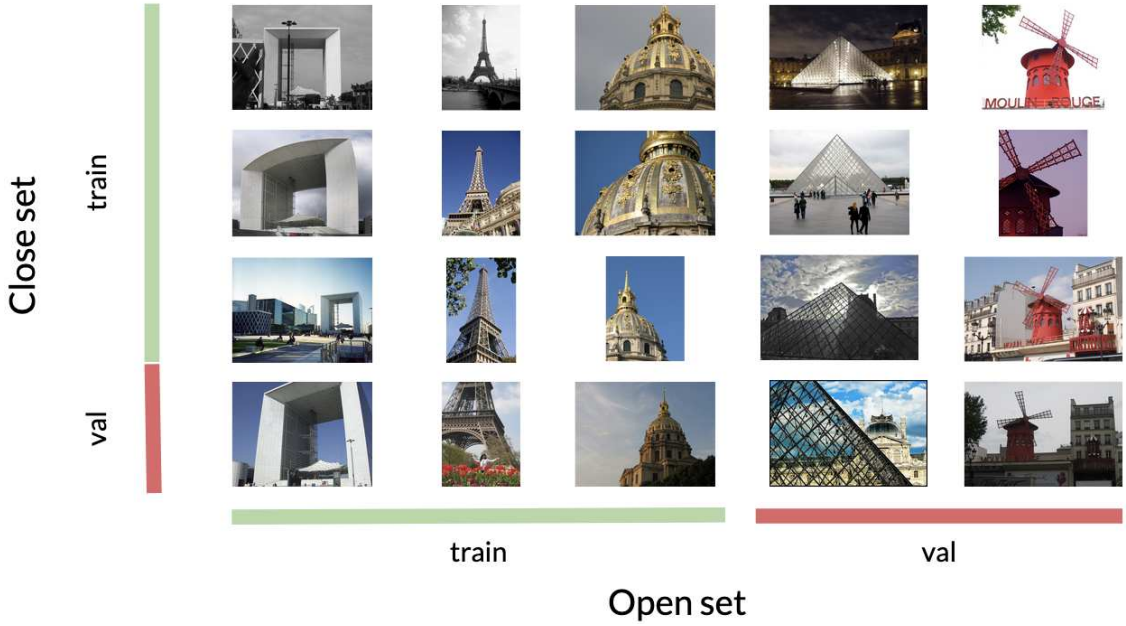


Figure 2.13: In closed set, the images from the training and test sets are from the same classes. In open set, the training and test classes are disjoint.

$$\text{MS} = \mathbb{E}_{\text{misclassified}} \left[\text{LCA}(y_{\text{pred}}, y_{\text{gt}}) \right] \tag{2.10}$$

This type of robustness was shown to be not achieved by solely having better fine-grained accuracy: [75] showed that, while model have improved over the years on fine-grained classification, their mistake severity has however remained the same, as shown in Fig. 2.12. However, hierarchical classification is evaluated in *closed set*, *i.e.* train and test classes are the same. Whereas, image retrieval considers the *open set* paradigm, where classes are distinct between train and test sets to better evaluate the generalization abilities of learned models. This difference of evaluation is illustrated on Fig. 2.13, limiting their application in image retrieval. Furthermore, recent successful hierarchical classification methods rely on using several classifiers [195], [196], whereas standard image retrieval setting mostly rely on learning a single embedding for the image. Using several embedding would strongly impact the already costly querying cost.

2.3.2 Graded predictions.

The Information Retrieval community uses datasets where documents can be more or less relevant depending on the query [197], [198]. As seen on Fig. 2.14, each movie in the database is assigned a score based on the movies a user has seen. The quality of their retrieval engine is quantified using ranking based metrics such as the NDCG [143], [144].

Figure 2.14: Based on the movies the query user has seen, each movie in the database has a relevance assigned to it.

The NDCG can accommodate multiple level of similarity between a query and a retrieve instance. This is done via a relevance function, denoted as "*rel*" that indicates how similar a

pair of instances are. The NDCG is defined as follows:

$$\begin{aligned}\text{DCG} &= \sum_{k\in\Omega^+} \frac{\text{rel}(k)}{\log_2(1+\text{rank}(k))} \\ \text{iDCG} &= \max_{\text{rank}} \text{DCG} \\ \text{NDCG} &= \frac{\text{DCG}}{\text{iDCG}}\end{aligned} \tag{2.11}$$

Similarly to metrics introduced previously, the NDCG is also defined using the ranking function. Several works have investigated how to optimize the NDCG, *e.g.* using pairwise losses [199] or smooth surrogates [79], [200]–[202]. These works however focused on NDCG, and are without any theoretical guarantees: the surrogates are approximations of the NDCG but not *lower bounds*, *i.e.* their maximization does not imply improved performances during inference. An additional drawback is that NDCG does not relate easily to average precision [203], the most common metric in image retrieval. Fortunately, there have been some works done to extend AP in a graded setting where relevance between instances is not binary [80], [204]. The graded Average Precision from [80] is the closest to our work as it leverages SoftRank for direct optimization of non-binary relevance, although there are significant shortcomings. There is no guarantee that the SoftRank surrogate actually minimizes the graded AP. In addition, it requires annotating datasets with pairwise relevances, which is impractical for large scale image retrieval settings.

2.3.3 Hierarchical image retrieval.

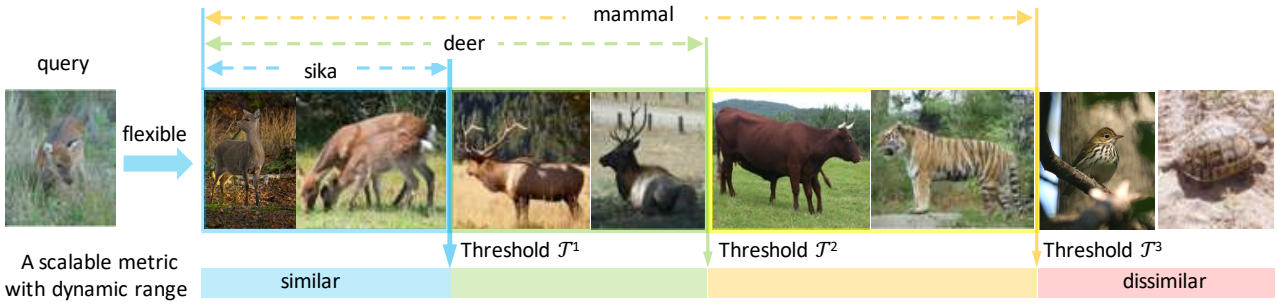


Figure 2.15: Image taken from [81]. Hierarchical image retrieval as two main goals. 1) reduce mistake severity, *i.e.* it is better to retrieve a "deer" than a "turtle" for a "sika" query. 2) When given a "sika" query, first images should be "sika", then "deer" and finally "mammal".

Hierarchical image retrieval is a complicated task as it involves learning sufficiently discriminative features to be able to retrieve the correct fine-grained instances while also ensuring

that fine-grained classes from the same super categories closer in the embedding space than fine-grained categories from other super categories. On Fig. 2.15 this would mean that when querying an image of a "sika" the closest images should be of other "sika", *i.e.* sufficiently discriminative from other "deer". But the image query of a "sika" should be closer to other "deer" than to different "mammal". As discussed previously in the standard image retrieval setup, images from the same super categories often are considered "hard negatives" [65], [173] and models are optimized to discriminated strongly images from the same super categories. Even works that consider non-binary losses, *e.g.* [148], [205], were observed in [122] to still make severe mistakes.. However, for application in search engine it is important to have two robustness properties:

1. Errors should not be too severe, *i.e.* it is better to have "deer" false positives on Fig. 2.15 than "birds" false positives.

2. After the fine-grained instances are retrieved, the results should somewhat still be relevant for the query, *i.e.* after all "sika" images are retrieved on Fig. 2.15 images of "deer" then "mammal" should come after.

Recently, the authors of [81] introduced three new hierarchical benchmarks datasets, DyML (DyML-Animal is illustrated on Fig. 2.15), for hierarchical image retrieval. Building on the idea that models should have strong performances on fine-grained, but also consider the super categories. Researchers have subsequently designed losses that extend proxy-based losses to the hierarchical setting [81], [206], [207]. In another fashion, [82] introduces the CLCD loss that uses a "cross distillation" loss, a type of pair loss at different level of semantics. These works extend standard proxy-losses to the hierarchical setting and try to structure the embedding space in a hierarchical manner. However, these methods face the same limitations as the usual proxy losses: minimizing them do not explicitly optimize a well-behaved hierarchical evaluation metric, *e.g.* NDCG or $\mathcal{H}$-AP introduced in Chapter 4.

2.4 Out-of-distribution detection.

Figure 2.16: Image from @_realrusty / ViralHog.com. A Tesla's Autopilot system appeared to confuse a horse-drawn carriage, which is not present in the training set, for a truck.

Out-of-distribution (OOD) detection is a major safety requirement for deep neural networks deployment in real world settings. OOD detection is another aspect of DNN robustness. In standard classification evaluation, the images at test time are assumed to be coming from the same distribution as the training set. However, in real scenarios some object at test time could not come from the in-distribution, *i.e.* the training set. This is illustrated by Fig. 2.16, in which an autonomous system misclassifies a horse-drawn carriage as a truck. It was shown in [84] that DNN suffer from overconfidence, which makes it hard to distinguish when a model is making a mistake.

OOD detection is formulated as a binary classification problem. At inference time, an OOD detection "scorer" must decide whether an image x is from the in distribution (ID) or not (OOD):

$$G_\lambda(x) = \begin{cases} \text{ID}, & \text{if } E(x) \leq \lambda \\ \text{OOD}, & \text{if } E(x) > \lambda \end{cases} \tag{2.12}$$

where samples with low energies $E(x)$ are classified as ID and vice versa, and λ is a threshold. The threshold λ is typically chosen so that a high fraction of ID data (*e.g.* 95%) is correctly classified.

2.4.1 Post-hoc out-of-distribution detection.

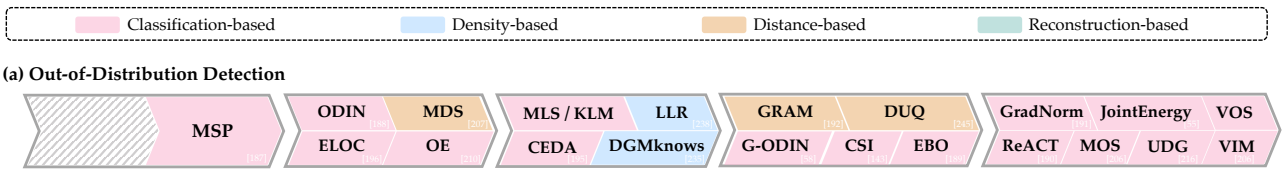


Figure 2.17: Image taken from [208]. Different families of out-of-distribution detection methods.

Seminal attempts for OOD detection used supervised methods based on external OOD samples [86], [87] or "Outlier Exposure" (OE) [209] enforcing a uniform OOD distribution. Although OOD datasets can improve OOD detection, their relevance is questionable since

collecting representative OOD datasets is arguably impossible as OOD lie anywhere outside the training distribution [210]. It can also have the undesirable effect of learning detectors biased towards certain types of OOD [211].

Another drawback from OE is that it requires training the predictive model, which limits the adoption of strong pre-trained backbones or "zero-shot" foundation models. This can also be prohibitive when using large neural networks on large amounts of data. To better address these issues, there has been a growing popularity amongst the OOD detection community for "post-hoc" approaches. This type of approaches uses pre-trained backbones, *e.g.* ResNet-50 [45] on ImageNet, and exploits aspects of these backbones to detect OOD samples. This rich literature was classified in [208] into several families of OOD detection method, and illustrated on Fig. 2.17.

First methods were based on the classification from a pre-trained DNN. For instance, the well-known maximum class probability of [85] uses the maximum softmax probability of the classifier as the score used in Eq. (2.12). Other classification based methods include Odin [212], energy-logits [94] or DICE [92]. Other type of methods rely on distances in the feature space. [91] and [89] use the Mahalanobis distance, [90] relies on the kNN distance, or [213] relies on Gram matrix computed from layer activations.

Another direction is using reconstruction approaches. For instance in [214] authors leverage masked-autoencoders [9] or in DiffGuard [215] authors use diffusion models, they both compute the reconstruction error of an image. However, they require a second neural network, *i.e.* the decoder in [214] or a diffusion model in [215], which can entail a large computational overhead in the case of diffusion models.

The last family of methods identified in [208] are density based methods, *e.g.* [88] that computes the likelihood of an image using autoencoder networks. Note that some methods can be classified in more than one family. For instance, [94] proposes an energy score derived from the `logsumexp` of the logits of the classifier, which is closely related to the density. Similarly, the class conditional GMM of [91] and [89] approximate the ID features by making a Gaussian assumption for the true ID distribution. Finally, authors of [90] argue that the kNN distribution approximates well the ID features distribution, which allows efficient OOD detection as illustrated on Fig. 2.18. Although these methods are efficient they lack expressiveness, *e.g.* whereas using a GMM to estimate the density as in [89], [91] is a good approximation it can not fully model the ID distribution with its rigid Gaussian assumption.

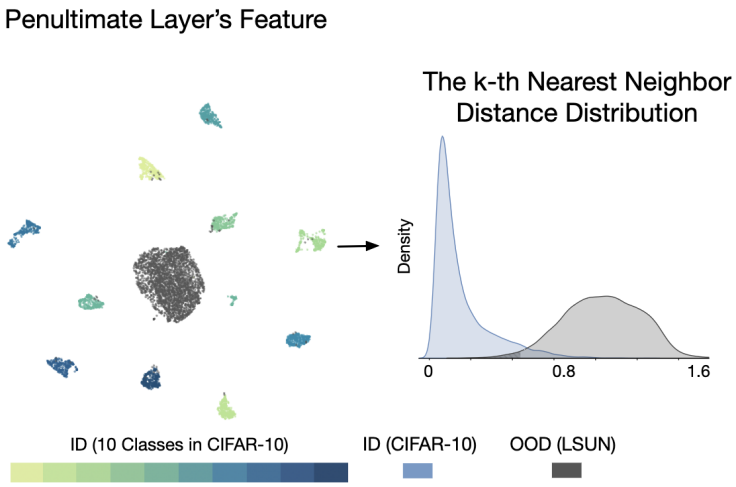


Figure 2.18: Image adapted from [90]. The kNN distribution allows to approximate the ID features distribution, thus allowing OOD detection.

2.4.2 Energy-based models.

EBMs [93] are another approach to estimate the ID density. They are unnormalized density model defined via an energy function $E_\theta : \mathbb{R}^m \to \mathbb{R}$ parameterized by a neural network with parameters θ. For $\mathbf{z} \in \mathbb{R}^m$, the probability density is given by the Boltzmann distribution

$$p_\theta(\mathbf{z}) = \frac{1}{Z_\theta} \exp\left(-E_\theta(\mathbf{z})\right), \tag{2.13}$$

where Z_θ is the partition function which is intractable in high dimension. EBMs are trained via maximum likelihood estimation (MLE), which amounts to perform stochastic gradient descent with the following loss:

$$\mathcal{L}_{\text{MLE}} = \mathbb{E}_{\mathbf{z}\sim p_{\text{in}}}\Big[E_\theta(\mathbf{z})\Big] - \mathbb{E}_{\mathbf{z}'\sim p_\theta}\Big[E_\theta(\mathbf{z}')\Big]. \tag{2.14}$$

This builds upon the fact that $\nabla_\theta(-\log p_{\theta_i}(\mathbf{z}))$ can be computed without computing the intractable normalization constant Z_θ (see Appendix C.1 for more details). The $\mathcal{L}_{\text{MLE}}$ is illustrated on Fig. 2.19, the energy of real data p_{in} is minimized, while the energy of generated data is maximized p_θ. To synthesize examples from p_θ, one can use gradient-based MCMC sampling, such as Stochastic Gradient Langevin Dynamics (SGLD) [216] or Hamiltonian Monte Carlo (HMC) [217]. We will focus in this thesis on SGLD, following recent success of SGLD for EBMs [218], [219].

EBMs are flexible, and do not require dedicated architectures, contrary to *e.g.* Normalizing Flows [220]. They have made incredible progress in generative modeling for images in recent

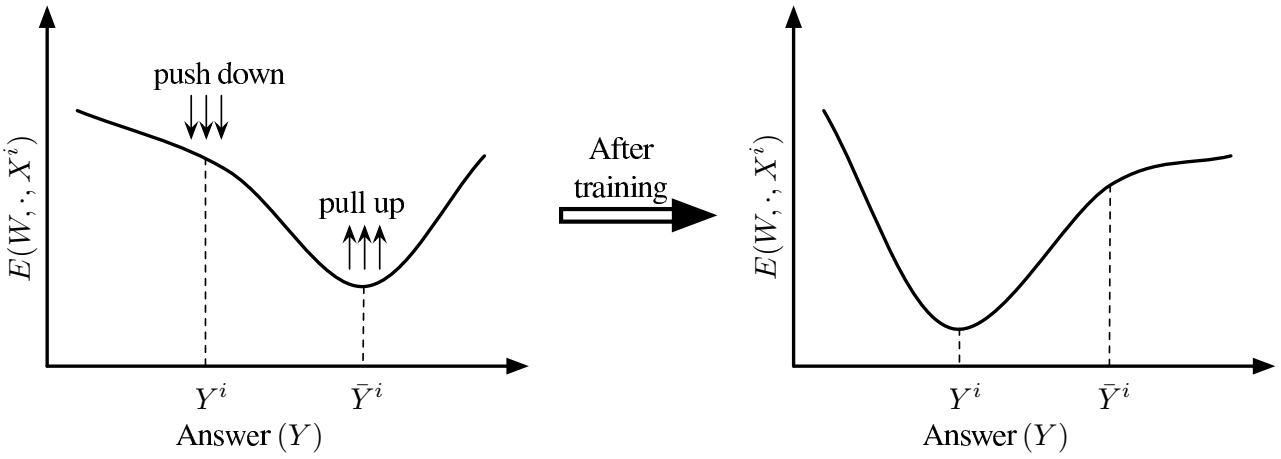


Figure 2.19: Image taken from [93]. The effect of training on the energy surface as a function of the answer Y in the continuous case. After training, the energy of the correct answer Y is lower than that of incorrect answers.

years [218], [219], [221]. However, their performances for OOD detection are not yet comparable with OOD methods based on the feature space [222]. [94] have proposed to perform OOD detection with an energy score defined by the *logsumexp* of the logits (EL) of the pre-trained classifier, showing improvement over using the classifier's predicted probabilities [85]. Furthermore, the authors of EL propose to fine-tune the logits of the classifier using external OOD datasets. However, this approach can again suffer from biases from the choice of the external OOD datasets.

2.4.3 Residual learning.

Training hybrid models, where a data-driven *residual* complements an approximate predictor, has been proposed in several contexts, *e.g.* in complex dynamic forecasting [224], in NLP [225], in video prediction [226], [227], or in robotics [223]. Different types of learning paradigm are illustrated on Fig. 2.20. [223] proposes to combine physic laws and a learned residual model for robotics.

Energy-based models have also been used to learn a correction of a reference model $q(\mathbf{z})$:

$$p_\theta^h(\mathbf{z}) = \frac{1}{Z(\theta)} p_\theta^r(\mathbf{z}) q(\mathbf{z}), \tag{2.15}$$

with $Z(\theta) = \int p_\theta^r(\mathbf{z}) q(\mathbf{z}) \mathrm{d}\mathbf{z}$ the normalization constant. The residual density $p_\theta^r(\mathbf{z})$ is learned with an EBM: $p_\theta^r(\mathbf{z}) \propto \exp\left(-E_\theta(\mathbf{z})\right)$.

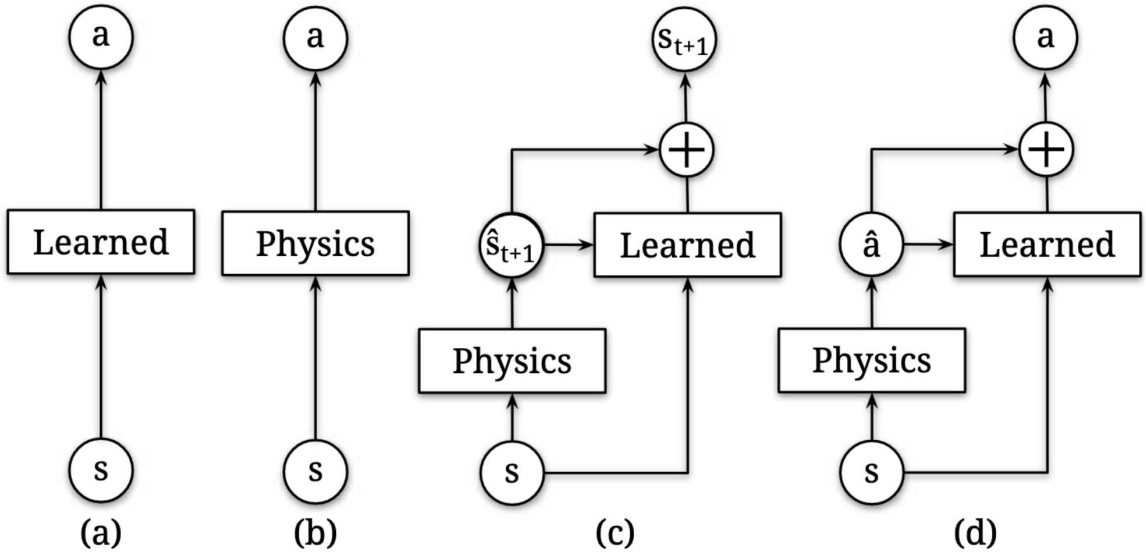


Figure 2.20: Image taken from [223]. (a) depicts a standard machine learning model, *i.e.* learned. (b) uses a prior, *e.g.* physics laws, to make prediction, (c) & (d) illustrate a residual approach by combining a prior and learned model.

This idea has been explored in noise contrastive estimation (NCE) [228] where the correction is obtained by discriminative learning. Learning an EBM in cooperation with a generator model has been introduced in [229] where an EBM learns to refine generated samples and has also been applied to cooperative learning of an EBM with a conditional generator [230], a VAE [231]–[233] a normalizing flow [234], [235]. However, these methods were not designed for OOD detection, as they focus on generation and cannot benefit from a fixed prior OOD detector as they use a cooperative learning strategy.

Such residual approaches have also emerged for OOD detection. ResFlow [236] uses a normalizing flow (NF) to learn the residual of a Gaussian density for OOD detection. However, NFs require invertible mapping, which intrinsically limit their expressive power and make the learned residual less accurate, whereas EBMs are more flexible and can be implemented using a simple multi-layer perceptron. Also, ViM [211] proposes to model the residual of the ID density by using the complement of a linear manifold on the ID manifold. This linear residual however limits the expressiveness of the method, and leads to lower experimental performances.

2.4.4 Ensembling & composition.

The question of merging several networks, also known as *ensembling* [238] has been among the first and most successful approaches for OOD detection. The ensemble can include different

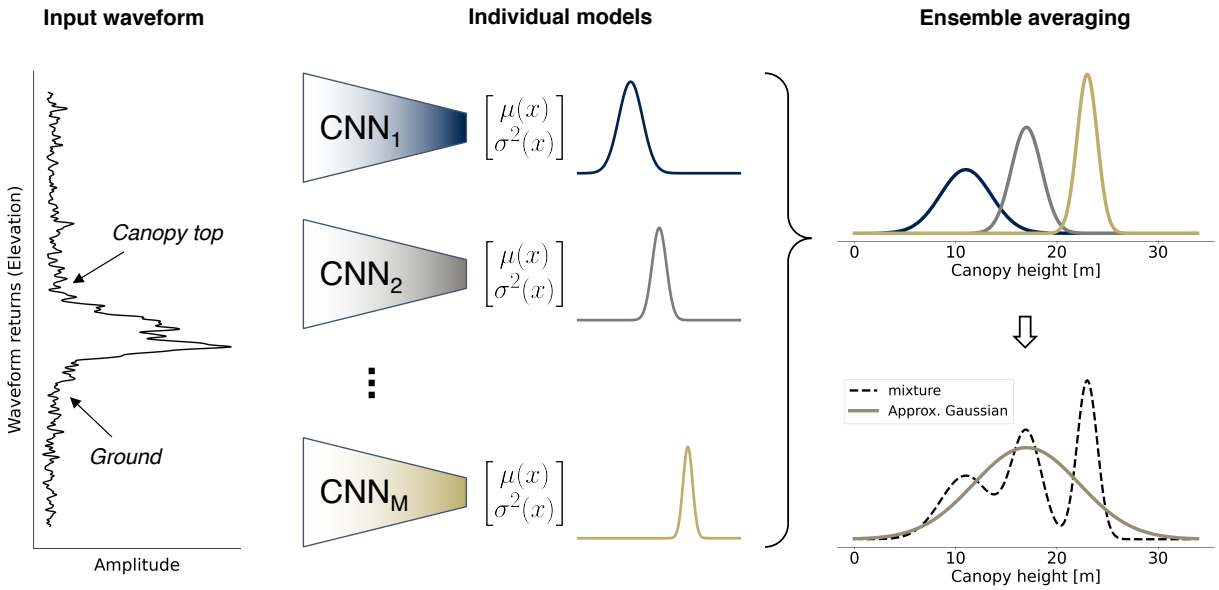


Figure 2.21: Image taken from [237]. In deep ensembles, multiple DNN are used to make the final predictions. It can also be a good method for OOD detection.

backbones or different training variants. For OOD detection, several post-hoc approaches also model the ID density at different layer depth of a pre-trained model, the overall density score being obtained by ensembling such predictions [91], [236], [239]. The main limitation of these approaches is their computational cost since the inference time is proportional to the number of networks, although other method for deep ensembles [240] would be worth exploring. The overhead quickly becomes prohibitive in contexts with limited resources. Several sources of prior densities are combined in [211] to refine OOD detection. This idea can be integrated in a principled framework such as the EBM composition model [241], [242]. Allowing to include several hybrid energy terms to refine ID density estimation, with limited computational overhead at inference time.

Chapter 3

Optimization of Ranking Losses for Image retrieval

As discussed previously, standard evaluation metrics in image retrieval rely on score ranking, *e.g.* average precision (AP), recall at k (R@k), normalized discounted cumulative gain (NDCG). In this chapter, we address the two major challenges presented in Chapter 2 for end-to-end training of deep neural networks with rank losses: non-differentiability and non-decomposability, by introducing a general framework for robust and decomposable rank losses optimization. Firstly, we propose a general surrogate for ranking operator, SupRank, that is amenable to stochastic gradient descent. It provides an upper-bound for rank losses, which guarantees that the target loss is optimized, and we show that it improves gradient flow compared to the sigmoid approximation discussed in Sec. 2.2.1. Secondly, we define the decomposability gap, DG, which is the gap between the averaged batch approximation of ranking losses and their values on the whole training set. We then propose a simple yet effective loss function to reduce it. We give theoretical analysis as to why this proposed loss will help reduce non-decomposability. We apply our framework to two standard metrics for image retrieval: AP and R@k and show the experimental gains brought by our framework compared to previous surrogate losses. Code is released at `github.com/elias-ramzi/ROADMAP`.

Content

3.1 Introduction.

As discussed in Sec. 2.2, performances of image retrieval systems are often measured using list-wise or ranking-based metrics, *e.g.* average precision (AP), recall rate at k (R@k), or Normalized Discounted Cumulative Gain (NDCG). These metrics penalize retrieving non-relevant images before other remaining relevant images and are used in several tasks implying a large imbalance between positive and negative samples. For instance, AP is also the *de facto* metric used in several computer vision tasks, *e.g.* object detection or long-tailed classifications.

Although these metrics are suited to evaluate image retrieval, their use for training deep neural networks is limited. They have two main drawbacks: i) they are not amenable to stochastic gradient descent (SGD) and thus cannot be used directly to train deep neural networks (DNN), ii) they are not decomposable as they do not decompose linearly between samples.

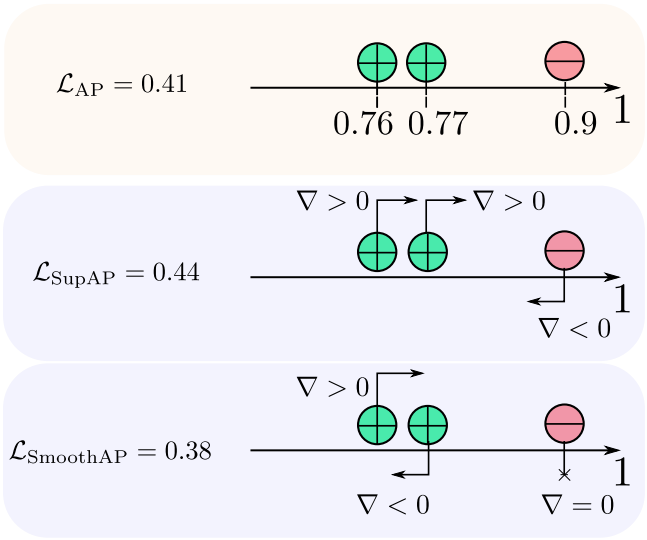


Figure 3.1: We define $\mathcal{L}_{\text{SupAP}}$ using SupRank. $\mathcal{L}_{\text{SupAP}} \geq \mathcal{L}_{\text{AP}}$ and $\nabla \mathcal{L}_{\text{SupAP}} > 0$ in this example, in contrast to SmoothAP [69]. This ensures robust training and comes from a new approximation of the rank function, SupRank.

Designing surrogate losses. Because rank losses are not directly amenable to gradient descent, there is a rich literature dedicated to designing appropriate surrogate losses that we discussed in Chapter 2. Most historical methods rely on n-uplet losses [65], [117], [118], [155]–[160] to optimize local rankings. Another family of losses are classification based losses [66], [161]–[164], [243] that reduce the number of comparisons required during training compared to n-uplet losses. Because n-uplet losses and classification losses do not directly relate to the

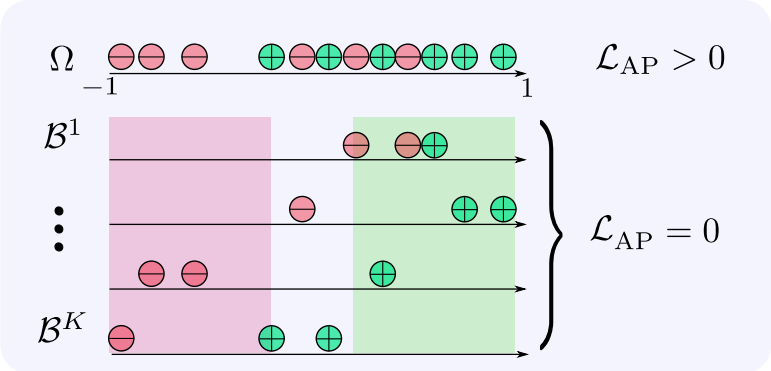


Figure 3.2: $\mathcal{L}_{\text{AP}}$ non-decomposability: $\mathcal{L}_{\text{AP}} = 0$ in all batches $\mathcal{B}^i$ despite $\mathcal{L}_{\text{AP}} \neq 0$ over the whole $\bigcup_i \mathcal{B}^i$.

evaluation metrics, there also has been extensive work to create list-wise losses amenable to gradient descent [165]–[168] that create coarse upper bounds of the target metric. Or tighter approximations thanks to fine approximation of the rank [67], [69], [70], [169]–[174] but loosen theoretical properties, *e.g.* being upper bounds on the true rank losses, guaranteeing that their optimization will increase the performances. The work presented in this chapter relates to recent approximations that use the sigmoid to approximate the rank to provide accurate surrogate losses for the AP in [69] or R@k in [70]. In this work, we show some of the shortcomings of this approximation, illustrated on Fig. 3.1 for $\mathcal{L}_{\text{SmoothAP}}$ of [69]. The sigmoid approximation results in surrogate losses that are not upper-bounds of the true AP loss. For example, on Fig. 3.1, we can see that $\mathcal{L}_{\text{AP}} = 0.41$ whereas $\mathcal{L}_{\text{SmoothAP}} = 0.38$, meaning that optimizing $\mathcal{L}_{\text{SmoothAP}}$ will not necessarily result in a lower $\mathcal{L}_{\text{AP}}$. This approximation also lacks good gradient flow, as shown in Fig. 3.1, $\mathcal{L}_{\text{SmoothAP}}$ produces gradients that tend to decrease the similarity of a positive instance, which is not a desirable behavior. This is further discussed in Sec. 3.4.1.

Addressing non-decomposability. Mini-batch training is mandatory in deep learning, both due to computational constraints and because SGD improves generalization [244]. However, it assumes that the loss functions decomposes linearly between samples, as discussed in Chapter 2, which is not the case for rank losses. Thus, during rank loss training, the loss averaged over batches generally underestimates its value on the whole training dataset, which we refer to as the *decomposability gap*, *DG*. This is illustrated on the toy example of Fig. 3.2, on each batch $\mathcal{L}_{\text{AP}} = 0$ as the optimal ranking is achieved, however on the whole set Ω we can see that $\mathcal{L}_{\text{AP}} \neq 0$. As discussed in Sec. 2.2.2 attempts in image retrieval to circumvent the problem involve *ad hoc* methods based on hard batch sampling strategies [65], [158], [177], [178], storing all training representations/scores [68], [73] or using larger batches [67], [70], [173], leading to

complex models with a large computational or memory overhead.

The core of our approach is a unified framework to optimize rank losses for image retrieval. We will see in Chapter 4 an extension to hierarchical image retrieval. Specifically, our contributions are:

- To propose a smooth approximation of the rank in Sec. 3.2.2, SupRank, that overcomes the theoretical shortcomings discussed previously. SupRank is amenable to SGD and is an upper bound on the true rank, with a different behavior for positive and negative examples. This design leads to smooth losses that are upper bounds of the true losses, and always back-propagates gradients when the correct ranking is not satisfied as detailed in Sec. 3.4.

- To define the decomposability gap in Sec. 3.2.3. We use an additional objective at training time to reduce DG, and thus the non-decomposability of smooth rank losses, without the need to increase the batch size. It is a simple yet effective training objective $\mathcal{L}_{\text{DG}}$, which calibrates the scores among different batches by controlling the absolute value of positive and negative samples. We provide in Sec. 3.4.2 a theoretical analysis showing that $\mathcal{L}_{\text{DG}}$ decreases the decomposability gap.

- To apply this framework to two popular metrics for image retrieval: AP in Sec. 3.3.1 and R@k in Sec. 3.3.2. The resulting surrogates losses are both upper bounds of their target metric.

- To provide a thorough experimental validation including three standard image retrieval datasets and show that optimizing AP with our framework outperforms state-of-the-art methods. We also report the large and consistent gain compared to rank approximation baselines on both AP and R@k, and we highlight in the ablation studies the importance of our two components.

3.2 Robust and decomposable rank losses.

We present in this section our framework for RObust and Decomposable (ROD) dedicated to direct optimization of rank losses with stochastic gradient descent (SGD), see Fig. 3.3.

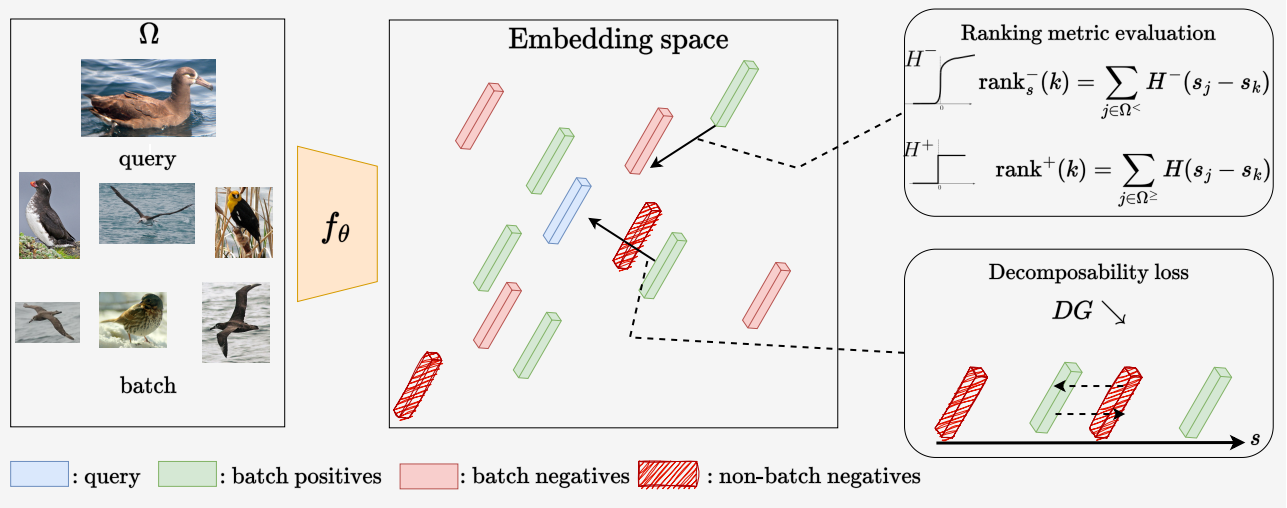


Figure 3.3: Illustration of our unified framework. We use a deep neural network f_θ to embed images. We then optimize its weights in an end-to-end manner using two losses: 1) we optimize the ranking-based evaluation metric using an upper bound approximation of the rank, rank_s^-, enforcing the batch's positive embeddings to have higher cosine similarity with the query than the batch's negatives; 2) we reduce the decomposability gap, DG, of rank losses using a decomposability loss, that supports that positives have higher similarity with the query than all negatives even outside the batch.

3.2.1 Preliminaries.

Let us consider a retrieval set $\Omega = \{\boldsymbol{x_j}\}_{j \in [\![1;N]\!]}$ composed of N elements, and a set of M queries $\mathcal{Q}$. For each query $\boldsymbol{q_i}$, each element in Ω is assigned a relevance [197] $\text{rel}(\boldsymbol{x_j}, \boldsymbol{q_i}) \in \{0, 1\}$, such that $\text{rel}(\boldsymbol{x_j}, \boldsymbol{q_i}) = 1$ (resp. $\text{rel}(\boldsymbol{x_j}, \boldsymbol{q_i}) = 0$) if $\boldsymbol{x_j}$ is relevant (resp. irrelevant) with respect to $\boldsymbol{q_i}$, *i.e.* if x_j and q_i share the same fine-grained label. In the next chapter Chapter 4, we will see that our framework can be extended to hierarchical image retrieval setting discussed in Sec. 2.3 by modeling more complex pairwise relevance with $\text{rel}(\boldsymbol{x_j}, \boldsymbol{q_i})$. Positive relevance defines the set of positives for a query, *i.e.* $\Omega_i^+ := \{\boldsymbol{x_j} \in \Omega | \, \text{rel}(\boldsymbol{x_j}, \boldsymbol{q_i}) = 1\}$. Instances with a relevance of 0 are the negatives, *i.e.* $\Omega_i^- := \{\boldsymbol{x_j} \in \Omega | \, \text{rel}(\boldsymbol{x_j}, \boldsymbol{q_i}) = 0\}$.

For each $\boldsymbol{x_j} \in \Omega$, we compute its embedding $\mathbf{v_j} \in \mathbb{R}^d$. To do so, we use a neural network $f_{\boldsymbol{\theta}}$ parameterized by $\boldsymbol{\theta}$: $\mathbf{v_j} := f_{\boldsymbol{\theta}}(\boldsymbol{x_j})$. In the embedding space $\mathbb{R}^d$, we compute the cosine similarity score between each query $\boldsymbol{q_i}$ and each element in Ω: $s(\boldsymbol{q_i}, \boldsymbol{x_j}) = {\mathbf{v_{q_i}}}^T \mathbf{v_j} / ||\mathbf{v_{q_i}}|| \cdot ||\mathbf{v_j}||$.

During training, our goal is to optimize, for each query $\boldsymbol{q_i}$, the model parameters $\boldsymbol{\theta}$ such that the ranking, *i.e.* decreasing order of cosine similarity, matches the ground truth ranking, *i.e.* decreasing order of relevances. More precisely, we optimize a ranking-based metric $0 \leq \mathcal{M}_i \leq 1$ that penalizes inversion between positive instances and negative ones, $\mathcal{M}_i = 1$ meaning

achieving the best ranking. The target loss is averaged over all queries:

$$\mathcal{L}_{\mathcal{M}}(\boldsymbol{\theta}) = 1 - \frac{1}{M}\sum_{i=1}^{M}\mathcal{M}_i(\boldsymbol{\theta}, \Omega_i) \tag{3.1}$$

As previously mentioned, there are two main challenges with SGD optimization of rank losses: i) they are not differentiable with respect to $\boldsymbol{\theta}$ which comes from the ranking operator, and ii) they do not linearly decompose into batches indeed for each query $\mathcal{L}_{\mathcal{M}}$ needs to be computed on the whole set Ω_i as seen on Eq. (3.1). We propose to address both issues: we introduce a robust differentiable ranking surrogate, SupRank (Sec. 3.2.2), and add a decomposable objective (Sec. 3.2.3) to improve rank losses' behavior in a batch setting. Our final **RO**bust and **D**ecomposable (ROD) loss $\mathcal{L}_{\text{ROD-}\mathcal{M}}$ combines a differentiable surrogate loss of a target metric ,$\mathcal{L}_{\text{Sup-}\mathcal{M}}$, which is an upper bound meaning that optimizing the surrogate results in optimizing the target metric; and the decomposable objective $\mathcal{L}_{\text{DG}}$ which allows optimizing the loss until the targeted global ranking is achieved; with a linear combination weighted by the hyperparameter λ:

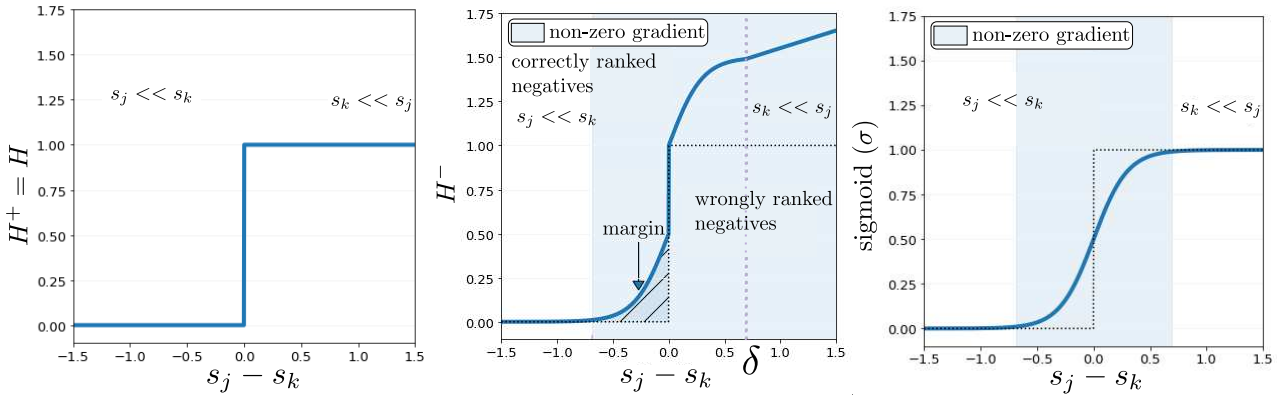
$$\mathcal{L}_{\text{ROD-}\mathcal{M}}(\boldsymbol{\theta}) = (1-\lambda)\cdot\mathcal{L}_{\text{Sup-}\mathcal{M}}(\boldsymbol{\theta}) + \lambda\cdot\mathcal{L}_{\text{DG}}(\boldsymbol{\theta}) \tag{3.2}$$

Our unified framework for end-to-end training of DNN is illustrated in Fig. 3.3. Using f_θ we encode both the query q and the rest of the images in the batch Ω. Optimizing the rank loss supports the correct –partial– ordering in a batch based on our surrogate of the rank, SupRank. Optimizing the decomposability loss supports that the positives will be ranked even before negative items that are not present in the batch. Both losses are amenable to gradient descent, which makes possible to update the model parameters with SGD.

3.2.2 Robustness in smooth rank approximation.

The non-differentiability in rank losses comes from the ranking operator, which can be viewed as counting the number of instances that have a similarity score greater than the considered instance: $rank(k) = 1 + \sum_{j\in\Omega} H(s_j - s_k)$, as seen earlier in Eq. (2.5). For the sake of readability, we drop in this section the dependence on the query, *i.e.* dependence with i. In this chapter, we propose to rewrite the rank, which will be motivated in Sec. 3.2.2.1:

$$\text{rank}(k) = \underbrace{1 + \sum_{j\in\Omega_k^+} H(s_j - s_k)}_{\text{rank}^+(k)} + \underbrace{\sum_{j\in\Omega_k^-} H(s_j - s_k)}_{\text{rank}^-(k)} \tag{3.3}$$



(a) $H^+(x) = H(x)$ in Eq. (3.3) (b) $H^-(x)$ in Eq. (3.4) (c) Sigmoid used in [69]

Figure 3.4: Proposed surrogate losses for the Heaviside (step): with $H^+(x)$ in Fig. 3.4a and $H^-(x)$ in Fig. 3.4b. Using H^- in Eq. (3.5) leads to smooth, and upper bounds rank losses. In addition, $H^-(x)$ back-propagates gradients until the correct ranking is satisfied, in contrast to the sigmoid used in [69] (Fig. 3.4c).

where H is the Heaviside (step) function $H(t) = 1$ if $t \geq 0$, 0 otherwise. Note that for both $\text{rank}^+(k)$ and $\text{rank}^-(k)$ in Eq. (3.3) k is always positive, *i.e.* in Ω^+, and x_j can either be negative, *i.e.* in Ω^-, in rank^- or positive in rank^+, *i.e.* in Ω^+.

From Eq. (3.3) it becomes clear that the rank is non-amenable to gradient descent optimization due to the Heaviside (step) function H (see Fig. 3.4a), whose derivatives are either zero or undefined.

3.2.2.1 SupRank: smooth approximation of the rank.

To provide rank losses amenable to SGD, we introduce a smooth approximation of the rank function. We propose a different behavior between $\text{rank}^+(k)$ and $\text{rank}^-(k)$ in Eq. (3.3) by defining two functions H^+ and H^-. For $\text{rank}^+(k)$, we keep the Heaviside function, *i.e.* $H^+ = H$ (see Fig. 3.4a), meaning that we do not approximate $\text{rank}^+(k)$. This ignores $\text{rank}^+(k)$ in gradient-based ranking optimization. This is done on purpose, indeed for metrics such as AP or R@k optimizing rank^+, *i.e.* switching the order of positive instances, does not improve the metrics. Furthermore, in the case of rank approximation, *e.g.* Smooth-AP, it can be shown that it adds noise to the final gradient. We give more details in the theoretical analysis of smooth approximation of the rank Sec. 3.4.1. It has also been observed in other works that optimizing rank^- is sufficient [245].

For $\text{rank}^-(k)$, we want a smooth surrogate H^- for H that is amenable to SGD and an upper bound on the Heaviside function. We define the following H^- function, illustrated in Fig. 3.4b,

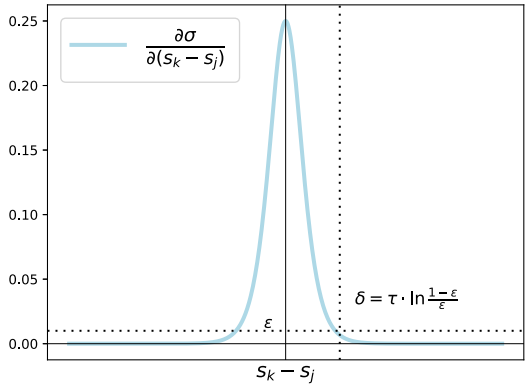


Figure 3.5: Gradient of the temperature scaled sigmoid ($\tau = 0.01$) *vs.* the difference of scores $s_k - s_j$ of a negative pair.

that is both:

$$H^-(t) = \begin{cases} \sigma(\frac{t}{\tau}) & \text{if } t \leq 0, \quad \text{where } \sigma \text{ is the sigmoid function (Fig. 3.4c)} \\ \sigma(\frac{t}{\tau}) + 0.5 & \text{if } t \in [0; \delta] \quad \text{with } \delta \geq 0 \\ \rho \cdot (t - \delta) + \sigma(\frac{\delta}{\tau}) + 0.5 & \text{if } t > \delta \end{cases} \tag{3.4}$$

where σ is the sigmoid function (Fig. 3.4c), δ, τ and ρ are hyperparameters. δ is chosen such that the sigmoidal part of H^- reaches the saturation regime. We keep τ as in [69] and study the robustness to ρ in the experimental section Sec. 3.5.

From H^- in Eq. (3.4), we define the following rank surrogate that can be used plug-and-play for rank losses optimization:

$$\text{rank}_s^-(k) = \sum_{j \in \Omega_k^<} H^-(s_j - s_k) \tag{3.5}$$

Choice of δ. δ is introduced in Eq. (3.4) to define H^-. We choose δ as the point where the gradient of the sigmoid function becomes low $< \epsilon$, and we then have $\delta = \tau \cdot \ln\frac{1-\epsilon}{\epsilon}$. This is illustrated in Fig. 3.5. For our experiments, we use $\epsilon = 10^{-2}$ giving $\delta \simeq 0.05$.

SupRank has two main features:

▶ ① **Surrogate losses based on SupRank are upper bound of the target metrics**, since H^- in Eq. (3.4) is an upper bound of a step function (Fig. 3.4b). This is an important property, since it ensures that the model keeps training until the correct ranking is obtained. It is worth

noting that existing smooth rank approximations in the literature [67], [69], [172], [173] do not fulfill this property.

► ② **SupRank brings training gradients until the correct ranking plus a margin is fulfilled.** When the ranking is incorrect, an instance with a lower relevance $\boldsymbol{x_j}$ is ranked before an instance of higher relevance $\boldsymbol{x_k}$, thus $s_j > s_k$ and $H^-(s_j - s_k)$ in Eq. (3.4) has a non-zero derivative. We use a sigmoid to have a large gradient when $s_j - s_k$ is small. To overcome vanishing gradients of the sigmoid for large values $s_j - s_k$, we use a linear function ensuring constant ρ derivative. When the ranking is correct ($s_j < s_k$), we enforce robustness by imposing a margin parameterized by τ (sigmoid in Eq. (3.4)). This margin overcomes the brittleness of rank losses, which vanish as soon as the ranking is correct [169], [170], [173].

Comparison to sigmoid approximation [69], [70]. SupRank differs from the sigmoid in [69], [70] by i) providing an upper bound on the target rank loss (*i.e.* AP and R@k), ii) improving the gradient flow (Fig. 3.4b vs Fig. 3.4c), and iii) overcoming adverse effects of the sigmoid for $rank^+$, as shown in Fig. 3.1. We experimentally verify the consistent gain brought out by SupRank over the sigmoid approximation. We formalize the intuitions of this paragraph later in Sec. 3.4.1.

3.2.3 Decomposable rank losses.

As illustrated in Eq. (3.1), rank losses decompose linearly between queries $\boldsymbol{q_i}$, but do not between retrieved instances. We therefore focus our analysis of the non-decomposability on a single query. For a retrieval set Ω of N elements, we consider $\{\mathcal{B}_b\}_{b\in\{1:K\}}$ batches of size B, such that $N/B = K \in \mathbb{N}$. Let $\mathcal{M}_b(\boldsymbol{\theta})$ be the metric $\mathcal{M}$ in batch b for a query, we define the "decomposability gap" DG as:

$$DG(\boldsymbol{\theta}) = \frac{1}{K}\sum_{b=1}^{K}\mathcal{M}_b(\boldsymbol{\theta}) - \mathcal{M}(\boldsymbol{\theta}) \tag{3.6}$$

DG in Eq. (3.6) is a direct measure of the non-decomposability of any metric $\mathcal{M}$. Our motivation here is to decrease DG, *i.e.* to have the average metric over the batches lower or equal to the metric computed on the whole set. Note that in some cases DG can be negative, *i.e.* $\frac{1}{K}\sum_{b=1}^{K}\mathcal{M}_b(\boldsymbol{\theta}) < \mathcal{M}(\boldsymbol{\theta})$, although this means the $\mathcal{M}$ is not well estimated, it is still a favorable case. It means that optimizing our batch approximate was sufficient to optimize the overall metric.

We illustrate the decomposability gap DG, for AP, on a toy dataset Fig. 3.6.

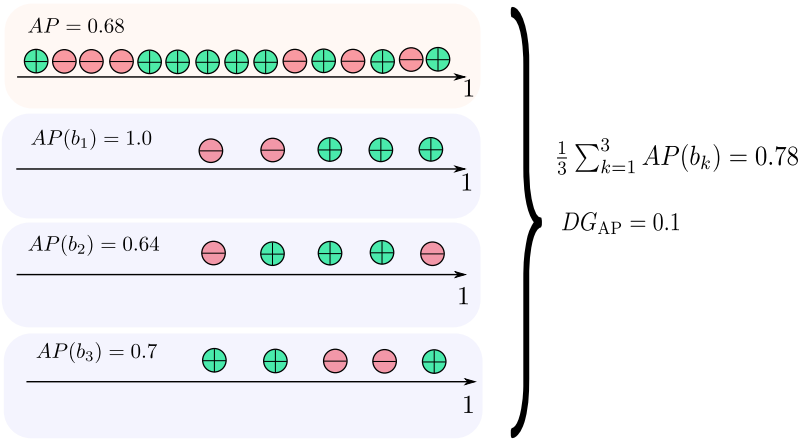


Figure 3.6: Illustration of AP's decomposability gap on a toy dataset.

In order to improve the decomposability of ranking losses, we use an additional loss, and we propose two different losses. This additional loss should support that the local ranking is closer to the global ranking. Using an additional loss introduce little overhead as it is still computed in batch-wise manner.

Pair-based decomposability loss. We use the following decomposability loss $\mathcal{L}_{\text{DG}}$, illustrated on Fig. 3.7:

$$\mathcal{L}_{\text{DG}}(\boldsymbol{\theta}) = \frac{1}{|\Omega^+|} \sum_{\boldsymbol{x_j} \in \Omega^+} [\alpha - s_j]_+ + \frac{1}{|\Omega^-|} \sum_{\boldsymbol{x_j} \in \Omega^-} [s_j - \beta]_+ \tag{3.7}$$

where $[x]_+ = \max(0, x)$. The loss $\mathcal{L}^+_{\text{DG}}$ enforces the score of the positive $\boldsymbol{x_i} \in \Omega^+$ to be larger than α, and $\mathcal{L}^-_{\text{DG}}$ enforces the score of the negative $\boldsymbol{x_j} \in \Omega^-$ to be smaller than $\beta < \alpha$. $\mathcal{L}_{\text{DG}}$ is a contrastive pair-based loss [118], which we revisit in our context to "calibrate" the scores between mini-batches. Intuitively, the fact that the positive (resp. negative) scores are above (resp. below) a threshold α (resp. β) in the mini-batches makes $\mathcal{M}_b$ closer to $\mathcal{M}$, which we support with an analysis in Sec. 3.4.2.

Proxy-based decomposability loss. Motivated by other works [96] we introduce a different decomposability objective, a proxy-based loss:

$$\mathcal{L}^*_{\text{DG}}(\theta) = -\log\left(\frac{\exp(\frac{v_y^T p_y}{\eta})}{\sum_{p_z \in \mathcal{Z}} \exp(\frac{v_y^T p_z}{\eta})}\right), \tag{3.8}$$

where p_y is the normalized proxy corresponding to the fine-grained class of the embedding v_y, $\mathcal{Z}$ is the set of proxies, and η is a temperature scaling parameter. $\mathcal{L}^*_{\text{DG}}$ is a classification-

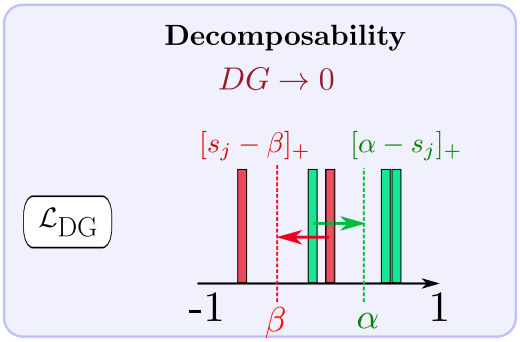


Figure 3.7: $\mathcal{L}_{\text{DG}}$ reduces the non-decomposability by comparing the cosine similarities in each batch to absolute references, α and β.

based proxy loss, or `NormSoftMax` [66], that imposes a margin between instances and the proxies. $\mathcal{L}^*_{\text{DG}}$ has thus a similar effect to $\mathcal{L}_{\text{DG}}$ on the decomposability of rank losses. In our experiments, we show that both decomposability losses improve ranking losses optimization. And in practice $\mathcal{L}^*_{\text{DG}}$ is easier to optimize in large scale settings, we hypothesize that this is because the comparison between batches, *i.e.* the references, is learned instead of fix α and β in $\mathcal{L}_{\text{DG}}$.

3.3 Instantiation to standard image retrieval.

In this section, we apply the framework described previously to standard image retrieval. Specifically, we show how to directly optimize two metrics that are widely used in the image retrieval community, *i.e.* AP and R@k.

3.3.1 Application to Average Precision.

The average precision measures the quality of a ranking by penalizing inversion between positives and negatives. It strongly penalizes inversion at the top of the ranking. It is defined for each query q_i as follows:

$$\text{AP}_i = \frac{1}{|\Omega_i^+|} \sum_{k \in \Omega_i^+} \frac{\text{rank}^+(k)}{\text{rank}^+(k) + \text{rank}^-(k)} \tag{3.9}$$

The overall AP loss $\mathcal{L}_{\text{AP}}$ is averaged over all queries:

$$\mathcal{L}_{\text{AP}}(\boldsymbol{\theta}) = 1 - \frac{1}{M} \sum_{i=1}^{M} \text{AP}_i(\boldsymbol{\theta}) \tag{3.10}$$

Using our surrogate of the rank, SupRank, we define the following AP surrogate loss:

$$\mathcal{L}_{\text{SupAP}}(\boldsymbol{\theta}) = 1 - \frac{1}{M}\sum_{i=1}^{M}\frac{1}{|\Omega_i^+|}\sum_{k\in\Omega_i^+}\frac{\text{rank}^+(k)}{\text{rank}^+(k) + \text{rank}_s^-(k)} \tag{3.11}$$

$\mathcal{L}_{\text{SupAP}}$ is an upper bound on $\mathcal{L}_{\text{AP}}$, as $\text{rank}_s^-(k) > \text{rank}^-(k)$ is on the denominator, which is then combined with the minus sign before the sum. Finally, we equip the AP surrogate loss with the $\mathcal{L}_{\text{DG}}$ loss to support the decomposability of the AP, yielding our **RO**bust **A**nd **D**eco**M**posable **A**verage **P**recision:

$$\mathcal{L}_{\text{ROADMAP}}(\boldsymbol{\theta}) = (1-\lambda)\cdot\mathcal{L}_{\text{SupAP}}(\boldsymbol{\theta}) + \lambda\cdot\mathcal{L}_{\text{DG}}(\boldsymbol{\theta}) \tag{3.12}$$

3.3.2 Application to the Recall at k.

Another metric often used in image retrieval is the recall rate at k. In the image retrieval community, it is often defined as:

$$\text{R@k} = \frac{1}{M}\sum_{i=1}^{M}\mathbb{1}(\text{positive element in top-}k) \tag{3.13}$$

However, in the literature, the recall is most often defined as:

$$\text{TR@k} = \frac{1}{M}\sum_{i=1}^{M}\frac{\#\text{ positive elements in top-}k}{\min(k, \#\text{ positive elements})} \tag{3.14}$$

It was shown in [70] that the TR@k can be written similarly to other ranking-based metrics, *i.e.* using the rank, for each query q_i as:

$$\text{TR@k} = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{\min(|\Omega_i^+|, k)}\sum_{p\in\Omega_i^+}H\left(k - \text{rank}(p)\right) \tag{3.15}$$

Using the expression of Eq. (3.15) and SupRank we can derive a surrogate loss function for the recall:query as:

$$\mathcal{L}_{\text{Sup-R@k}}(\boldsymbol{\theta}) = 1 - \frac{1}{M}\sum_{i=1}^{M}\frac{1}{\min(|\Omega_i^+|, k)}\sum_{p\in\Omega_i^+}\sigma\left(\frac{k - \left(\text{rank}^+(p) + \text{rank}_s^-(p)\right)}{\tau^*}\right) \tag{3.16}$$

$\mathcal{L}_{\text{Sup-R@k}}$ is again an upper bound, as $\text{rank}^-(k)$ is on the numerator, followed by two minus signs. The authors of [70] use different level of recalls in their loss, which we follow *i.e.*

$\mathcal{L}_{\text{Sup-R@}\mathcal{K}} = \frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}}\mathcal{L}_{\text{Sup-R@k}}$, it is necessary to provide enough gradient signal to all positive items. To train $\mathcal{L}_{\text{Sup-}R@k}$, it is also necessary to approximate a second time the Heaviside function, using a sigmoid with temperature factor τ^*. We combine it with $\mathcal{L}_{\text{DG}}$, yielding the resulting differentiable and decomposable R@k loss:

$$\mathcal{L}_{\text{ROD-R@}\mathcal{K}}(\boldsymbol{\theta}) = (1-\lambda)\cdot\mathcal{L}_{\text{Sup-R@}\mathcal{K}}(\boldsymbol{\theta}) + \lambda\cdot\mathcal{L}_{\text{DG}}(\boldsymbol{\theta}) \tag{3.17}$$

3.4 Theoretical analysis and intuitions.

3.4.1 Properties of SupAP & comparison to SmoothAP.

We further discuss and give additional explanations of the property of our $\mathcal{L}_{\text{SupAP}}$ approximation, and especially its comparison with respect to the sigmoid used in SmoothAP [69].

As shown in Fig. 3.1 the smooth rank approximation in [69] has several drawbacks. Specifically, we explain in more detail the following three limitations of SmoothAP, which come from the use of the sigmoid function to approximate the Heaviside (step) function for computing the rank:

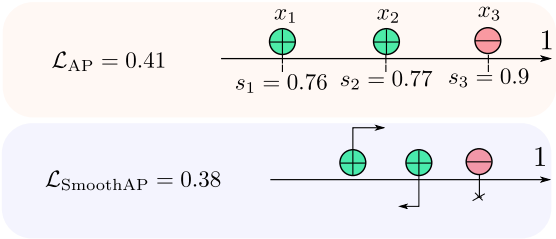


Figure 3.8: Limitation of the smooth rank approximation in [69]: contradictory gradient flow for the positives samples $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ (in green), vanishing gradient for the negative example $\boldsymbol{x_3}$ (in red), and no guarantees of having an upper bound of $\mathcal{L}_{\text{AP}}$.

i **Contradictory gradient flow for positives samples:** Firstly, we can see on the toy dataset of Fig. 3.8 that the gradients of the two positive examples (in green) with SmoothAP have opposite directions. The positive with the lowest rank $\boldsymbol{x_1}$ has a gradient in the good direction, since it leads to increase $\boldsymbol{x_1}$'s score because the correct ordering is not reached (the negative instance $\boldsymbol{x_3}$ has a better rank). But the gradient of the positive with the highest rank $\boldsymbol{x_2}$ is on the wrong direction, since it tends to decrease $\boldsymbol{x_2}$'s score. This is an undesirable behavior, which comes from the use of the sigmoid in $\mathcal{L}_{\text{SmoothAP}}$. In the example of Fig. 3.8, we can actually show that

$$\boxed{\frac{\partial \mathcal{L}_{\text{SmoothAP}}}{\partial s_1} = -\frac{\partial \mathcal{L}_{\text{SmoothAP}}}{\partial s_2}}$$

Proof in Appendix A.1.1.

ii **Vanishing gradients:** Secondly, SmoothAP [69] has vanishing gradients due to its use of the sigmoid function. This is illustrated on the toy dataset in Fig. 3.8. The negative instance $\boldsymbol{x_3}$ has a high score s_3, but does not receive any gradient, which does not enable it to lower its score, although it would improve the overall ranking. This is because the score difference between $\boldsymbol{x_3}$ and $\boldsymbol{x_2}$ is large, *i.e.* $s_3 - s_2 = 0.13$. Similarly, $s_3 - s_1 = 0.14$. Consequently, both $s_3 - s_2$ and $s_3 - s_1$ fall into the saturation regime of the sigmoid, preventing to propagate any gradient (see Fig. 3.4c).

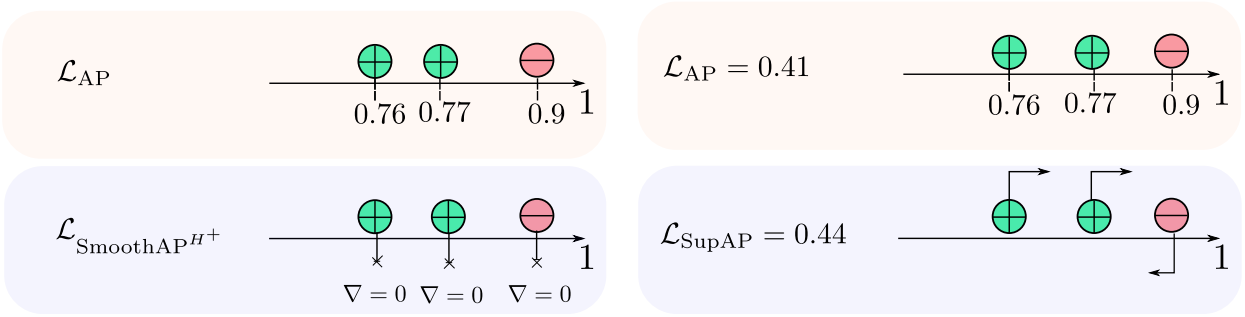
iii **Finally, $\boldsymbol{\mathcal{L}_{\text{SmoothAP}}}$ is not an upper bound of $\boldsymbol{\mathcal{L}_{\text{AP}}}$.** The use of the sigmoid means that both rank^+ and rank^- can be over or underestimated. If rank^+ is overestimated (resp. underestimated) $\mathcal{L}_{\text{SmoothAP}}$ underestimates $\mathcal{L}_{\text{AP}}$ (resp. overestimates). And if rank^- is overestimated (resp. underestimated) $\mathcal{L}_{\text{SmoothAP}}$ overestimates $\mathcal{L}_{\text{AP}}$ (resp. overestimated). Therefore, $\mathcal{L}_{\text{SmoothAP}}$ can be larger or lower than $\mathcal{L}_{\text{AP}}$ in general. In the example of Fig. 3.8, we show that $\mathcal{L}_{\text{SmoothAP}}$ is lower than $\mathcal{L}_{\text{AP}}$.

We address those three issues with SupRank:

i **Using the true Heaviside (step) function $\mathbf{H}^+$ for rank^+** allows having the expected behavior regarding the gradients of positives. When Changing $\mathbf{H}^+$ for rank^+ in Fig. 3.9a, we can see that we fix the problem of opposite gradients for the positive examples $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ - although the gradient is zero.

ii **Using $\mathbf{H}^-$ for rank^- overcomes vanishing gradients**. By using $\mathbf{H}^-$ in Eq. (3.4), we design a linear function for positive $(s_j - s_k)$ values, where s_j (resp. s_k) is the score of a negative (resp. positive) example - see Fig. 3.4b. We can see in Fig. 3.9b that this change enables to have gradients in the correct directions for the two positive instances $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ (tending to increase their scores), and for the negative instance $\boldsymbol{x_3}$ (tending to decrease its score).

iii **$\boldsymbol{\mathcal{L}_{\text{SupAP}}}$ is an upper bound of $\boldsymbol{\mathcal{L}_{\text{AP}}}$.** By the proposed design of $\mathbf{H}^-$ in Eq. (3.4), we have $\text{rank}^-_s(k) \geq \text{rank}^-(k)$. Since we do not approximate $\text{rank}^+(k)$ by keeping the Heaviside function, it leads to $\frac{\text{rank}^+(k)}{\text{rank}^+(k)+\text{rank}^-_s(k)} \leq \frac{\text{rank}^+(k)}{\text{rank}^+(k)+\text{rank}^-(k)}$, and therefore $\boldsymbol{\mathcal{L}}_{\text{SupAP}} \geq \mathcal{L}_{\text{AP}}$.



(a) When replacing H^+ by the Heaviside function in SmoothAP we stop the unexpected behavior of the gradient flow. However, there are still vanishing gradients.

(b) Our $\mathcal{L}_{\text{SupAP}}$ has gradients that do not stop until the correct ranking is achieved.

Figure 3.9: We illustrate the different steps to build $\mathcal{L}_{\text{SupAP}}$. On Fig. 3.9a, we change H^+ to be the true Heaviside (step) function. On Fig. 3.9b, we replace the sigmoid by H^- defined in Eq. (3.4). Using H^+ and H^-, $\mathcal{L}_{\text{SupAP}}$ is an upper bound of $\mathcal{L}_{\text{AP}}$.

Overall, $\mathcal{L}_{\text{SupAP}}$ has all the desired properties : i) A correct gradient flow during training, ii) No vanishing gradients while the correct ranking is not reached, iii) Being an upper bound on the AP loss $\mathcal{L}_{\text{AP}}$.

3.4.2 Properties of the $\mathcal{L}_{\text{DG}}$ loss function.

Upper bound on the decomposability gap for AP. To formalize the intuition behind $\mathcal{L}_{\text{DG}}$, we provide a theoretical analysis of the impact on the global ranking of $\mathcal{L}_{\text{DG}}$ in Eq. (3.7) for AP. Firstly, we can see that if $\mathcal{L}^-_{\text{DG}} = \mathcal{L}^+_{\text{DG}} = 0$, on each batch, the overall AP and the AP in batches is null, *i.e.* $DG(\boldsymbol{\theta}) = 0$ and we get a decomposable AP. In a more general setting, we show that minimizing $\mathcal{L}_{\text{DG}}$ on each batch reduces the decomposability gap, hence improving the decomposability of the AP.

Let's consider K batches $\{\mathcal{B}^b\}_{b\in\{1:K\}}$ of batch size B divided in Ω^+_b positive instances and Ω^-_b negative instances w.r.t. the query $\boldsymbol{q_i}$. To give some insight, we assume that the AP of each batch is one (*i.e.* $\text{AP}^b_i = 1$), and give the following upper bound of DG_{AP}:

$$0 \leq DG \leq 1 - \frac{1}{\sum_{b=1}^{K} |\Omega^+_b|} \left(\sum_{b=1}^{K} \sum_{j=1}^{|\Omega^+_b|} \frac{j + |\Omega^+_1| + \cdots + |\Omega^+_{b-1}|}{j + |\Omega^+_1| + \cdots + |\Omega^+_{b-1}| + |\Omega^-_1| + \cdots + |\Omega^-_{b-1}|} \right) \tag{3.18}$$

This upper bound of the decomposability gap is given in the worst case for the global AP: the global ranking is built from the juxtaposition of the batches Fig. 3.10. Proof in Appendix A.1.3.

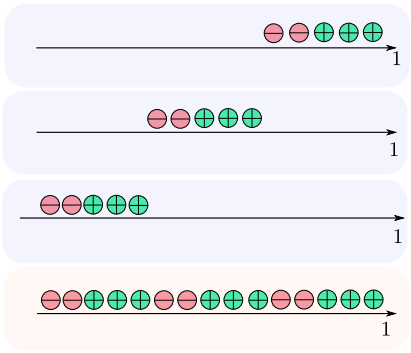


Figure 3.10: The worst case when computing the global AP would be that each batch is juxtaposed.

Refined upper bound on the decomposability gap. We can refine this upper bound by introducing the calibration loss $\mathcal{L}_{\text{DG}}$ and constraining the scores of positive and negative instances to be well calibrated. On each batch we define the following quantities $E_b^- = \sum_{j \in \Omega_i^-} \mathbb{1}(s_j > \beta)$ which are the negative instances that do not respect the constraints and $G_b^- = \sum_{j \in \Omega_i^-} \mathbb{1}(s_j \leq \beta)$ the negative instances that do. We similarly define E_b^+ and G_b^+. We then have the following upper bound on the decomposability gap :

$$0 \leq DG \leq 1 - \frac{1}{\sum_{b=1}^{K} |\Omega_b^+|} \left(\sum_{b=1}^{K} \left[\sum_{j=1}^{G_b^+} \frac{j + G_1^+ + \cdots + G_{b-1}^+}{j + G_1^+ + \cdots + G_{b-1}^+ + E_1^- + \ldots E_{b-1}^-} + \sum_{j=1}^{E_b^+} \frac{j + G_b^+ + |\Omega_1^+| + \cdots + |\Omega_{b-1}^+|}{j + G_b^+ + |\Omega_1^+| + \cdots + |\Omega_{b-1}^+| + |\Omega_1^-| + \cdots + |\Omega_{b-1}^-|} \right] \right) \tag{3.19}$$

This refined upper bound is tighter than the upper bound of Eq. (3.18). Our new $\mathcal{L}_{\text{DG}}$ loss directly optimizes this upper bound (by explicitly optimizing $E_b^-, E_b^+, G_b^+, G_b^+$), making it tighter, hence improving the decomposability of the AP. We give the proof in Appendix A.1.4.

3.5 Experiments.

3.5.1 Experimental setup.

We evaluate ROADMAP on the following three image retrieval datasets:

Stanford Online Product (SOP) [246] is a standard dataset for Image Retrieval it has two levels of semantic scales, the object Id (fine) and the object category (coarse). It depicts EBay on-

line objects, with 120053 images of 22634 objects (Id) classified into 12 (coarse) categories (*e.g.* bikes, coffee makers *etc.*), see Fig. 3.11. We use the reference train and test splits from [246]. The dataset can be downloaded at: https://cvgl.stanford.edu/projects/lifted_struct/.

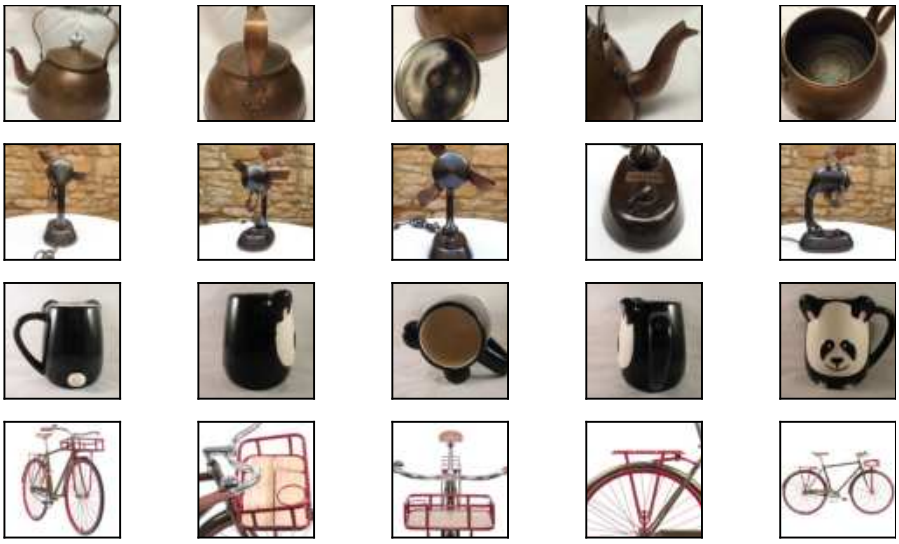


Figure 3.11: Images from Stanford Online Products.

iNaturalist-2018 [29] is a dataset that has been used for image retrieval in recent works [69], [95]. It depicts animals, plants, mushroom *etc.* in wildlife, see Fig. 3.12, it has in total 461 939 images and 8142 fine-grained classes ("Species"). We use the standard Image Retrieval splits from [69]. The dataset can be downloaded at: github.com/visipedia/inat_comp, and the retrieval splits at: drive.google.com.

CUB-200-2011 [247] contains 11788 images of birds classified into 200 fine-grained classes. We follow the standard protocol and use the first (resp. last) 100 classes for training (resp. evaluation). The dataset can be downloaded at: https://www.vision.caltech.edu/datasets/cub_200_2011/.

Details of the backbones used. We briefly describe the backbones used throughout out the experiments.

- **ResNet-50** [45] We use the well-known convolutional neural network ResNet-50. We remove the linear classification layer. We also add a linear projection layer to reduce the dimension (*e.g.* from 2048 to 512).

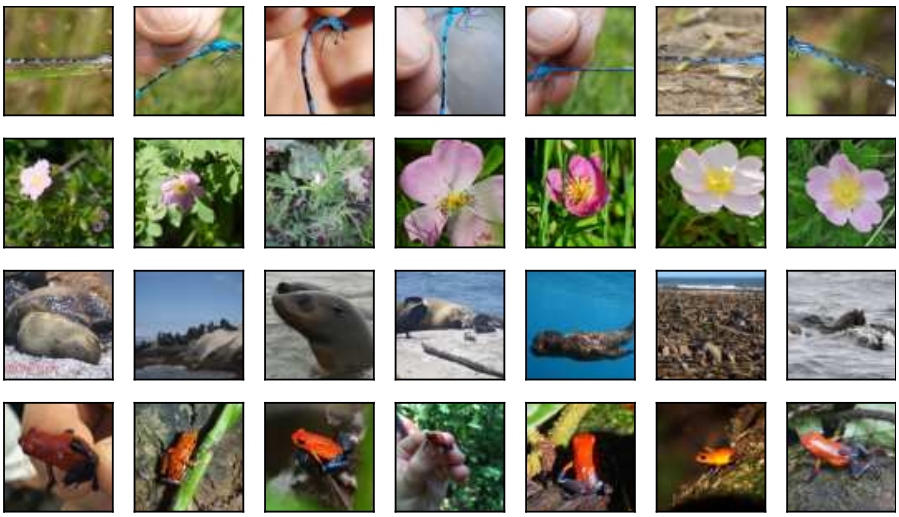


Figure 3.12: Images from iNaturalist-2018.

- **DeiT [47]** Recently, transformer models have been introduced for computer vision [47], [248]. They establish new state-of-the-art performances on computer vision tasks. We use the DeiT-S from [47] which has fewer parameters than the ResNet-50 ($\sim$ 21 million for DeiT *vs.* 25 for ResNet-50). We use the pre-trained version with distillation from [47] and its implementation in the `timm` library [249].

Detail on experimental setup. We use the standard data augmentation strategy during training: images are resized so that their shorter side has a size of 256, we then make a random crop that has a size between 40 and 256, and aspect ratio between 3/4 and 4/3. This crop is then resized to 224x224, and flipped horizontally with a 50% chance. During evaluation, images are resized to 256 and then center cropped to 224.

We use two different strategy to sample each mini-batch. On CUB and iNaturalist, we choose a batch size (*e.g.* 128) and a number of samples per classes (*e.g.* 4). We then randomly sample classes (*e.g.* 32) to construct our batches. For SOP, we use the hard sampling strategy from [173]. For each pair of category (*e.g.* bikes and coffee makers) we use the preceding sampling strategy. This sampling technique is used because it yields harder and more informative batches. The intuition behind this sampling is that it will be harder to discriminate two bikes from one another, than a bike and a sofa.

Test protocol. Methods are evaluated using the standard recall at k (R@k) and mean average precision at R [250] (mAP@R) metrics, detailed below.

Recall@K The Recall@K metrics is often used in the literature. For a single query, the Recall@K is 1 if a positive instance is in the K nearest neighbors, and 0 otherwise. The Recall@K is then averaged on all the queries. Researcher use different values of K for a given dataset (*e.g.* 1, 2, 4, 8 on CUB), for details see Eq. (2.3).

mAP@R Recently, the mAP@R has been introduced in [250]. The authors show that this metric is less noisy and better captures the performance of a model. The mAP@R is a partial AP, computed on the R first instances retrieved, with R being set to the number of positive instances wrt. a query. mAP@R is a lower bound of the AP (mAP@R = AP when the correct ranking is achieved, *i.e.* mAP@R = AP = 1).

$$mAP@R_i = \frac{1}{R}\sum_{j=1}^{R} P(j), \quad \text{where } P(j) = \begin{cases} \text{precision at j if the jth retrieval is correct} \\ 0 \quad \text{otherwise} \end{cases} \tag{3.20}$$

3.5.2 ROADMAP validation.

Table 3.1: Comparison between ROADMAP and state-of-the-art AP ranking based methods.

	SOP		iNaturalist	
Method	R@1	mAP@R	R@1	mAP@R
Fast-AP [173]	77.8	50.5	59.9	24.0
SoftBin-AP [67]	79.7	52.7	63.6	25.4
BlackBox-AP [68]	80.0	53.1	52.3	15.2
Smooth-AP [69]	80.9	54.3	67.3	26.5
ROADMAP	**81.9**	**55.7**	**71.8**	**29.5**

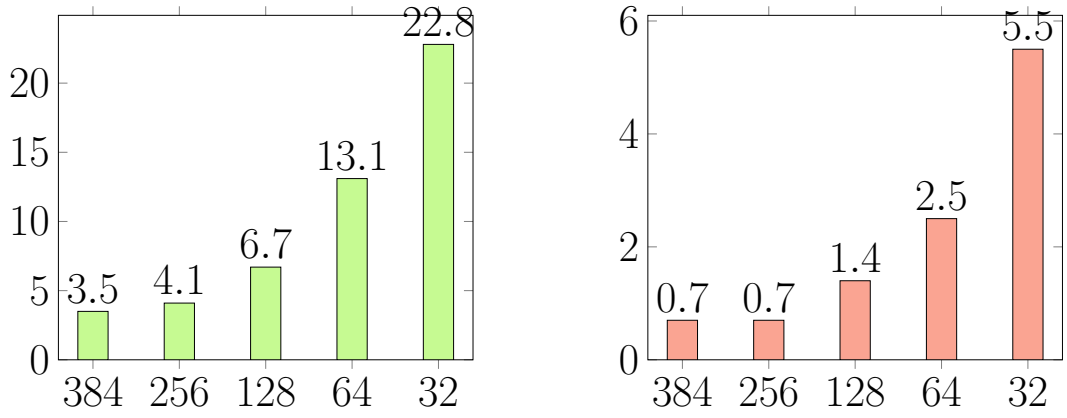
In this section, we run all experiments under the same settings, and use publicly available implementations of all baselines. We use a ResNet-50 backbone with average pooling, layer normalization without affine parameters and a projection head that reduces the dimension from 2048 to 512. We use a batch size of 256 by sampling 4 images per class and the hierarchical sampling of [173] for SOP, with resolution 224×224, standard data augmentation (random resize crop, horizontal flipping), the Adam optimizer (with learning rate of $5 \cdot 10^{-5}$ on SOP and $1 \cdot 10^{-5}$ on iNaturalist, with cosine decay) and train for 100 epochs.

3.5.2.1 Comparison to AP approximations.

In Tab. 3.1, we compare ROADMAP to AP loss approximations including soft-binning approaches Fast-AP [173] and SoftBin-AP [67], the generic solver BlackBox-AP [68], and the smooth rank approximation [69]. We observe that ROADMAP outperforms all the current AP approximations by a large margin. The gains are especially pronounced on the large-scale dataset iNaturalist.

3.5.2.2 Analysis on decomposability.

The decomposability gap depends on the batch size Eq. (3.6). To illustrate this, we monitor on Fig. 3.13 the relative improvement when adding $\mathcal{L}^*_{\text{DG}}$ to $\mathcal{L}_{\text{SupAP}}$ as the batch size decreases. We can see that the relative improvement becomes larger as the batch size gets smaller. This confirms our intuition that the decomposability loss $\mathcal{L}^*_{\text{DG}}$ has a stronger effect on smaller batch sizes, for which the AP estimation is noisier and *DG* larger. This is critical on the large-scale dataset iNaturalist, where the batch AP on usual batch sizes is a very poor approximation of the global AP.



(a) iNaturalist (b) SOP

Figure 3.13: Relative increase of mAP@R *vs.* batch size when adding $\mathcal{L}_{\text{DG}}$ to $\mathcal{L}_{\text{SupAP}}$.

In Tab. 3.2 we compare ROADMAP to the cross-batch memory [73] (XBM) which is used to reduce the gap between batch-AP and global AP. We use XBM with a batch size of 128 and store all the dataset, and use the setup described previously otherwise. ROADMAP outperforms XBM both on SOP and iNaturalist, with gains more pronounced on iNaturalist with +12.5pt R@1 and +11 mAP@R. $\mathcal{L}^*_{\text{DG}}$ allows us to train models even with smaller batches.

Table 3.2: Comparison between XBM [73] and ROADMAP.

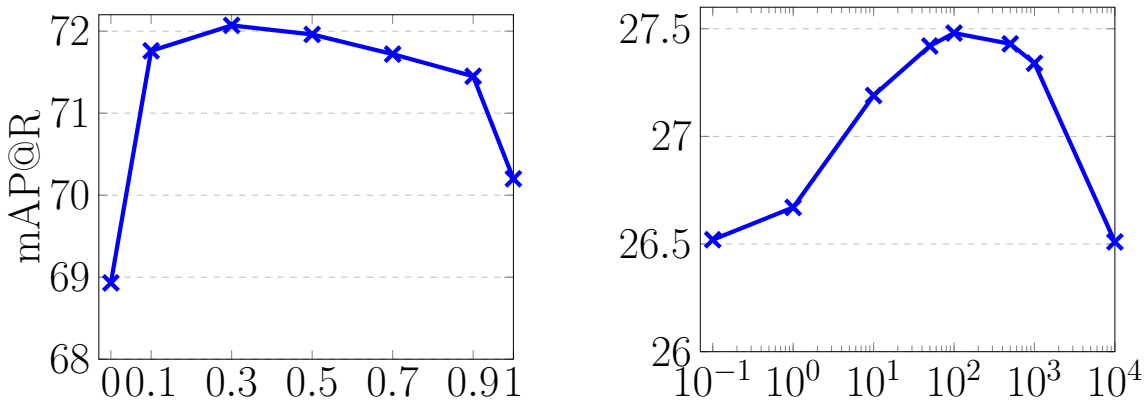
	SOP		iNaturalist	
Method	R@1	mAP@R	R@1	mAP@R
XBM [73]	80.6	54.9	59.3	18.5
ROADMAP	**81.9**	**55.7**	**71.8**	**29.5**

3.5.2.3 Ablation study.

To investigate more in-depth the impact of the two components of our framework, we perform ablation studies in Tab. 3.3. We show the improvements against Smooth-AP [69] and Smooth-R@k [70] when replacing the sigmoid by SupRank Eq. (3.10), and the use of $\mathcal{L}_{\text{DG}}$ Eq. (3.7) or $\mathcal{L}^*_{\text{DG}}$ Eq. (3.8). We can see that both $\mathcal{L}_{\text{SupAP}}$ and $\mathcal{L}_{\text{Sup-R@k}}$ consistently improve performances over the baselines, +0.5pt mAP@R on SOP and +1pt mAP@R on iNaturalist for both Sup-AP and Sup-R@k. Both $\mathcal{L}_{\text{DG}}$ and $\mathcal{L}^*_{\text{DG}}$ improve over the smooth surrogates, with strong gains on iNaturalist, *e.g.* $\mathcal{L}^*_{\text{DG}}$ improves by +2.9pt R@1 over Sup-AP and +3.7pt R@1 over Sup-R@k. This is because the batch vs. dataset size ratio $\frac{B}{N}$ is tiny ($\sim 8 \cdot 10^{-4} \ll 1$), making the decomposability gap in Eq. (3.6) huge. On SOP $\mathcal{L}_{\text{DG}}$ and $\mathcal{L}^*_{\text{DG}}$ work similarly, however on the large scale iNaturalist $\mathcal{L}^*_{\text{DG}}$ performs better than $\mathcal{L}_{\text{DG}}$, as discussed in Sec. 3.2.3 this could come from the fact that the margin in $\mathcal{L}^*_{\text{DG}}$ (*i.e.* the distance to the proxies) are learnable. In the following, we choose to keep only $\mathcal{L}^*_{\text{DG}}$.

Table 3.3: Ablation study of the two components of our framework.

			SOP		iNaturalist	
Method	rank	*DG*	R@1	mAP@R	R@1	mAP@R
Smooth-AP	sigmoid	✗	80.9	54.3	67.3	26.5
Sup-AP	SupRank	✗	81.2	54.8	68.9	27.5
ROADMAP	SupRank	$\mathcal{L}_{\text{DG}}$	81.7	**55.7**	69.1	27.6
		$\mathcal{L}^*_{\text{DG}}$	**81.9**	**55.7**	**71.8**	**29.5**
Smooth-R@k	sigmoid	✗	80.5	53.7	66.4	25.5
Sup-R@k	SupRank	✗	80.7	54.2	68.2	26.4
ROD-R@k	SupRank	$\mathcal{L}_{\text{DG}}$	**82.4**	**56.6**	69.3	27.0
		$\mathcal{L}^*_{\text{DG}}$	81.9	55.8	**71.9**	**29.8**



(a) R@1 *vs.* λ for $\mathcal{L}_{\text{ROADMAP}}$ (b) mAP@R *vs.* ρ for $\mathcal{L}_{\text{SupAP}}$

Figure 3.14: Robustness to hyperparameters on iNaturalist.

3.5.2.4 ROADMAP hyperparameters.

We demonstrate the robustness of our framework to hyperparameters in Fig. 3.14. Firstly, Fig. 3.14a illustrates the complementarity between the two terms of $\mathcal{L}_{\text{ROADMAP}}$. For $0 < \lambda < 1$, $\mathcal{L}_{\text{ROADMAP}}$ outperforms both $\mathcal{L}_{\text{SupAP}}$ and $\mathcal{L}^{*}_{\text{DG}}$. While we use $\lambda = 0.1$ in our experiments, hyperparameter tuning could yield better results, *e.g.* with $\lambda = 0.3$ $\mathcal{L}_{\text{ROADMAP}}$ has 72.1 R@1 *vs.* 71.8 R@1 reported in Tab. 3.1. Secondly, Fig. 3.14b shows the influence of the slope ρ that controls the linear regime in H^{-}. As shown in Fig. 3.14b, the improvement is important and stable in $[10, 100]$. Note that $\rho > 1$ already improves the results compared to $\rho = 0$ in [69]. There is a decrease when $\rho \gg 10^3$, probably due to the high gradient that takes over the signal for correctly ranked samples.

3.5.3 State-of-the-art comparison.

In this section, we compare our AP approximation method, ROADMAP, to state-of-the-art methods, on SOP, CUB, and iNaturalist. We use ROADMAP with a memory [73] to virtually increase the batch size. Note that using batch memory is less computationally expensive than methods such as [70] which trade computational time for memory footprint by using two forward passes. We apply ROADMAP on both a convolutional backbone, ResNet-50 with GeM pooling [255] and layer normalization, and Vision transformer models [248], DeiT-S [47] (Imagenet-1k pre-trained as in [254]) and ViT-B (Imagenet21k pre-trained as in [70]). For convolutional backbones, we choose to keep the standard images of size 224×224 for both training and inference on SOP and iNaturalist, and use more recent settings [157], [164] for CUB and use images of size 256×256. Vision transformers experiments use images of size 224×224.

Table 3.4: Comparison of state-of-the-art performances on R@K from the literature on SOP, CUB, and iNaturalist with the proposed ROADMAP. Except for the ViT categories, all methods rely on a standard convolutional backbone (generally ResNet-50).

			SOP			CUB				iNaturalist			
	Method	dim	1	10	100	1	2	4	8	1	4	16	32
Metric learning	Triplet SH [65]	512	72.7	86.2	93.8	63.6	74.4	83.1	90.0	58.1	75.5	86.8	90.7
	MS [160]	512	78.2	90.5	96.0	65.7	77.0	86.3	91.2	-	-	-	-
	SEC [251]	512	78.7	90.8	96.6	68.8	79.4	87.2	92.5	-	-	-	-
	HORDE [252]	512	80.1	91.3	96.2	66.8	77.4	85.1	91.0	-	-	-	-
	XBM [73]	128	80.6	91.6	96.2	65.8	75.9	84.0	89.9	-	-	-	-
	Triplet SCT [157]	512	81.9	92.6	96.8	57.7	69.8	79.6	87.0	-	-	-	-
Classification	ProxyNCA [161]	512	73.7	-	-	49.2	61.9	67.9	72.4	61.6	77.4	87.0	90.6
	ProxyGML [243]	512	78.0	90.6	96.2	66.6	77.6	86.4	-	-	-	-	-
	NSoftmax [66]	512	78.2	90.6	96.2	61.3	73.9	83.5	90.0	-	-	-	-
	NSoftmax [66]	2048	79.5	91.5	96.7	65.3	76.7	85.4	91.8	-	-	-	-
	Cross-Entropy [253]	2048	81.1	91.7	96.3	69.2	79.2	86.9	91.6	-	-	-	-
	ProxyNCA++ [164]	512	80.7	92.0	96.7	69.0	79.8	87.3	92.7	-	-	-	-
	ProxyNCA++ [164]	2048	81.4	92.4	96.9	**72.2**	**82.0**	**89.2**	**93.5**	-	-	-	-
Ranking	FastAP [173]	512	76.4	89.0	95.1	-	-	-	-	60.6	77.0	87.2	90.6
	BlackBox [68]	512	78.6	90.5	96.0	64.0	75.3	84.1	90.6	62.9	79.4	88.7	91.7
	SmoothAP [69]	512	80.1	91.5	96.6	-	-	-	-	67.2	81.8	90.3	93.1
	R@k [70]	512	82.8	92.9	97.0	-	-	-	-	71.2	84.0	91.3	93.6
	R@k + SiMix [70]	512	82.1	92.8	97.0	-	-	-	-	71.8	84.7	91.9	94.3
	ROADMAP (ours)	512	**83.3**	**93.6**	**97.4**	69.4	79 4	87.2	92.1	**73.1**	**85.7**	**92.7**	**94.8**
DeiT-S	IRT_R [254]	384	84.2	93.7	97.3	76.6	85.0	91.1	94.3	-	-	-	-
	ROADMAP (ours)	384	**85.2**	**94.5**	**97.9**	**77.6**	**86.2**	**91.6**	**95.0**	**74.7**	**86.9**	**93.4**	**95.4**
ViT-B	R@k + SiMix [70]	512	88.0	96.1	98.6	-	-	-	-	83.9	92.1	95.9	97.2
	ROADMAP (ours)	512	**88.4**	**96.4**	**98.7**	**86.8**	**91.7**	**94.6**	**96.5**	**85.1**	**93.0**	**96.6**	**97.7**

In Tab. 3.4, using convolutional backbones, ROADMAP outperforms most state-of-the-art methods when evaluated at different (standard) R@k. As ROADMAP optimizes directly the evaluation metrics, it outperforms metric learning and classification-based methods, *e.g.* +1.4pt R@1 on SOP compared to Triplet SCT [157] or +1.9pt R@1 on SOP *vs.* ProxyNCA++ [164]. ROADMAP also outperforms R@k [70] with +1.2pt R@1 on SOP and +1.3pt R@1 on iNaturalist. This is impressive as R@k [70] uses a strong setup, *i.e.* a batch size of 4096 and Similarity mixup. On the small-scale dataset CUB, our method is competitive with methods such as ProxyNCA++ with the same embedding size of 512.

Finally, we show that ROADMAP also improves Vision Transformers for image retrieval. With DeiT-S, ROADMAP outperforms [254] on both SOP and CUB by +1pt R@1, this again shows the interest of directly optimizing the metrics rather than the pair loss of [73] used in [254]. With ViT-B, ROADMAP outperforms [70] by +0.4pt R@1 and +1.2pt R@1 on SOP and iNaturalist, respectively. We attribute this to the fact that our loss is an actual upper bound of the metric, in addition to our decomposability loss.

Preliminary results on Landmarks retrieval. We show in Tab. 3.5 preliminary experiments to evaluate ROADMAP on $\mathcal{R}$Oxford and $\mathcal{R}$Paris [255], by training our model on the SfM-120k dataset and using the standard GitHub code for evaluation[1]. We can see that ROADMAP is significantly better than [254] with the DeiT-S [47] on $\mathcal{R}$Oxford and $\mathcal{R}$Paris medium protocol, and has similar performances for $\mathcal{R}$Paris hard protocol. This highlights the relevance of using ROADMAP instead of the contrastive loss used in [254].

Table 3.5: Comparison of ROADMAP vs IRT [254] on $\mathcal{R}$Oxford and $\mathcal{R}$Paris [255]. Models are DeiT-S [47], ROADMAP is trained with a batch size of 128.

Method	$\mathcal{R}$Oxford		$\mathcal{R}$Paris	
	Medium	Hard	Medium	Hard
IRT [254]	34.5	15.8	65.8	42.0
ROADMAP (ours)	**38.9**	**20.7**	**67.5**	**42.3**

Results on Coexya's data. In this section we apply ROADMAP to one of Coexya's mono-label dataset, Shapes that 20 classes and $30k$ images. We evaluate in two different settings, an open-set one, similarly to the standard image retrieval setting, and a closed-setting. We can see that ROADMAP outperforms both the baseline (classification based) and Smooth-AP on AP evaluation.

Table 3.6: Comparison of ROADMAP vs. baselines on Coexya's Shapes. Models are DeiT-S [47].

Method	Open-set		Closed-set	
	R@1	AP	R@1	AP
Baseline	60.0	36.0	63.0	25.0
Smooth-AP [69]	63.0	49.0	**64.6**	35.2
ROADMAP (ours)	**65.0**	**52.0**	64.0	**37.9**

[1] https://github.com/filipradenovic/cnnimageretrieval-pytorch

3.5.4 Qualitative results.

As a qualitative assessment, we show in Fig. 3.15 some results of ROADMAP on iNaturalist. We show the queries (in purple) and the 4 most similar retrieved images (in green). We can appreciate the semantic quality of the retrieval.

Fig. 3.16 shows another qualitative assessment on iNaturalist, where ROADMAP corrects some failing cases of the SmoothAP baseline.

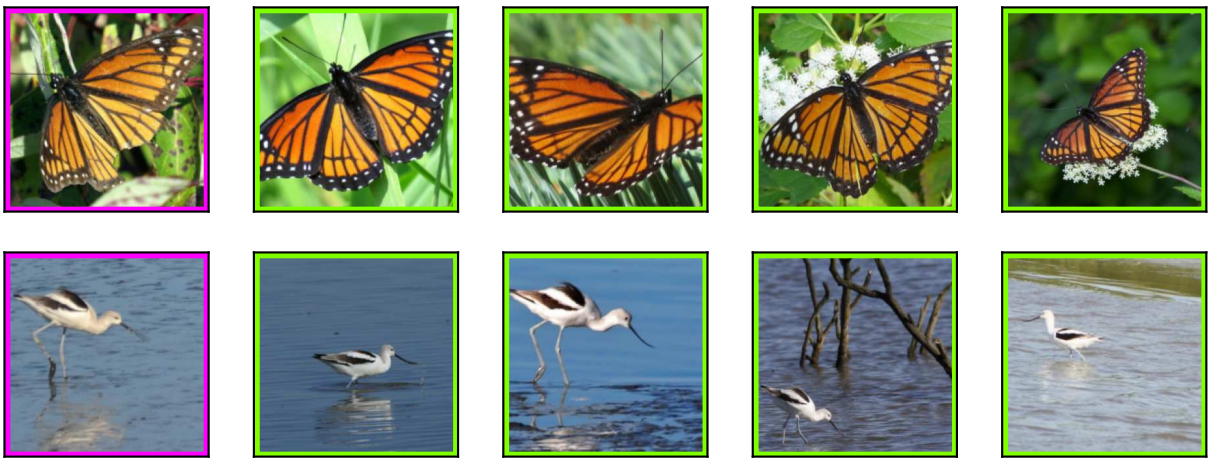


Figure 3.15: Results on iNaturalist: a query (purple) with the 4 most similar retrieved images (green).

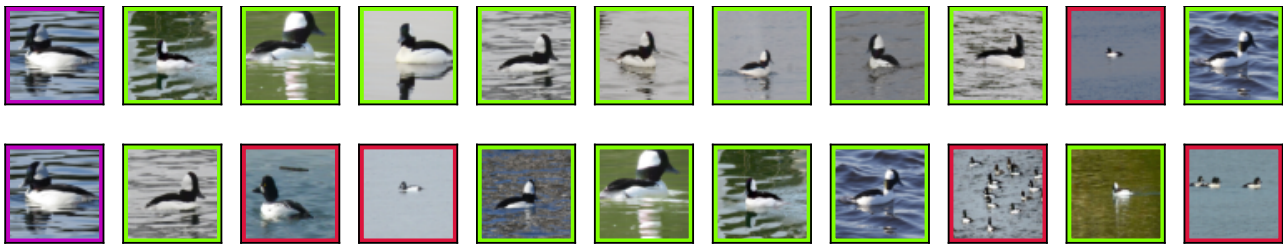


Figure 3.16: Results on iNaturalist: a query (purple) with the 9 most similar retrieved images, green for relevant images, red otherwise. Top line results with ROADMAP. Bottom line results with SmoothAP.

3.6 Conclusion.

In this chapter, we have introduced a general framework for rank losses optimization. It tackles two issues of rank losses optimization: 1) non-differentiability using smooth and upper bound rank approximation, 2) non-decomposability using an additional objective; providing a robust training for image retrieval models. We apply our framework to fine-grained image retrieval, by optimizing both AP and R@k. We show that using our framework outperforms other rank loss surrogates on several standard fine-grained image retrieval benchmarks. We also show that our framework sets state-of-the-art results for fine-grained image retrieval.

Fine-grained retrieval is the main task of image retrieval. However, models learned with *e.g.* ROADMAP lack robustness wrt. the mistakes that they commit and what they retrieve once the fine-grained instances are retrieved. This is illustrated on Fig. 3.17 in a failure where the model trained with ROADMAP (indicated as the "baseline") commits severe mistakes.

Figure 3.17: Examples of a failure case from a fine-grained model trained with ROADMAP *vs.* a mistake severity aware model, HAPPIER.

Our framework is general and can be used to optimized other metrics, for instance metrics that leverage hierarchical information to learn robust ranking. This will be illustrated in the next chapter with an extension of average precision to hierarchical rankings, and the NDCG.

Chapter 4

Hierarchical Image Retrieval for Robust Ranking

We have seen in the previous chapter how to optimize ranking based metrics commonly used to evaluate image retrieval. Yet, those metrics are limited to binary labels and do not take into account the mistake severity. In this chapter, we leverage hierarchical relations between labels to i) integrate errors' importance during training and ii) better evaluate rankings' robustness. We introduce a new hierarchical $\mathcal{H}$-AP metric that extends the AP beyond binary labels. We then show how to use the ROADMAP framework introduced previously to optimize $\mathcal{H}$-AP (HAPPIER) and NDCG (ROD-NDCG). We show that building a hierarchy of labels is a realistic goal by creating the first hierarchical landmarks retrieval dataset. We use a semi-automatic pipeline to extend with hierarchical labels the large scale Google Landmarks v2 dataset, publicly available at `github.com/cvdfoundation/google-landmark`. Extensive experiments on 7 datasets show that HAPPIER and ROD-NDCG significantly outperform state-of-the-art hierarchical retrieval algorithms, while being also on par with the most effective approaches when evaluating fine-grained ranking performance. Finally, we show that HAPPIER leads to a better organization of the embedding space and prevents most severe failure cases of non-hierarchical methods. Our code is publicly available at `github.com/elias-ramzi/HAPPIER`.

Content

4.1 Introduction.

We have seen in the previous chapter that ranking metrics used to evaluate image retrieval, *e.g.* AP and R@k are not differentiable and non-decomposable, leading us to introduce a framework to optimize them. However, these metrics are only defined for binary ($\oplus/\ominus$) labels, in practice relying on *fine-grained labels*: an image is negative as soon as it has not the same fine-grained label as the query. Binary metrics are by design unable to take into account the mistake severity of a ranking. On Fig. 4.1, some negative instances are "less negative" than others, *e.g.* given the "Brown Bear" query, "Polar bear" is more relevant than "Butterfly". However, AP is 0.9 for both the top and bottom rankings. Consequently, training on binary metrics (*e.g.* AP or R@k) develops no incentive to produce ranking such as the top row, and often produces rankings similar to the bottom one. This leads methods that optimize this metrics to lack robustness: they tend to make severe errors when they make errors.

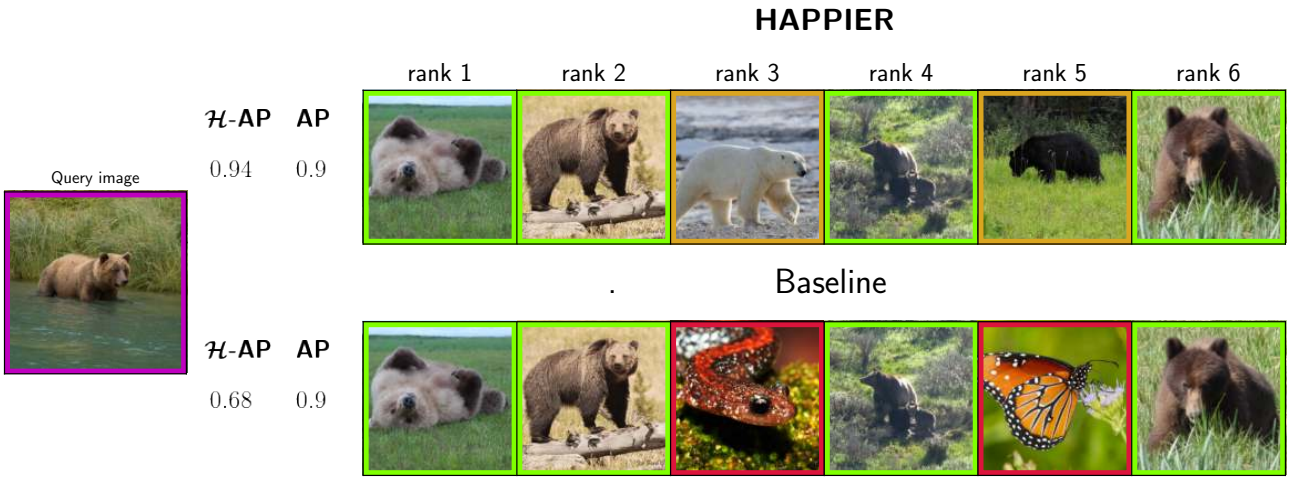


Figure 4.1: **Proposed hierarchical retrieval framework for pertinent image retrieval**. Standard ranking metrics based on binary labels, *e.g.* Average Precision (AP), assign the same score to the bottom and top row rankings (0.9). We introduce the $\mathcal{H}$-AP metric based on non-binary labels, that takes into account mistakes' severity. $\mathcal{H}$-AP assigns a smaller score to the bottom row (0.68) than the top one (0.94). HAPPIER maximizes $\mathcal{H}$-AP during training and thus explicitly supports to learn rankings similar to the top one, in contrast to binary ranking losses.

Hierarchical image retrieval can be used to mitigate this issue by taking into account non-binary similarities between labels by learning pairwise graded "relevance" scores. We introduce the hierarchical average precision, $\mathcal{H}$-AP, a new metric that extends the AP to non-binary settings. Using our optimization framework ROADMAP, we show how to optimize graded AP derivatives such as $\mathcal{H}$-AP, with HAPPIER, and the well known NDCG, leading to competitive

results for fine-grained image retrieval metrics, while outperforming by significant margins both fine-grained methods and hierarchical baselines when considering hierarchical metrics.

First, we define a new Hierarchical AP metric ($\mathcal{H}$-AP) that leverages the hierarchical tree between concepts and enables a fine weighting between errors in rankings. As shown in Fig. 4.1, $\mathcal{H}$-AP assigns a larger score (0.94) to the top ranking than to the bottom one (0.68). We show that $\mathcal{H}$-AP provides a consistent generalization of AP for the non-binary setting. We also introduce our HAPPIER$_\text{F}$ variant, giving more weights to fine-grained levels of the hierarchy.

Since $\mathcal{H}$-AP and NDCG, like AP, are non-differentiable metrics, we then use the ROADMAP framework introduced in Chapter 3 to directly optimize $\mathcal{H}$-AP with HAPPIER and NDCG with ROD-NDCG by gradient descent. Similarly to ROADMAP and ROD-R@k in Chapter 3, optimizing $\mathcal{H}$-AP and NDCG with the ROADMAP framework leads to robust optimization with good theoretical properties, in contrast with [79], [80] discussed in Sec. 2.3.2. Furthermore, by optimizing principled metrics, HAPPIER and ROD-NDCG outperforms other methods for hierarchical image retrieval [81], [82] discussed in Sec. 2.3.3, that are extended proxy and triplet-based losses.

Finally, we introduce the first hierarchical landmarks retrieval dataset, $\mathcal{H}$-GLDv2, extending the well-known Google Landmarks v2 landmarks retrieval (GLDv2) dataset [28]. While landmarks retrieval has been one of the most popular domain in image retrieval, it was lacking until now a hierarchical dataset. $\mathcal{H}$-GLDv2 is a large scale dataset with 1.4m images and three levels of hierarchies: including 100k unique landmarks, 78 super-categories and 2 final labels. These new labels are publicly available at `github.com/cvdfoundation/google-landmark`.

We validate HAPPIER and ROD-NDCG on seven IR datasets, including three standard datasets (Stanford Online Products [246] and iNaturalist-base/full [29]), three recent hierarchical datasets (DyML [81]), and our novel $\mathcal{H}$-GLDv2. We show that, when evaluating on hierarchical metrics (*e.g.* $\mathcal{H}$-AP), our hierarchical methods outperform state-of-the-art methods for fine-grained ranking [65], [66], [95], [164], the baselines and the latest hierarchical methods [81], [82], and only slightly under-performs *vs.* state-of-the-art IR methods at the fine-grained level (*e.g.* AP, R@1). HAPPIER$_\text{F}$ performs on par on fine-grained metrics while still outperforming fine-grained methods on hierarchical metrics.

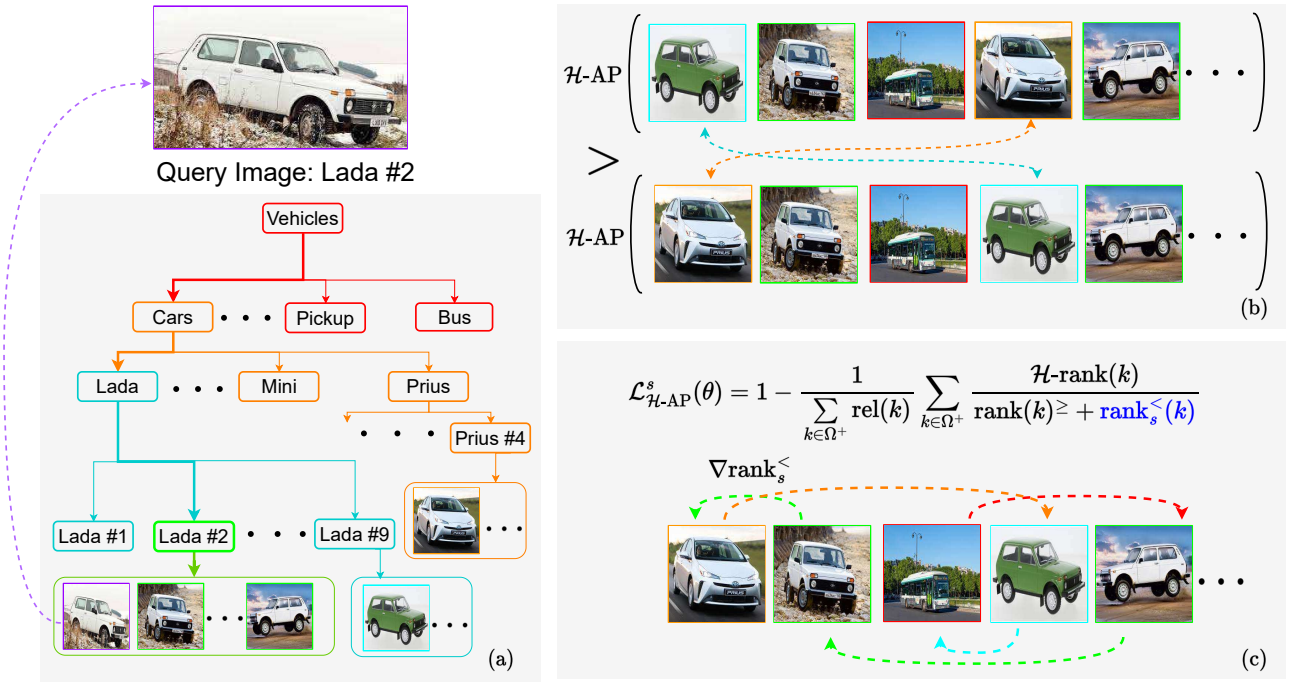


Figure 4.2: HAPPIER leverages a hierarchical tree representing the semantic similarities between concepts in (a) to introduce a new hierarchical metric, $\mathcal{H}$-AP in Eq. (4.3), see (b). $\mathcal{H}$-AP exploits the hierarchy to weight rankings' inversion: given the query image of a "Lada #2", $\mathcal{H}$-AP penalizes an inversion with a "Lada #9" less than with a "Prius #4". To directly train models with $\mathcal{H}$-AP, we carefully study the structure of the problem and introduce the $\mathcal{L}_{\text{Sup-}\mathcal{H}\text{-AP}}$ loss in Eq. (4.10), which provides a smooth upper bound of $\mathcal{L}_{\mathcal{H}\text{-AP}}$, see (c). We also train HAPPIER with the $\mathcal{L}^*_{\text{DG}}$ to enforce the partial ordering in stochastic optimization to mach the global ones.

4.2 Hierarchical Image Retrieval.

Standard metrics (*e.g.* AP or R@k) are only defined for binary labels: an image is either a negative or a positive for the query, in practice they are defined using fine-grained labels. These metrics are by design unable to take into account the mistake severity. To mitigate this issue, we propose to optimize a new ranking-based metric, $\mathcal{H}$-AP introduced in Sec. 4.2.2, that extends AP beyond binary labels, and the standard NDCG in Sec. 4.2.4. The Hierarchical Average Precision, $\mathcal{H}$-AP in Sec. 4.2.2, leverages a hierarchical tree (Fig. 4.2a) of labels. It is based on the hierarchical rank, $\mathcal{H}$-rank, and evaluates rankings so that more relevant instances are ranked before less relevant ones (Fig. 4.2b). We then show how to directly optimize $\mathcal{H}$-AP and NDCG with SGD using HAPPIER Sec. 4.2.3 and ROD-NDCG Sec. 4.2.4 using the framework presented in Chapter 3.

4.2.1 Additional training context.

We define in this section additional notations to Sec. 3.2.1 that we use in this chapter. We assume that we have access to a hierarchical tree defining semantic relationships between concepts, as in Fig. 4.2a. For a query $\boldsymbol{q}$, we partition the set of retrieved instances into $L+1$ disjoint subsets $\left\{\Omega^{(l)}\right\}_{l\in[\![0;L]\!]}$. $\Omega^{(L)}$ is the subset of the most similar instances to the query (*i.e.* fine-grained level): for $L=3$ and a "Lada #2" query (purple), $\Omega^{(3)}$ are the images of the same "Lada #2" (green) in Fig. 4.2a. The set $\Omega^{(l)}$ for $l<L$ contains instances with smaller relevance with respect to the query: $\Omega^{(2)}$ in Fig. 4.2a is the set of "Lada" that are not "Lada #2" (blue) and $\Omega^{(1)}$ is the set of "Cars" that are not "Lada" (orange). We also define $\Omega^{-}:=\Omega^{(0)}$, as the set of negative instances that share no common semantics with the query, *i.e.* the set of vehicles that are not "Cars" (in red) in Fig. 4.2a and $\Omega^{+}=\bigcup_{l=1}^{L}\Omega^{(l)}$. Given a query q, we define the relevance of $k\in\Omega^{(l)}$, $\mathrm{rel}(k):=\mathrm{rel}(x_k,q)$. For a query $\boldsymbol{q}\in\Omega$, we aim to order all $x_j\in\Omega$ so that more relevant (*i.e.* similar) instances are ranked before less relevant instances.

4.2.2 Hierarchical Average Precision.

As seen before, Average Precision (AP) is one of the most common metric in image retrieval, it is therefore natural to seek an extension to the hierarchical setting. Let us recall the definition of the average precision (AP) and the rank:

$$\mathrm{AP}=\frac{1}{|\Omega^{+}|}\sum_{k\in\Omega^{+}}\frac{\mathrm{rank}^{+}(k)}{\mathrm{rank}(k)},\ \text{with}\ \begin{cases}\mathrm{rank}(k)=1+\sum_{j\in\Omega}H(s_j-s_k)\\ \mathrm{rank}^{+}(k)=1+\sum_{j\in\Omega^{+}}H(s_j-s_k)\end{cases} \tag{4.1}$$

4.2.2.1 Extending AP to hierarchical image retrieval.

We propose an extension of AP that leverages non-binary labels. To do so, we extend the concept of rank^{+} to the hierarchical case with the concept of hierarchical rank, $\mathcal{H}$-rank:

$$\mathcal{H}\text{-rank}(k)=\mathrm{rel}(k)+\sum_{j\in\Omega^{+}}\min(\mathrm{rel}(k),\mathrm{rel}(j))\cdot H(s_j-s_k)\ . \tag{4.2}$$

Intuitively, $\min(\mathrm{rel}(k),\mathrm{rel}(j))$ corresponds to seeking the *closest ancestor* shared by instances k and j with the query in the hierarchical tree. As illustrated in Fig. 4.3, $\mathcal{H}$-rank induces a smoother penalization for instances that do not share the same fine-grained label as the query but still share some coarser semantics, which is not the case for the usual binary rank^{+}.

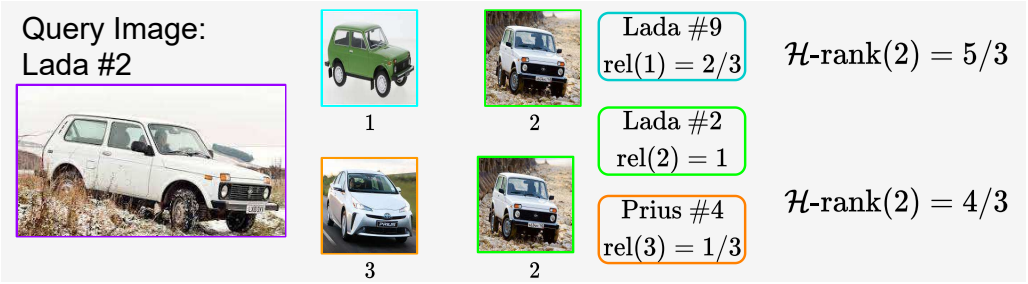


Figure 4.3: Given a "Lada #2" query, the top inversion is less severe than the bottom one. Indeed, on the top row, instance 1 is semantically closer to the query – as it is a "Lada"– than instance 3 on the bottom row. Indeed, instance 3's closest common ancestor with the query, "Cars", is farther in the hierarchical tree (see Fig. 4.2a). Because of that, $\mathcal{H}$-rank(2) is greater on the top row (5/3) than on the bottom row (4/3), leading to a greater $\mathcal{H}$-AP in Fig. 4.2b for the top row.

We detail in Fig. 4.5 how the $\mathcal{H}$-rank in Eq. (4.2) is computed in the example from Fig. 4.2b. Given a "Lada #2" query, we set the relevances as follows and illustrated on Sec. 4.2.2.1:

1. if $k \in \Omega^{(3)}$ (*i.e.* k is also a "Lada #2"), $\text{rel}(k) = 1$.
2. if $k \in \Omega^{(2)}$ (*i.e.* k is another model of "Lada"), $\text{rel}(k) = 2/3$.
3. if $k \in \Omega^{(1)}$ (k is a "Car"), $\text{rel}(k) = 1/3$.
4. Relevance of negatives (other vehicles) is set to 0.

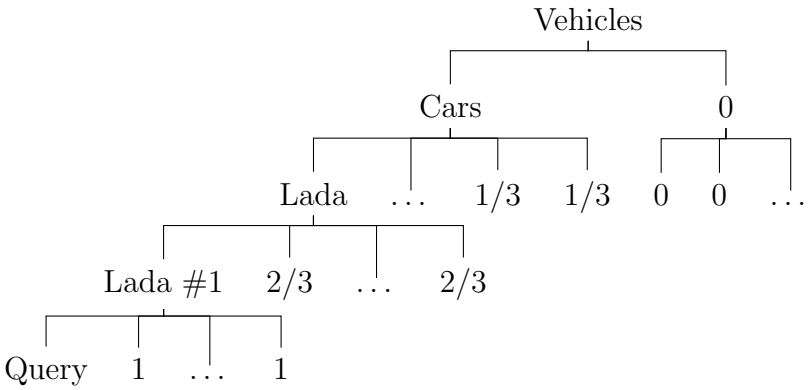


Figure 4.4: Toy relevances for the hierarchical tree of Fig. 4.2a.

In this instance, $\mathcal{H}$-rank(2) = 4/3 because rel(2) = 1 and $\min(\text{rel}(1), \text{rel}(2)) = \text{rel}(1) = 1/3$. Here, the closest common ancestor in the hierarchical tree shared by the query and instances 1 and 2 is "Cars". For binary labels, we would have $\text{rank}^+(2) = 1$; this would not take into

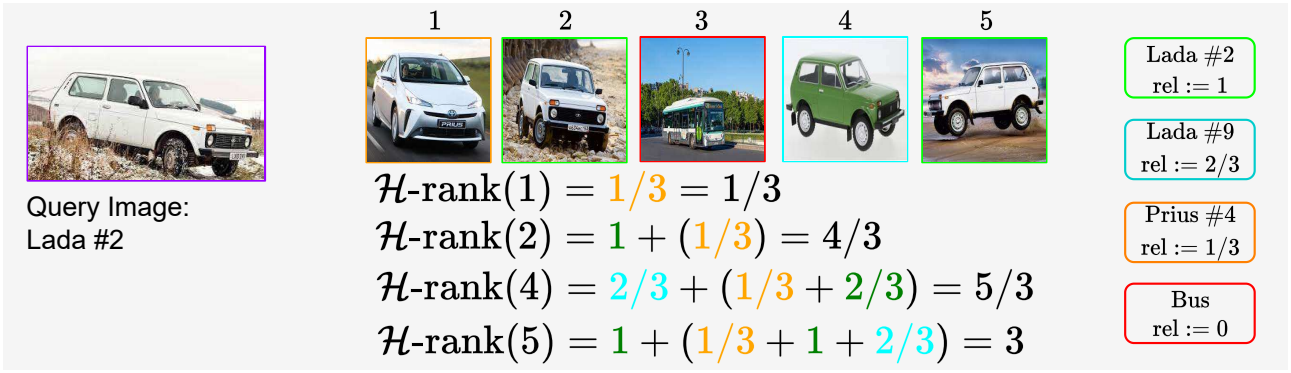


Figure 4.5: $\mathcal{H}$-rank for each retrieval results given a "Lada #2" query and the hierarchical tree of Fig. 4.2a.

account the semantic similarity between the query and the instance with rank 1.

From $\mathcal{H}$-rank in Eq. (4.2), we define the Hierarchical Average Precision, $\mathcal{H}$-AP, by replacing rank^+ of AP:

$$\mathcal{H}\text{-AP} = \frac{1}{\sum_{k\in\Omega^+} \text{rel}(k)} \sum_{k\in\Omega^+} \frac{\mathcal{H}\text{-rank}(k)}{\text{rank}(k)} \tag{4.3}$$

Eq. (4.3) extends the AP to non-binary labels. We replace rank^+ by our hierarchical rank $\mathcal{H}$-rank and the normalization term $|\Omega^+|$ is replaced by $\sum_{k\in\Omega^+} \text{rel}(k)$, which both represent the "sum of positives".

Normalization constant for $\mathcal{H}$-AP. When all instances are perfectly ranked, all instances j that are ranked before instance k ($s_j \geq s_k$) have a relevance that is higher or equal than k's, *i.e.* $\text{rel}(j) \geq \text{rel}(k)$ and $\min(\text{rel}(j), \text{rel}(k)) = \text{rel}(k)$. So, for each instance k:

$$\begin{aligned}
\mathcal{H}\text{-rank}(k) &= \text{rel}(k) + \sum_{j\in\Omega^+} \min(\text{rel}(k), \text{rel}(j)) \cdot H(s_j - s_k) \\
&= \text{rel}(k) + \sum_{j\in\Omega^+} \text{rel}(k) \cdot H(s_j - s_k) \\
&= \text{rel}(k) \cdot \left(1 + \sum_{j\in\Omega^+} H(s_j - s_k)\right) = \text{rel}(k) \cdot \text{rank}(k)
\end{aligned}$$

The total sum $\sum_{k\in\Omega^+} \frac{\mathcal{H}\text{-rank}(k)}{\text{rank}(k)} = \sum_{k\in\Omega^+} \text{rel}(k)$. Therefore, we need to normalize by $\sum_{k\in\Omega^+} \text{rel}(k)$ (to constrain $\mathcal{H}$-AP between 0 and 1). This results in the definition of $\mathcal{H}$-AP from Eq. (4.3).

$\mathcal{H}$-AP extends the desirable properties of the AP. It evaluates the quality of a ranking by:

1. penalizing inversions of instances that are not ranked in decreasing order of relevances with respect to the query.
2. giving stronger emphasis to inversions that occur at the top of the ranking.

$\mathcal{H}$-AP is a consistent generalization of AP. $\mathcal{H}$-AP is equivalent to AP in a binary setting ($L = 1$). Indeed, the relevance function can be set to 1 positive instances and 0 otherwise in the binary case, without loss of generality. Therefore, $\mathcal{H}\text{-rank}(k) = 1 + \sum_{j\in\Omega^+} H(s_j - s_k)$ which is the same definition as rank^+ in AP. Furthermore, the normalization constant of $\mathcal{H}$-AP, $\sum_{k\in\Omega^+} \text{rel}(k)$, is equal to the number of fine-grained instances in the binary setting, *i.e.* $|\Omega^+|$. This means that $\mathcal{H}\text{-AP} = \text{AP}$ in this case.

Link with recall and precision. One other property of AP is that it can be interpreted as the area under the precision-recall curve. $\mathcal{H}$-AP from Eq. (4.3) can also be interpreted as the area under a hierarchical-precision-recall curve by defining a Hierarchical Recall ($\mathcal{H}$-R@k) and a Hierarchical Precision ($\mathcal{H}$-P@k) as:

$$\mathcal{H}\text{-R@k} = \frac{\sum_{j=1}^{k} \text{rel}(j)}{\sum_{j\in\Omega^+} \text{rel}(j)} \tag{4.4}$$

$$\mathcal{H}\text{-P@k} = \frac{\sum_{j=1}^{k} \min(\text{rel}(j), \text{rel}(k))}{k \cdot \text{rel}(k)} \tag{4.5}$$

So that $\mathcal{H}$-AP can be re-written as:

$$\mathcal{H}\text{-AP} = \sum_{k=1}^{|\Omega|} (\mathcal{H}\text{-R@[k]} - \mathcal{H}\text{-R@[k-1]}) \cdot \mathcal{H}\text{-P@k} \tag{4.6}$$

Eq. (4.6) recovers Eq. (4.3), meaning that $\mathcal{H}$-AP generalizes this property of AP beyond binary labels.

$\mathcal{H}$-R@k is also a consistent generalization of R@k, indeed using a binary relevance we have:

$$\mathcal{H}\text{-R@k} = \frac{\sum_{j=1}^{k} \text{rel}(j)}{\sum_{j\in\Omega^+} \text{rel}(j)} = \frac{\sum_{j=1}^{k} \mathbb{1}(k \in \Omega^+)}{\sum_{j\in\Omega^+} \mathbb{1}(k \in \Omega^+)} = \frac{\#\text{ number of positive before k}}{|\Omega^+|} = \text{R@k}$$

Finally, $\mathcal{H}$-P@k is also a consistent generalization of P@k:

$$\mathcal{H}\text{-P@k} = \frac{\sum_{j=1}^{k} \min(\text{rel}(j), \text{rel}(k))}{k \cdot \text{rel}(k)} = \frac{\#\text{ number of positive before k}}{k} = \text{P@k}$$

4.2.2.2 Relevance function design.

Base relevance for $\mathcal{H}$-AP. The relevance $\text{rel}(k)$ defines how "similar" an instance $k \in \Omega^{(l)}$ is to the query q [197]. While $\text{rel}(k)$ might be given in some Information Retrieval datasets [256], [257], in our case we define it based on the hierarchical tree. We want to enforce the constraint that the relevance decreases when the closest common ancestor between the query and an instance is at a lower level in the hierarchical tree, *i.e.* $\text{rel}(k) > \text{rel}(k')$ for $k \in \Omega^{(l)}$, $k' \in \Omega^{(l')}$ and $l > l'$. To do so, we assign a total weight of $(l/L)^\alpha$ to each semantic level l, where $\alpha \in \mathbb{R}^+$ controls the decrease rate of similarity in the tree. For example, for $L = 3$ and $\alpha = 1$, the total weights for each level are 1, $\frac{2}{3}$, $\frac{1}{3}$ and 0. The instance relevance $\text{rel}(k)$ is normalized by the cardinal of $\Omega^{(l)}$:

$$\text{rel}(k) = \frac{(l/L)^\alpha}{|\Omega^{(l)}|} \text{ if } k \in \Omega^{(l)} \tag{4.7}$$

We set $\alpha = 1$ in Eq. (4.7) for the $\mathcal{H}$-AP metric and in our main experiments. Setting α to larger values supports better performances on fine-grained levels, as their relevances will relatively increase. This variant is denoted HAPPIER_F and discussed in Sec. 4.4.

Other definitions fulfilling the decreasing similarity behavior in the tree are possible. An interesting option for the relevance enables to recover a weighted sum of AP, denoted as $\sum w\text{AP} := \sum_{l=1}^{L} w_l \cdot \text{AP}^{(l)}$ [96], *i.e.* the weighted sum of AP is a particular case of $\mathcal{H}$-AP and detailed below in Property 1.

Link between $\mathcal{H}$-AP and the weighted average of AP. Let us define the AP for the semantic level $l \geq 1$ as the binary AP with the set of positives being all instances that belong to the same level, *i.e.* $\Omega^{+,l} = \bigcup_{q=l}^{L} \Omega^{(q)}$:

$$\text{AP}^{(l)} = \frac{1}{|\Omega^{+,l}|} \sum_{k \in \Omega^{+,l}} \frac{\text{rank}^{+,l}(k)}{\text{rank}(k)}, \ \text{rank}^{+,l}(k) = 1 + \sum_{j \in \Omega^{+,l}} H(s_j - s_k) \tag{4.8}$$

Property 1 *For any relevance function* $\text{rel}(k) = \sum_{p=1}^{l} \frac{w_p}{|\Omega^{+,q}|}$, $k \in \Omega^{(l)}$, *with positive weights* $\{w_l\}_{l \in [\![1;L]\!]}$ *such that* $\sum_{l=1}^{L} w_l = 1$*:*

$$\mathcal{H}\text{-}AP = \sum_{l=1}^{L} w_l \cdot AP^{(l)} \tag{4.9}$$

i.e. *$\mathcal{H}$-AP is equal to the weighted average of the AP at all semantic levels.*

We give the proof in annex Appendix B.1.

Application to trademark logo retrieval. Trademarked logos are multi-label images, *i.e.* multiple different concepts can occur on an image. These concepts follow the Vienna classification[1], a system that classifies the figurative elements of trademarked logos. Furthermore, the Vienna classification is also a hierarchical classification. For instance Category 1 groups all "CELESTIAL BODIES, NATURAL PHENOMENA, GEOGRAPHICAL MAPS", which is further divided in "STARS, COMETS", "SUN" *etc.* finally the fine-grained categories from "STARS, COMETS" are "One star", "Concentric stars" *etc.*. To adapt the relevance of the $\mathcal{H}$-AP to trademark logo retrieval we want to take into account both the number of concepts matching between two images and the proximity of those labels.

We consider two images q and k that are annotated with a set of label $\{y_q^1, \dots, y_q^n\}$ and $\{y_k^1, \dots, y_k^n\}$. We then use the notion of "lowest common ancestor" (LCA) which is a distance in the hierarchical tree. The lower the LCA, the closest two labels are. For instance the LCA between two "Lada # 1" is 0, between a "Lada #1" and "Lada #2" is 1, between a "Lada #1" and "Cars" that are not "Lada" is 2, *etc.*

We then define $0 \leq \text{rel}(q, k) \leq 1$ as follows:

Algorithm 1 Computation of Coexya's relevance.

input : q with labels $\{y_q^1, \dots, y_q^n\}$. k with labels $\{y_k^1, \dots, y_k^m\}$. L the number of hierarchical levels
output: $0 \leq \text{rel}(q, k) \leq 1$
$\text{rel}(q, k) = 0$
for y_q^i **do**
  Select $l_k^* = L - \min_{y_k^i} \text{LCA}(y_q^i, y_k^i)$
  $\text{rel}(q, k) \stackrel{+}{=} \frac{l_k^*}{n \cdot L}$
end

This relevance function computes the sum for each label of an individual relevance computed with Vienna classification's hierarchical tree.

4.2.3 Direct optimization of $\mathcal{H}$-AP.

$\mathcal{H}$-AP has the same drawbacks as other ranking loss when it comes to optimization. It is neither differentiable nor decomposable. To optimize deep models with this metric, we follow the framework presented in Chapter 3. We first define our surrogate loss using SupRank to optimize $\mathcal{H}$-AP:

[1]Vienna classification: https://nivilo.wipo.int/vienna9/index.htm?lang=EN

$$\mathcal{L}_{\text{Sup-}\mathcal{H}\text{-AP}}(\boldsymbol{\theta}) = 1 - \frac{1}{M}\sum_{i=1}^{M} \frac{1}{\sum\limits_{k\in\Omega_i^+} \text{rel}(k)} \sum_{k\in\Omega_i^+} \frac{\mathcal{H}\text{-rank}(k)}{\text{rank}^+(k) + \text{rank}_s^-(k)} \tag{4.10}$$

Note that in the hierarchical case $\text{rank}_s^-(k)$ is the number of instances of relevances $< \text{rel}(k)$ meaning that it may contain images that are similar to some extent to the query. Finally, our ranking loss, **H**ierarchical **A**verage **P**recision training for **P**ertinent **I**mag**E** **R**etrieval (HAPPIER), is obtained by adding the decomposability constraint $\mathcal{L}_{\text{DG}}^*$:

$$\mathcal{L}_{\text{HAPPIER}}(\boldsymbol{\theta}) = (1-\lambda)\cdot\mathcal{L}_{\text{Sup-}\mathcal{H}\text{-AP}}(\boldsymbol{\theta}) + \lambda\cdot\mathcal{L}_{\text{DG}}^*(\boldsymbol{\theta}) \tag{4.11}$$

4.2.4 Application to the NDCG.

Although studying the extension of AP to the hierarchical setting with $\mathcal{H}$-AP is natural because AP is one of the most common metric in image retrieval, we propose to study the NDCG as well. The NDCG [143], [144] is one of the most common metric in information retrieval. The NDCG, discussed in Eq. (2.11), uses a relevance that can be graded:

$$\begin{aligned}
\text{DCG}_i &= \sum_{k\in\Omega_i^+} \frac{\text{rel}(k)}{\log_2(1+\text{rank}^+(k)+\text{rank}^-(k))} \\
\text{iDCG}_i &= \max_{\text{rank}} \text{DCG}_i \\
\text{NDCG} &= \frac{1}{M}\sum_{i=1}^{M} \frac{\text{DCG}_i}{\text{iDCG}_i}
\end{aligned} \tag{4.12}$$

The DCG decreases the overall score brought by one instance when its ranking increases. When the perfect ranking is achieved DCG = iDCG thus NDCG = 1. We choose a relevance function from the information retrieval community for the NDCG: $\text{rel}(k) = 2^l - 1$, if $k \in \Omega^{(l)}$. The exponentiation is a standard procedure [144] as it allows putting more emphasis on instances of higher relevance. We then use our SupRank surrogate to approximate the NDCG:

$$\text{DCG}_{i,s} = \sum_{k\in\Omega_i^+} \frac{\text{rel}(k)}{\log_2(1+\text{rank}^+(k)+\text{rank}_s^-(k))} \tag{4.13}$$

We then define the $\mathcal{L}_{\text{Sup-NDCG}}$ loss as:

$$\mathcal{L}_{\text{Sup-NDCG}}(\boldsymbol{\theta}) = 1 - \frac{1}{M}\sum_{i=1}^{M} \frac{\text{DCG}_{i,s}}{\text{iDCG}_i} \tag{4.14}$$

Note that, once again, our surrogate loss, $\mathcal{L}_{\text{Sup-NDCG}}$ is an upper bound on the true loss $1 - \text{NDCG}$. Indeed, $\text{rank}_s^-(k) > \text{rank}^-(k)$ so $\text{DCG}_{i,s} < \text{DCG}_i$. Finally, our training loss, including the decomposability constraint, is:

$$\mathcal{L}_{\text{ROD-NDCG}}(\boldsymbol{\theta}) = (1 - \lambda) \cdot \mathcal{L}_{\text{Sup-NDCG}}(\boldsymbol{\theta}) + \lambda \cdot \mathcal{L}^*_{\text{DG}}(\boldsymbol{\theta}) \tag{4.15}$$

4.3 Hierarchical Landmark dataset.

Figure 4.6: Image taken from [28]. Place de la Concorde – Significant scale changes, historical photographs and paintings. The dataset can be explored with `gld-v2/web/index.html`.

As stated in Sec. 4.2.1, in this chapter, we assume that we have access to hierarchical labels. In this section, we show how to create a hierarchy in practice.

To start, hierarchical trees are available for many datasets, such as CUB-200-2011 [247], Cars196 [258], InShop [259], Stanford Online Products [246] composed of 22634 objects (Id) classified into 12 (coarse) categories (*e.g.* bikes, coffee makers *etc.*), and notably *large-scale* ones such as iNaturalist [29], the three DyML datasets [81] and ImageNet [25]. Coarse labels are also less difficult to obtain than fine-grained ones, since hierarchical relations can be semi-automatically obtained by grouping fine-grained labels. This was previously done by [195] or by using the large lexical database WordNet [78] *e.g.* for ImageNet in [25] and for the SUN

database in [260].

Whereas human-made and natural landmarks are one of the most popular domains for image retrieval [28], [255], [261]–[263], there is no prior hierarchical landmark dataset. Google Landmarks Dataset v2 (GLDv2) is the largest and most diverse landmark dataset [28]. It is annotated at a fine-grained level for specific monument, *e.g.* "Place de la Concorde" on Fig. 4.6. It features diverse points of view, taken from possibly very different times. Its original version consists of over $5M$ images and $200k$ distinct instance labels, it was subsequently "cleaned" [264], resulting in a dataset of $1.5M$ images and $80k$ distinct instance labels.

In this chapter, we rely on coarse labels that can be found on Wikimedia Commons[2] to create a hierarchical extension of the GLDv2 dataset, $\mathcal{H}$-GLDv2. It is the first of its kind hierarchical landmark dataset. $\mathcal{H}$-GLDv2 is a large scale dataset with 1.4m images and three levels of hierarchies: including 100k unique landmarks, 78 super-categories and 2 final labels. The labels are publicly available at `github.com/cvdfoundation/google-landmark`.

4.3.1 Scraping Wikimedia Commons.

The landmarks from GLDv2 are sourced from Wikimedia Commons, the world's largest crowdsourced collection of landmark photos. Many of the landmarks in GLDv2 can be associated to super categories by leveraging the "Instance of" annotations available in Wikimedia Commons – see Fig. 4.7. Out of the original $203k$ landmarks in GLDv2-train, we were able to scrape on Wikimedia Commons super categories for $129.1k$. For the $101k$ landmarks in GLDv2-index, we were able to scrape super categories for $68.1k$. We apply a lightweight manual cleaning process to remove landmarks assigned to more than one super category and those with irrelevant super categories (*e.g.*, super categories named "Wikimedia category" or "Wikimedia disambiguation page"). Approximately 0.25% of landmarks end up being removed in this process, leading to a total number of selected landmarks of $128.8k$ and $67.9k$ for the GLDv2-train and GLDv2-index dataset splits, respectively. The number of unique scraped super categories is $5.7k$.

4.3.2 Post-processing super categories.

The scraped super categories are noisy and do not have the same level of granularity, *e.g.* "church building" *vs.* "church building (1172–1954)", which comes from the different sources that created the Wikimedia Commons pages. To mitigate this issue, after the scraping, we

[2]GlDv2 landmarks are crawled from: `https://commons.wikimedia.org/wiki/Accueil`

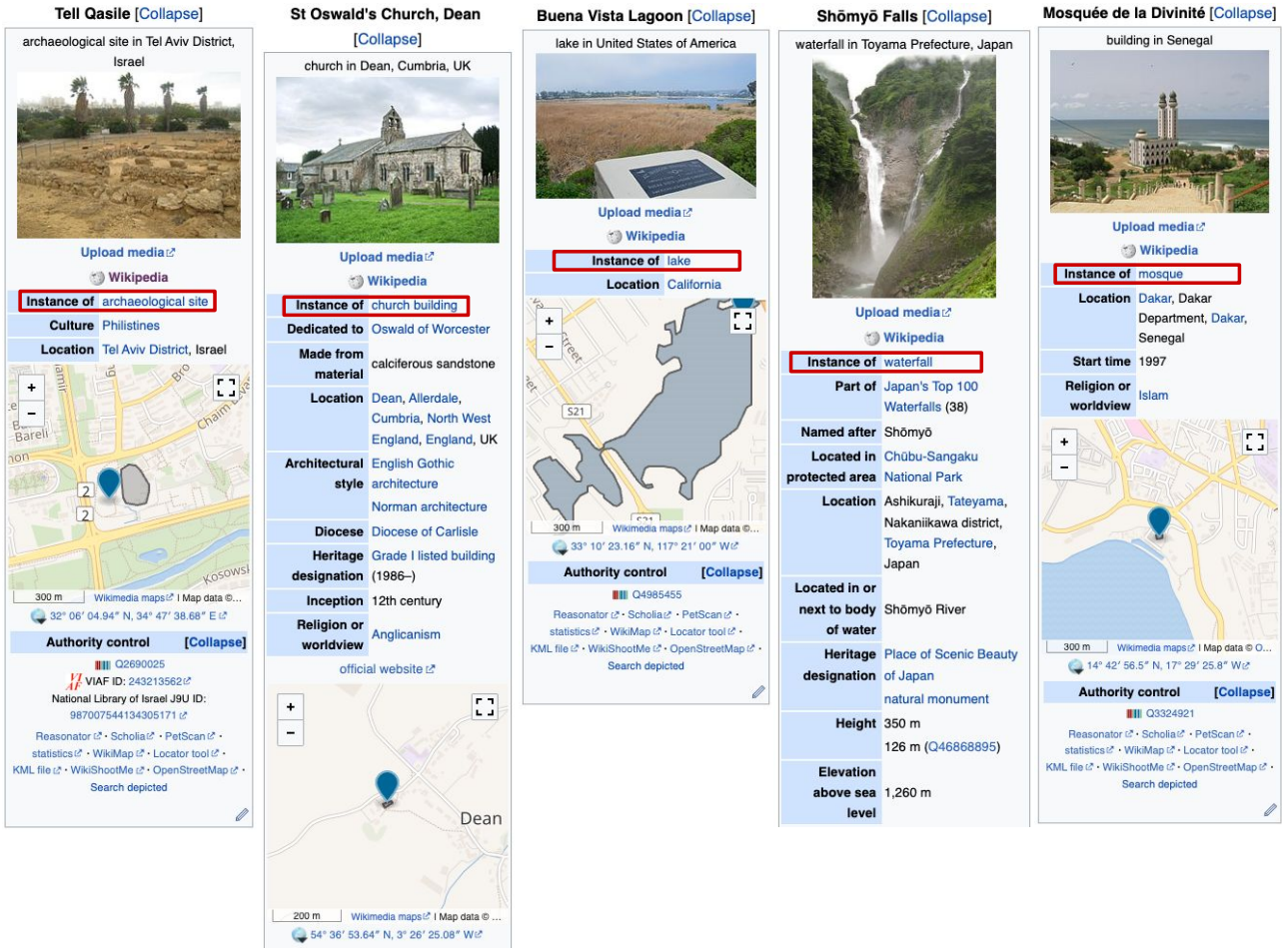


Figure 4.7: Screen captures of Wikimedia Commons web-pages. It depicts the "Instance of" (within red rectangles), from which we collect hierarchical landmark labels: *e.g. lake*, *waterfall*, *mosque*.

perform a two-step post-processing to obtain the final super categories.

1. **K-means clustering:** We first encode all the labels using the CLIP [12] textual encoder, as it creates good embeddings for textual content similarly to S-BERT [265]. We perform a k-means with $K = 12$ on the latent representations. This initial clustering allows showing different prominent categories, *e.g.* "Church", "Castle" *etc.*

2. **Manual verification:** We manually assess the obtained clusters based on the scraped label names. We create semantic groups by dividing the k-means clusters into sub-clusters. This leads to 78 super categories that we further group into human-made and natural landmarks.

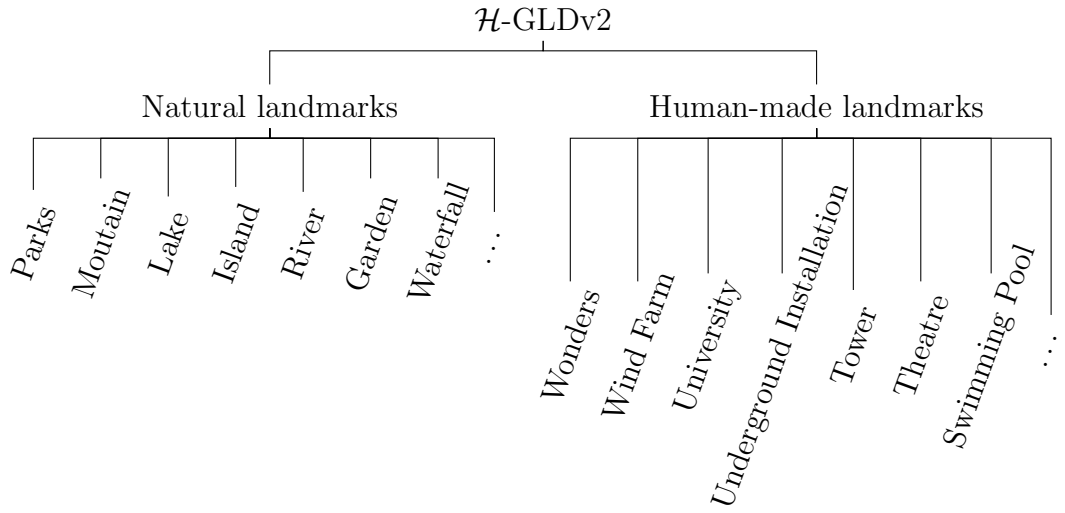


Figure 4.8: We illustrate for both final labels, "Natural landmarks" and "Human-made landmarks" the 7 most represented super category.

We illustrate some of the created groups in Figs. 4.9a to 4.9c. These new hierarchical labels are released under the CC BY 4.0 license.

(a) Waterfall. (b) Bridge. (c) Castle.

Figure 4.9: Figs. 4.9a to 4.9c illustrate some of the super categories of our $\mathcal{H}$-GLDv2 dataset.

4.3.3 Discussion and limitations.

$\mathcal{H}$-GLDv2 is a large scale dataset, we were thus not able to manually check all images. This leads to a dataset that can have some noise. We release along with $\mathcal{H}$-GLDv2 the scraped labels to allow further work on the "super categories". Furthermore, that there is an imbalance

between super categories, see Fig. 4.10, that comes from unbalance of the classes represented in GLDv2 [28].

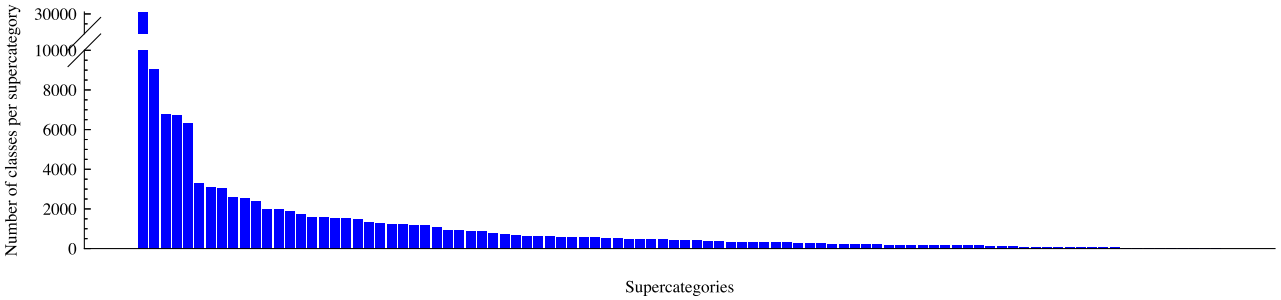


Figure 4.10: We can observe a large imbalance between the super categories of our $\mathcal{H}$-GLDv2. This is notably due to the super category "Christian religious building" that groups more than $30k$ distinct landmarks.

A fundamental question on the hierarchy that we had to address in order to annotate the dataset: should the hierarchy be consistent with the visual features or the semantic of human defined concepts? As discussed in Sec. 2.3 we made the choice of creating a hierarchy based on human-defined concepts, hoping to build a dataset that would closely align with human preferences. This lead to $\mathcal{H}$-GLDv2 being a hard task, as there is an ambiguity of some super categories. For instance, the bottom right image of Fig. 4.9b is labeled as "Bridge", however it could be labeled as "River", another super category.

4.4 Experiments.

4.4.1 Experimental setup.

Datasets. We use the standard benchmark Stanford Online Products [246] (SOP) with two levels of hierarchy ($L = 2$), *i.e.* fine-grained ID (*e.g.* bike # 100) and the categories (*e.g.* bikes, sofa, tea pot *etc.*); and iNaturalist-2018 [29] with the standard splits from [69] in two settings: i) iNat-base with two levels of hierarchy ($L = 2$), *i.e.* the Species (fine-grained) and groups of Species such as Plants, Mushrooms, Animals ii) iNat-full with the full biological taxonomy composed of 7 levels ($L = 7$)[3].

We also evaluate on the recent dynamic metric learning (DyML) datasets introduced in [81]. The DyML benchmark is composed of three datasets with 3 semantic levels ($L = 3$). DyML-V depicts vehicles, it is annotated at the fine-grained with a vehicle ID, then a "model" (*e.g.*

[3]Biological taxonomy: https://en.wikipedia.org/wiki/Taxonomy_(biology)

Toyota Camry, Honda Accord, Audi A4) and finally a "body type" (*e.g.* car, SUV, microbus, pickup). DyML-A depicts animals annotated at the fine-grained level with a specie, then one of 47 categories corresponding to "order", "family" or "genus", and finally one of 5 "classes" for the coarse level, see Footnote 3 for the biological taxonomy. DyML-P that depicts online products, the dataset is a subset from iMaterialist-2019[4], fine-grained product are organized in a hierarchical structure[5]. The training set has three levels of semantic ($L = 3$), and each image is annotated with the label corresponding to each level (like SOP and iNat-base/full), however the test protocol is different. At test time for each dataset there are three sub-datasets, each sub-dataset aims at evaluating the model on a specific hierarchical level (*e.g.* "Fine"), so we can only compute binary metrics on each sub-dataset.

Metrics. For SOP and iNat, we evaluate the models based on three hierarchical metrics: $\mathcal{H}$-AP – which we introduced in Sec. 4.2.2 – the NDCG and the Average Set Intersection (ASI):

$$\begin{aligned} \text{SI}(n) &= \frac{|\{a_1, \dots, a_n\} \cap \{b_1, \dots, b_n\}|}{n} \\ \text{ASI} &= \frac{1}{N} \sum_{n=1}^{N} SI(n) \end{aligned} \tag{4.16}$$

The ASI [266] measures at each rank $n \leq N$ the set intersection proportion (SI) between the ranked list $a_1, \dots, a_N$ and the ground truth ranking $b_1, \dots, b_N$, with N the total number of positives. As it compares intersections, the ASI can naturally take into account the different levels of semantic.

We also report the AP for each semantic level l by considering that all instances with semantic levels $\geq l$ are positives:

$$\text{AP}^{(l)} = \sum_{k \in \bigcup_{q=l}^{L} \Omega^{(q)}} \frac{\text{rank}^l(k)}{\text{rank}(k)}, \text{ where } \text{rank}^l(k) = 1 + \sum_{j \in \bigcup_{q=l}^{L} \Omega^{(q)}} H(s_j - s_k) \tag{4.17}$$

Finally, for DyML, we follow the evaluation protocols of [81] and compute AP, ASI and R@1 on each semantic level before averaging them. We cannot compute $\mathcal{H}$-AP or NDCG on those datasets, as the full hierarchical tree is not available on the test set.

[4] iMaterialist-2019 dataset: https://github.com/msight-tech/imaterialist-product-2019
[5] the original hierarchical structure of iMaterialist-2019 can be observed at: product_tree.pdf.

Baselines. We compare HAPPIER and ROD-NDCG to several recent image retrieval methods optimized at the fine-grained level, which represent strong baselines for IR when training with binary labels: Triplet SH (TL_{SH}) [65], NormSoftMax (NSM) [66], ProxyNCA++ (NCA++) [164] and ROADMAP [95]. We also benchmark against hierarchical methods obtained by summing binary losses at different levels (denoted by Σ), and with respect to the recent hierarchical losses CSL [81] and CLCD [82].

Implementation details. Unless specified otherwise, all reported results are obtained with $\alpha = 1$ in Eq. (4.7) and $\lambda = 0.1$ for $\mathcal{L}_{\text{HAPPIER}}$. We study the impact of these parameters in Sec. 4.4.3.

SOP & iNat-base/full. Our model is a ResNet-50 [45] pre-trained on ImageNet, to which we append a `LayerNormalization` layer [267] with no affine parameters after the (average) pooling and a Linear layer that reduces the embeddings size from 2048 to 512. We use the Adam [268] optimizer with a base learning rate of $1e^{-5}$ and weight decay of $1e^{-4}$ for SOP and a base learning rate of $1e^{-5}$ and weight decay of $4e^{-4}$ for iNat-base/full. The learning rate is decreased using cosine annealing decay, for 75 epochs on SOP and 100 epochs on iNat-base/full. We "warm up" our model for 5 epochs, *i.e.* the pre-trained weights are not optimized. We use standard data augmentation: `RandomResizedCrop` and `RandomHorizontalFlip`, with a final crop size of 224, at test time we use `CenterCrop`. We use a fixed batch size of 256 and use the hard sampling strategy from [173] on SOP and the standard class balanced sampling [66] (4 instances per class) on iNat-base/full.

DyML. We use a ResNet-34 [45] randomly initialized on DyML-V&A and pre-trained on ImageNet for DyML-P, following [81]. We use an SGD optimizer with Nesterov momentum (0.9), a base learning rate of 0.1 on DyML-V&A and 0.01 on DyML-P with a weight decay of $1e^{-4}$. We use cosine annealing decay to reduce the learning rate for 100 epochs on DyML-V&A and 20 on DyML-P. We use the same data augmentation and random seed as for SOP and iNat-base. We also use the class balanced sampling (4 instances per class) with a fixed batch size of 256.

$\mathcal{H}$-GLDv2. We use a ResNet-101 with GeM pooling [64] and initialize a linear projection with a PCA [67]. We use a batch size of 256 and train for $\sim$ 55k steps with Adam and a learning rate of 10^{-5} decayed using a cosine schedule. We report the mAP@100 [28], and the hierarchical metrics $\mathcal{H}$-AP, ASI and NDCG.

4.4.2 Experimental Results.

4.4.2.1 Hierarchical results.

We first evaluate HAPPIER on hierarchical metrics. On Tab. 4.1, we notice that HAPPIER significantly outperforms methods trained on the fine-grained level only, with a gain on $\mathcal{H}$-AP over the best performing methods of +16.1pt on SOP, +13pt on iNat-base and +12.7pt on iNat-full. HAPPIER also exhibits significant gains compared to hierarchical methods. On $\mathcal{H}$-AP, HAPPIER has important gains on all datasets (*e.g.* +6.3pt on SOP, +4.2pt on iNat-base over the best competitor), but also on ASI and NDCG. This shows the strong generalization of the method on standard metrics. Compared to the recent CSL loss [81], we observe a consistent gain over all metrics and datasets, *e.g.* +6pt on $\mathcal{H}$-AP, +8pt on ASI and +2.6pt on NDCG on SOP. This shows the benefits of optimizing a well-behaved hierarchical metric compared to an ad-hoc proxy method.

Table 4.1: Comparison of HAPPIER on SOP and iNat-base/full when using hierarchical metrics. Best results in **bold**, second best underlined.

	Method	SOP			iNat-base			iNat-full		
		$\mathcal{H}$-AP	ASI	NDCG	$\mathcal{H}$-AP	ASI	NDCG	$\mathcal{H}$-AP	ASI	NDCG
Fine	Triplet SH [65]	42.2	22.4	78.8	39.5	63.7	91.5	36.1	59.2	89.8
	NSM [66]	42.8	21.1	78.3	38.0	51.6	88.9	33.3	51.7	88.2
	NCA++ [164]	43.0	21.5	78.4	39.5	57.0	90.1	35.3	55.7	89.0
	Smooth-AP [69]	42.9	20.6	78.2	41.3	64.2	91.9	37.2	60.1	90.1
	ROADMAP [95]	43.3	19.1	77.9	40.3	61.0	91.2	34.7	59.6	89.5
Hier.	ΣTL$_{\text{SH}}$ [65]	53.1	53.3	89.2	44.0	87.4	96.4	39.9	<u>85.5</u>	92.0
	ΣNSM [66]	50.4	49.7	87.0	47.9	75.8	94.4	<u>46.9</u>	74.2	**93.8**
	ΣNCA++ [164]	49.5	52.8	87.8	48.9	78.7	95.0	44.7	74.3	92.6
	CSL [81]	52.8	57.9	88.1	<u>50.1</u>	**89.3**	<u>96.7</u>	45.1	84.9	93.0
	ROD-NDCG	<u>58.3</u>	<u>65.0</u>	<u>91.1</u>	<u>53.1</u>	87.8	<u>96.6</u>	44.8	81.1	93.1
	HAPPIER	**59.4**	**65.9**	**91.5**	**54.3**	**89.3**	**96.9**	**47.9**	**87.2**	**93.8**

On Tab. 4.2, we evaluate HAPPIER on the recent DyML benchmarks. HAPPIER again shows significant gains in mAP and ASI compared to methods only trained on fine-grained labels, *e.g.* +9pt in mAP and +10pt in ASI on DyML-V. HAPPIER also outperforms other hierarchical baselines: +4.8pt mAP on DyML-V, +0.9 on DyML-A and +1.8 on DyML-P. In R@1, HAPPIER performs on par with the best methods on DyML-V and outperforms hierarchical baselines by a large margin on DyML-P: 63.7 *vs.* 60.8 for ΣNSM. Interestingly, HAPPIER

Table 4.2: Performance comparison on Dynamic Metric Learning benchmarks [81].

	Method	DyML-Vehicle			DyML-Animal			DyML-Product		
		mAP	ASI	R@1	mAP	ASI	R@1	mAP	ASI	R@1
Fine	TL_{SH} [65]	26.1	38.6	84.0	37.5	46.3	66.3	36.32	46.1	59.6
	NSM [66]	27.7	40.3	88.7	38.8	48.4	<u>69.6</u>	35.6	46.0	57.4
	Smooth-AP [69]	27.1	39.5	83.8	37.7	45.4	63.6	36.1	45.5	55.0
	ROADMAP [95]	27.1	39.6	84.5	34.4	42.6	62.8	34.6	44.6	62.5
Hier.	ΣTL_{SH} [65]	25.5	38.1	81.0	38.9	47.2	65.9	36.9	46.3	58.5
	ΣNSM [66]	32.0	45.7	**89.4**	42.6	50.6	**70.0**	36.8	46.9	60.8
	CSL [81]	30.0	43.6	87.1	40.8	46.3	60.9	31.1	40.7	52.7
	CLCD-ACR [82]	16.0	42.9	-	36.0	57.1	-	29.4	58.8	-
	CLCD-ICR [82]	16.6	43.7	-	35.7	56.0	-	30.2	59.5	-
	ROD-NDCG	<u>36.1</u>	<u>49.2</u>	88.7	<u>43.2</u>	<u>50.7</u>	69.1	**38.9**	**48.6**	**65.4**
	HAPPIER	**37.0**	**49.8**	<u>89.1</u>	**43.8**	**50.8**	68.9	<u>38.0</u>	<u>47.9</u>	<u>63.7</u>

also consistently outperforms CSL [81] on its own datasets.

4.4.2.2 Detailed evaluation.

Tabs. 4.3 and 4.4 shows the different methods' performances on all semantic hierarchy levels. We evaluate both HAPPIER and HAPPIER_F ($\alpha > 1$ for Eq. (4.7) in Sec. 4.2.2), with $\alpha = 5$ on SOP and $\alpha = 3$ on iNat-base/full. HAPPIER optimizes the overall hierarchical performances, while HAPPIER_F is meant to be optimal at the fine-grained level while still optimizing coarser levels.

On Tab. 4.3, we observe that HAPPIER gives the best performances at the coarse level, with a significant boost compared to fine-grained methods, *e.g.* +43.9pt AP compared to the best non-hierarchical TL_{SH} [65] on SOP. HAPPIER even outperforms the best fine-grained methods in R@1 on iNat-base, but is slightly below on SOP. HAPPIER_F performs on par with the best methods at the finest level on SOP, while further improving performances on iNat-base, and still significantly outperforms fine-grained methods at the coarse level.

The satisfactory behavior of HAPPIER and HAPPIER_F are confirmed and even more pronounced on iNat-full (Tab. 4.4): HAPPIER gives the best results on coarser levels (from "Order"), while being very close to the best results on finer ones. HAPPIER_F gives the best results at the finest levels, even outperforming very competitive fine-grained baselines.

Again, note that HAPPIER outperforms the hierarchical CSL [81] on all semantic levels and

Table 4.3: Comparison of HAPPIER *vs.* methods trained only on fine-grained labels on SOP and iNat-base. Metrics are reported for both semantic levels.

		SOP			iNat-base		
		Fine		Coarse	Fine		Coarse
	Method	R@1	AP	AP	R@1	AP	AP
Fine	TL_{SH} [65]	79.8	59.6	14.5	66.3	33.3	51.5
	NSM [66]	81.3	61.3	13.4	70.2	<u>37.6</u>	38.8
	NCA++ [164]	81.4	61.7	13.6	67.3	37.0	44.5
	Smooth-AP [69]	81.3	61.7	13.4	67.3	35.2	53.1
	ROADMAP [95]	**82.2**	**62.5**	12.9	69.3	35.1	50.4
Hier.	CSL [81]	79.4	58.0	<u>45.0</u>	62.9	30.2	<u>88.5</u>
	HAPPIER	81.0	60.4	**58.4**	<u>70.7</u>	36.7	**88.6**
	HAPPIER$_F$	<u>81.8</u>	<u>62.2</u>	36.0	**71.6**	**37.8**	78.8

Table 4.4: Comparison of HAPPIER *vs.* methods trained only on fine-grained labels on iNat-Full. Metrics are reported for all 7 semantic levels.

	Method	Species		Genus	Family	Order	Class	Phylum	Kingdom
		R@1	AP	AP	AP	AP	AP	AP	AP
Fine	TL_{SH} [65]	66.3	33.3	34.2	32.3	35.4	48.5	54.6	68.4
	NSM [66]	<u>70.2</u>	**37.6**	<u>38.0</u>	31.4	28.6	36.6	43.9	63.0
	NCA++ [164]	67.3	37.0	37.9	33.0	32.3	41.9	48.4	66.1
	Smooth-AP [69]	67.3	35.2	36.3	33.5	35.0	49.3	55.8	69.9
	ROADMAP [95]	69.3	35.1	35.4	29.3	29.6	46.4	54.7	69.5
Hier.	CSL [81]	59.9	30.4	32.4	36.2	50.7	<u>81.0</u>	<u>87.4</u>	<u>91.3</u>
	HAPPIER	<u>70.2</u>	36.0	37.0	<u>38.0</u>	**51.9**	**81.3**	**89.1**	**94.4**
	HAPPIER$_F$	**70.8**	**37.6**	**38.2**	**38.8**	<u>50.9</u>	76.1	82.2	83.1

datasets on Tabs. 4.3 and 4.4, *e.g.* +5pt on the fine-grained AP ("Species") and +3pt on the coarsest AP ("Kingdom") on Tab. 4.4.

4.4.2.3 Hierarchical landmark results.

In Tab. 4.5 we report the first results of ROADMAP and HAPPIER *vs.* other fine-grained methods and hierarchical methods on our $\mathcal{H}$-GLDv2 dataset. Tab. 4.5 demonstrates once again the interest of our AP surrogate, ROADMAP and HAPPIER $_F$ perform the best on the fine-grained metric mAP@100. Furthermore, HAPPIER has the best hierarchical results. It

Table 4.5: Comparison of ROADMAP and HAPPIER *vs.* baselines on our $\mathcal{H}$-GLDv2.

	Method	mAP@100	$\mathcal{H}$-AP	ASI	NDCG
Fine	SoftBin [67]	39.0	35.2	74.6	94.4
	Smooth-AP [69]	42.5	37.3	76.9	94.7
	R@k [70]	41.6	36.8	77.1	94.7
	ROADMAP	<u>42.9</u>	37.0	75.0	94.4
Fine	CSL [81]	37.5	36.2	**85.4**	**95.7**
	HAPPIER	41.6	**38.8**	<u>83.8</u>	**95.7**
	HAPPIER$_\mathbf{F}$	**43.7**	<u>38.3</u>	77.5	94.8

outperforms ROADMAP by +2.8pt $\mathcal{H}$-AP and +8.8pt ASI. It also outperforms CSL by +2.6pt $\mathcal{H}$-AP.

4.4.2.4 HAPPIER on trademark logos.

In this section, we report results of HAPPIER on trademark logo retrieval. Both TM Logo-1760 and TM Logo-1399 are Coexya's private datasets compose of $700k$ images and 1760 classes and $1.4m$ images and 1399 classes respectively. We can see that HAPPIER outperforms the standard image classification methods previously used at Coexya on both the fine-grained AP and on $\mathcal{H}$-AP. This again shows the interest of optimizing a well-designed surrogate loss. It also showcases the expressivity of the relevance used for HAPPIER, *i.e.* it can be defined to accommodate for a hierarchical multi-label setting.

Table 4.6: Comparison of ROADMAP and HAPPIER *vs.* standard classification on private trademark logo retrieval datasets.

Method	TM Logo-1760		TM Logo-1399	
	AP	$\mathcal{H}$-AP	AP	$\mathcal{H}$-AP
NSM [66]	15.6	14.1	18.8	19.1
HAPPIER	**37.0**	**37.1**	**41.9**	**38.1**

4.4.3 HAPPIER analysis.

Ablation study. In Tab. 4.7, we study the impact of our different choices regarding the optimization of $\mathcal{H}$-AP. The baseline method uses a sigmoid to optimize $\mathcal{H}$-AP, as in Smooth-AP [69]. Switching to our surrogate loss $\mathcal{L}^s_{\mathcal{H}\text{-AP}}$ yields a +0.8pt increase in $\mathcal{H}$-AP. Finally, the combination with $\mathcal{L}^*_{\text{DG}}$ in HAPPIER results in an additional +1.3pt improvement in $\mathcal{H}$-AP. This again shows the interest of using the proposed ROADMAP framework from Chapter 3.

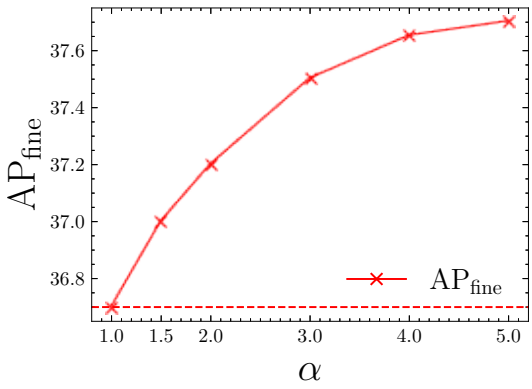
Impact of the relevance function. Tab. 4.8 compares models that are trained with the relevance function of Eq. (4.7), *i.e.* $\mathcal{H}$-AP, and $\sum wAP$ Property 1. We report results for $\mathcal{H}$-AP, $\sum wAP$ and NDCG. Both $\mathcal{H}$-AP, $\sum wAP$ perform better when trained with their own metric: +1.1pt $\mathcal{H}$-AP for the model trained to optimize it and +0.7pt $\sum wAP$ for the model trained to optimize it. Both models show similar performances in NDCG (96.4 *vs.* 97.0). This shows how the relevance choice in $\mathcal{H}$-AP will impact the performances of model.

Table 4.7: Impact of optimization choices for $\mathcal{H}$-AP (cf. Sec. 4.2.3) on iNat-base.

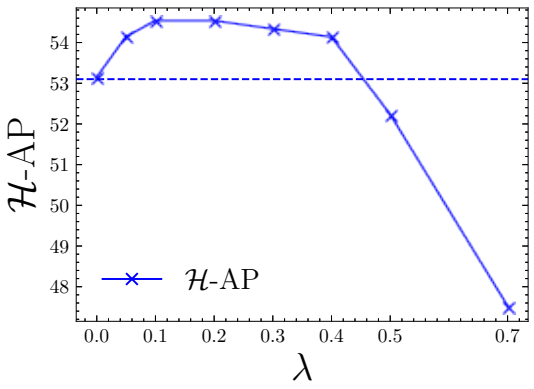
$\mathcal{L}^s_{\mathcal{H}\text{-AP}}$	$\mathcal{L}^*_{\text{DG}}$	$\mathcal{H}$-AP
✗	✗	52.3
✓	✗	53.1
✓	✓	**54.3**

Table 4.8: Comparison of $\mathcal{H}$-AP (Eq. (4.7)) and ΣwAP from Property 1.

test→ train↓	$\mathcal{H}$-AP	$\sum wAP$	NDCG
$\mathcal{H}$-AP	**53.1**	39.8	**97.0**
$\sum wAP$	52.0	**40.5**	96.4



(a) AP_{fine} vs α in Eq. (4.7).



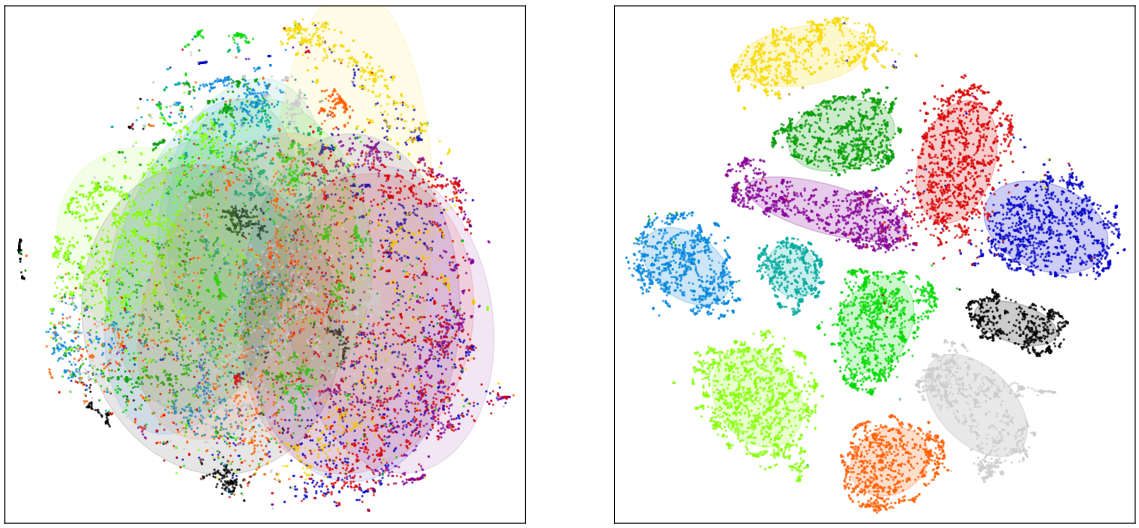
(b) $\mathcal{H}$-AP *vs.* λ for $\mathcal{L}_{\text{HAPPIER}}$.

Figure 4.11: Impact on iNat-base of α in Eq. (4.7) for setting the relevance of $\mathcal{H}$-AP (a) and of the λ hyperparameter on HAPPIER results (b).

Hyperparameters. Fig. 4.11a studies the impact of α for setting the relevance in Eq. (4.7): increasing α improves the performances of the AP at the fine-grained level on iNat-base, as expected. We also show in Fig. 4.11b the impact of λ weighting $\mathcal{L}^s_{\mathcal{H}\text{-AP}}$ and $\mathcal{L}^*_{\text{DG}}$ in HAPPIER performances: we observe a stable increase in $\mathcal{H}$-AP within $0 < \lambda < 0.5$ compared to optimizing only $\mathcal{L}^s_{\mathcal{H}\text{-AP}}$, while a drop in performance is observed for $\lambda > 0.5$. This shows the complementarity of $\mathcal{L}^s_{\mathcal{H}\text{-AP}}$ and $\mathcal{L}^*_{\text{DG}}$, and how, when combined, HAPPIER reaches its best performance.

4.4.4 Qualitative study.

We provide here qualitative assessments of HAPPIER, including embedding space analysis and visualization of HAPPIER's retrievals.

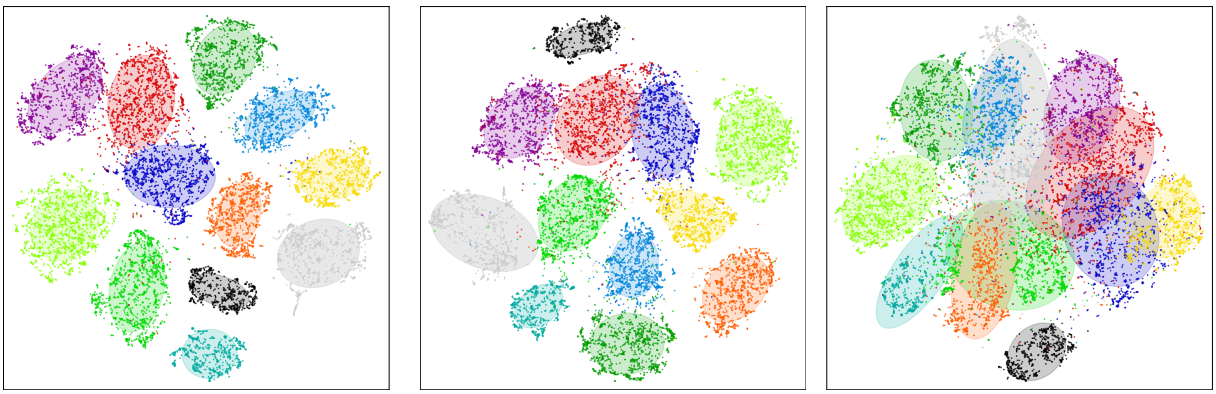


(a) t-SNE visualization of a model trained only on the fine-grained labels.

(b) t-SNE visualization of a model trained with **HAPPIER**.

Figure 4.12: t-SNE visualization of the embedding space of two models trained on SOP. Each point is the average embedding of each fine-grained label (object instance) and the colors represent coarse labels (object category, *e.g.* bike, coffee maker).

t-SNE: organization of the embedding space. In Fig. 4.12, we plot using t-SNE [269], [270] how HAPPIER learns an embedding space on SOP ($L = 2$) that is well-organized. We plot the mean vector of each fine-grained class, and we assign the color based on the coarse level. We show on Fig. 4.12a the t-SNE visualization obtained using a baseline method trained on the fine-grained labels, and in Fig. 4.12b we plot the t-SNE of the embedding space of a model trained with HAPPIER. We cannot observe any clear clusters for the coarse level on Fig. 4.12a,



(a) $\lambda = 0.3$ (b) $\lambda = 0.5$ (c) $\lambda = 0.9$

Figure 4.13: t-SNE visualization of the embedding space of models trained with HAPPIER on SOP with different values of λ. Each point is the average embedding of each fine-grained label (object instance) and the colors represent coarse labels (object category, *e.g.* bike, coffee maker).

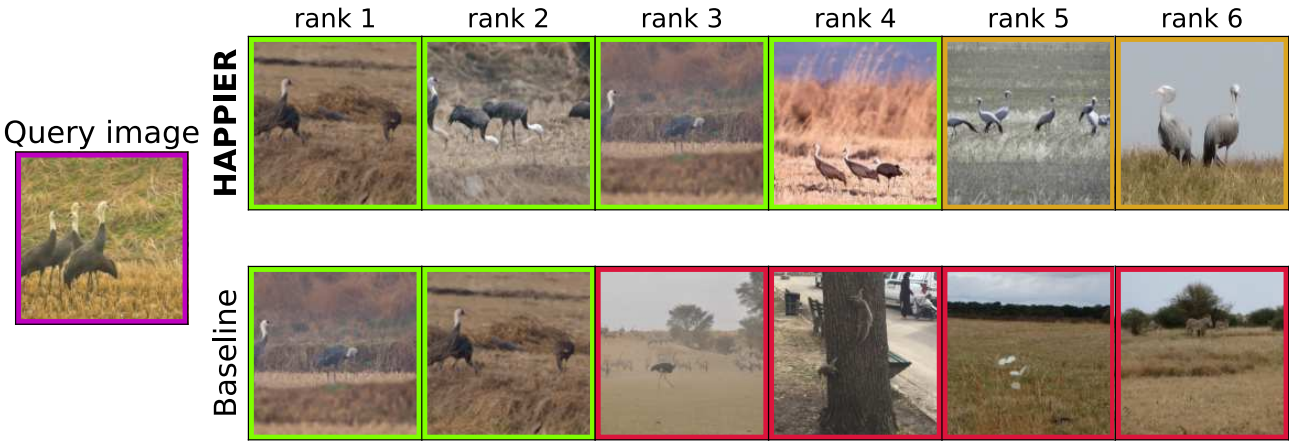
whereas we can appreciate the quality of the hierarchical clusters formed on Fig. 4.12b.

Robustness to λ. Fig. 4.11b illustrates that HAPPIER is robust with respect to λ, with performances increasing for most values between 0.1 and 0.9. In addition, we also show in Fig. 4.13 that for $0 < \lambda < 0.9$ HAPPIER leads to a better organization of the embedding space than a fine-grained baseline (see Fig. 4.12a). This is expected since the lower λ is, the more emphasis is put on optimizing $\mathcal{H}$-AP, which organizes the embedding space in a hierarchical structure.

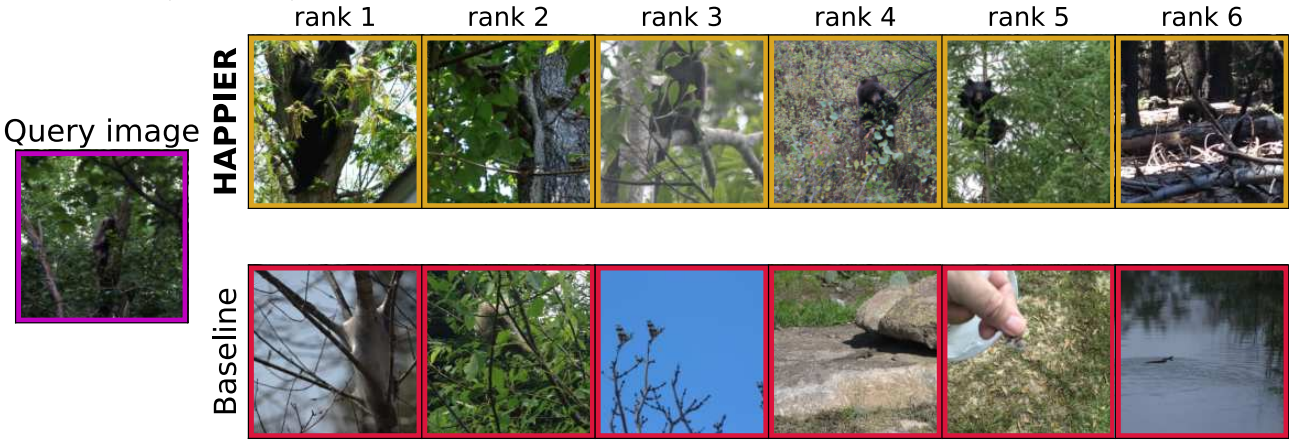
Mistake severity on iNat and SOP. We showcase errors of HAPPIER *vs.* a fine-grained baseline on iNat-base Fig. 4.14 and on SOP Fig. 4.15. On Figs. 4.14a and 4.15a, we illustrate how a model trained with HAPPIER has a lower mistake severity than a baseline model trained only on the fine-grained level. On Figs. 4.14b and 4.15b, we show an example where both models fail to retrieve the correct fine-grained instances, however on Fig. 4.14b the model trained with HAPPIER retrieves images of birds that are semantically more similar to the query, and on Fig. 4.15b the model trained with HAPPIER still retrieve instance of bikes.

Similarly, we illustrate in Figs. 4.16 and 4.17 an example of a query image and the top 25 retrieved results on iNat-full ($L = 7$). Given the same query, both models failed to retrieve the correct fine-grained images (*i.e.* in $\Omega^{(7)}$). The fine-grained model in Fig. 4.17 retrieves images that are semantically more distant than the images retrieved with HAPPIER in Fig. 4.16. For example, HAPPIER retrieves images that are either in $\Omega^{(5)}$ or $\Omega^{(4)}$ (only one instance is in $\Omega^{(3)}$)

whereas the standard model retrieves instances that are in $\Omega^{(2)}$ or $\Omega^{(1)}$.

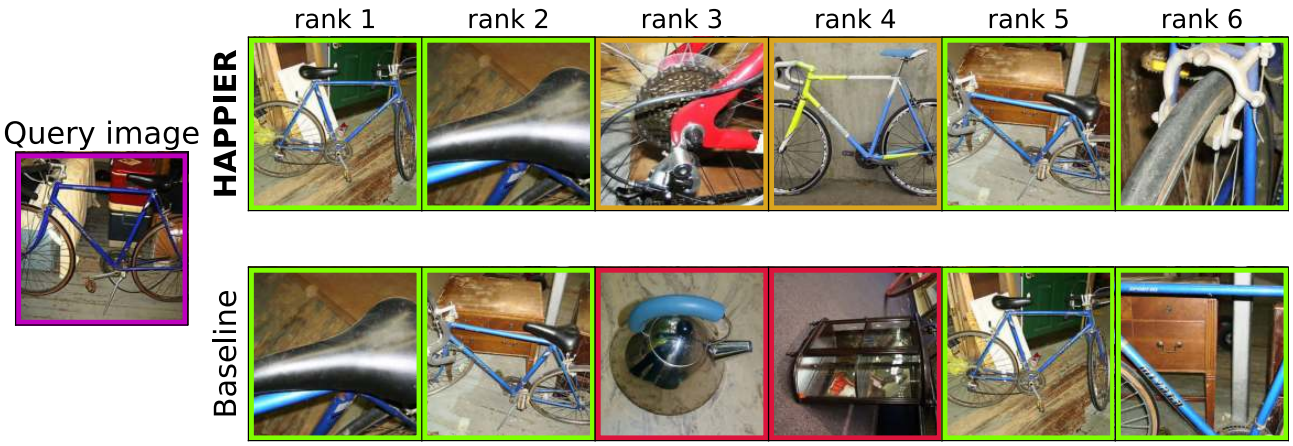


(a) HAPPIER can help make less severe mistakes. The inversion on the bottom row are with negative instances (in red), where as with HAPPIER (top row) inversions are with instances sharing the same coarse label (in orange).



(b) In this example, the models fail to retrieve the correct fine-grained images. However HAPPIER still retrieves images with the same coarse label (in orange) whereas the baseline retrieves images that are dissimilar semantically to the query (in red).

Figure 4.14: Qualitative examples of failure cases from a standard fine-grained model corrected by training with HAPPIER.



(a) HAPPIER can help make less severe mistakes. The inversion on the bottom row are with negative instances (in red), whereas with HAPPIER (top row) inversions are with instances sharing the same coarse label "bike" (in orange).

(b) In this example, the models fail to retrieve the correct fine-grained images. However, HAPPIER still retrieves images of very similar bikes (in orange) whereas the baseline retrieves images that are dissimilar semantically to the query (in red).

Figure 4.15: Qualitative examples of failure cases from a standard fine-grained model corrected by training with HAPPIER.

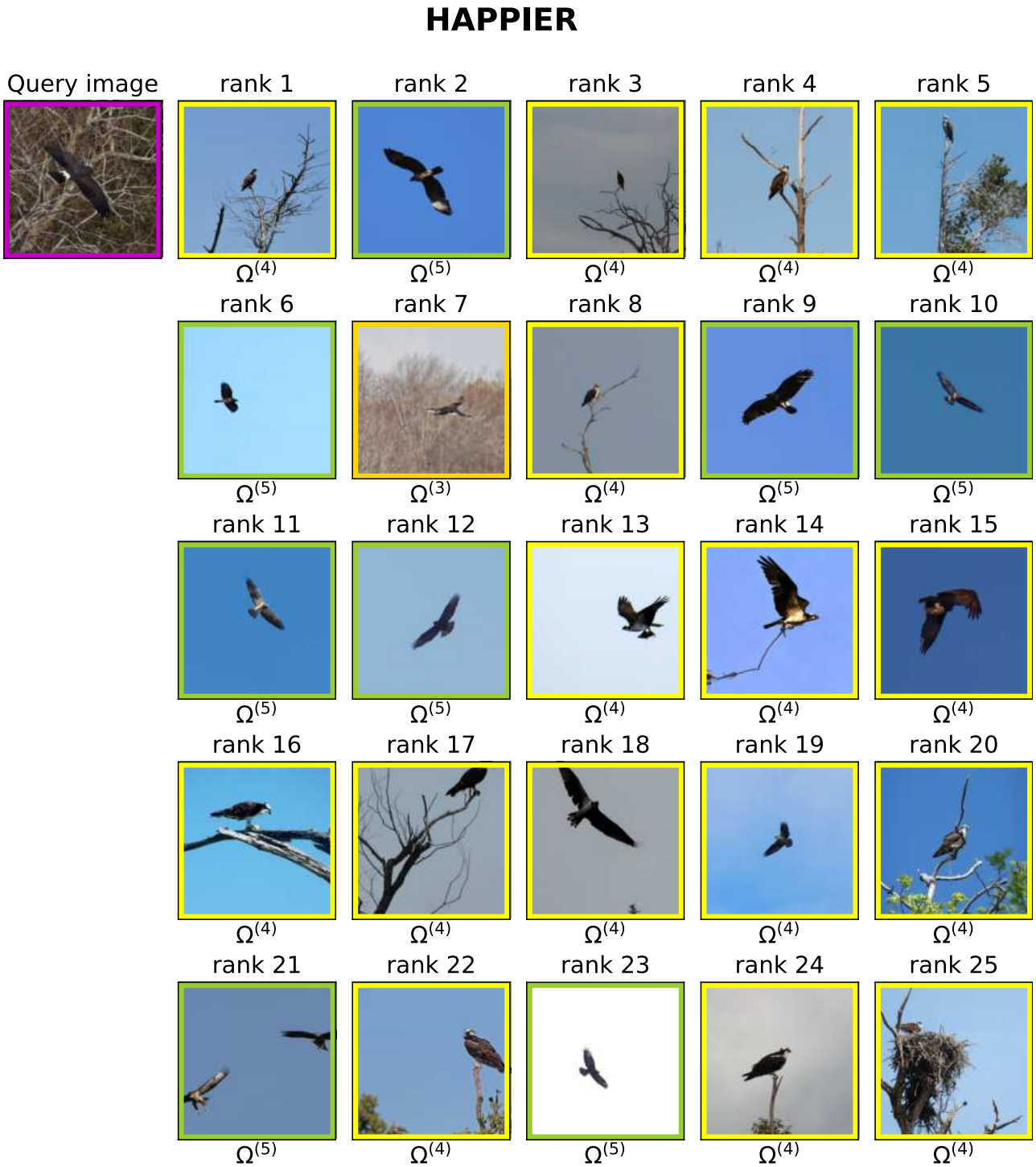


Figure 4.16: Images retrieved for the query image by a model trained with **HAPPIER** on iNat-full ($L = 7$).

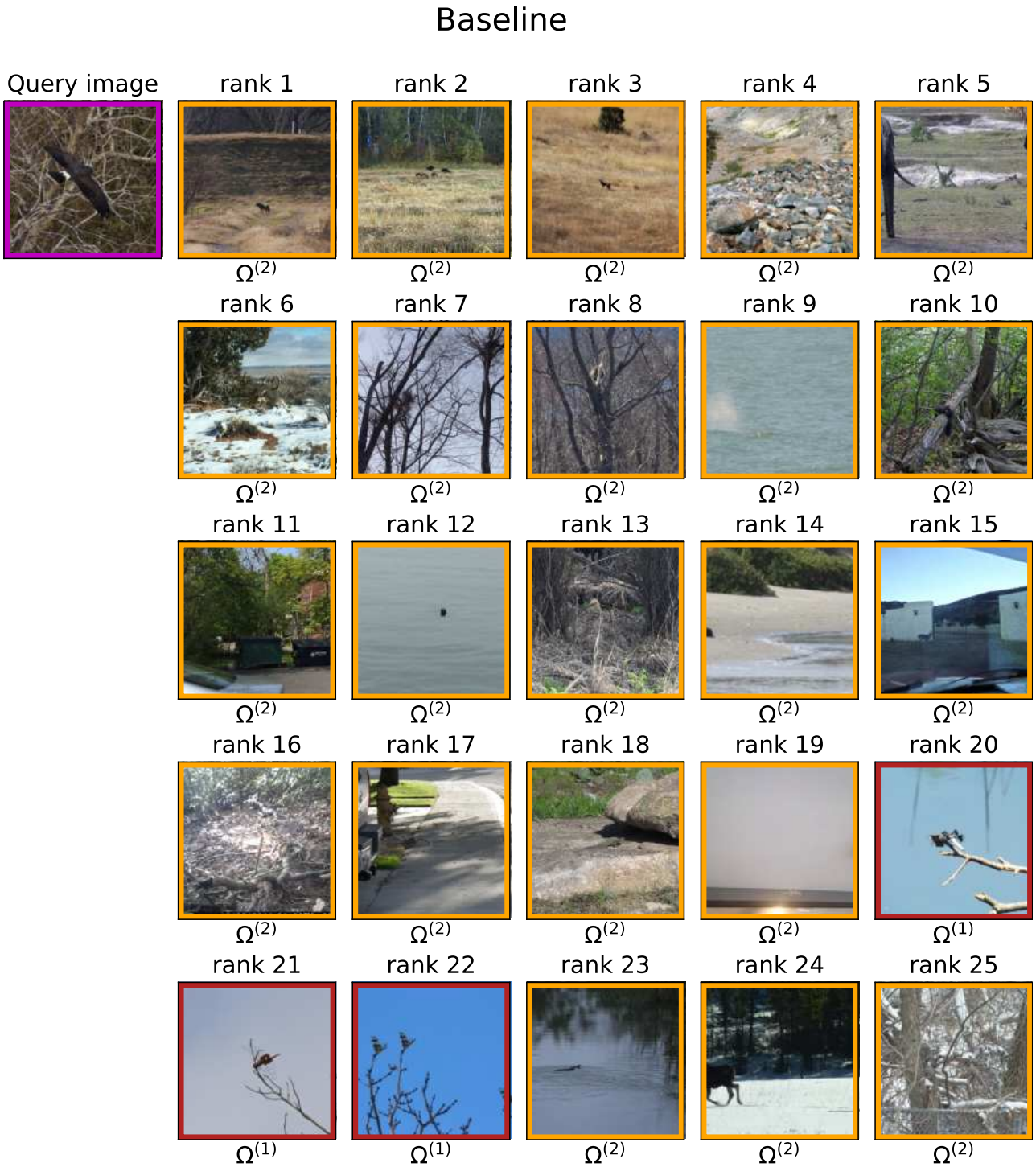


Figure 4.17: Images retrieved for the query image by a model trained with standard model on iNat-full ($L = 7$).

4.5 Conclusion.

In this chapter, we leverage hierarchical relations between concepts to learn robust rankings. We introduced a new metric $\mathcal{H}$-AP that evaluates hierarchical rankings by extending the well-known average precision. We show that using the ROADMAP framework introduced in Chapter 3 we were able to optimize to hierarchical metrics, our novel $\mathcal{H}$-AP and the NDCG. Furthermore, we were able to address one of the shortcomings of this framework: its brittleness on the mistake severity. We showed that creating hierarchical labels on a large scale dataset is feasible, and we released hierarchical annotations for our hierarchical landmark dataset $\mathcal{H}$-GLDv2. Extensive experiments show that HAPPIER and ROD-NDCG performs on par to state-of-the-art image retrieval methods on fine-grained metrics and exhibits significant improvements *vs.* recent hierarchical methods on hierarchical metrics. Learning more robust rankings reduces the severity of ranking errors, and is qualitatively related to a better organization of the embedding space with HAPPIER.

In this chapter, we focused on one aspect of deep neural networks robustness: "mistake severity". We proposed a direction that involves training (or fine-tuning) a model. However, in real world application the model has been trained and will face instances at inference time that are too far from the training distribution and where mistake severity is not sufficient. We will focus on a different aspect of model robustness, out-of-distribution detection.

Chapter 5

Post-hoc out-of-distribution detection

In this chapter, we tackle out-of-distribution (OOD) detection, where a model must decide if inputs come from the same distribution as the training set. OOD detection is a critical requirement for the deployment of deep neural networks. Contrary to the previous chapter where we studied mistake severity with HAPPIER, in this chapter we address the issue of discriminating instances even before classifying them. Furthermore, in this chapter, the robustness is studied once the network is trained, *i.e.* in a *post-hoc* fashion, not enforced during training. Ensuring robustness in a post-hoc fashion allows the use of off-the-shelf models with good predictive performances, which would be prohibitive to train. We introduce in this chapter an **H**ybrid **E**nergy-based model in the fe**AT**ure space, HEAT, which is a new post-hoc OOD detection method. It estimates the density of in-distribution (ID) samples using hybrid energy-based models (EBM) in the feature space of pre-trained backbones. HEAT complements prior OOD detectors, *e.g.* parametric models like Gaussian Mixture Models (GMM), to provide an accurate yet robust density estimation. A second contribution is to leverage the EBM framework to provide a unified density estimation and to compose several energy terms. Extensive experiments demonstrate the significance of the two contributions. We validate HEAT *vs.* state-of-the-art OOD detection methods on the CIFAR-10 / CIFAR-100 benchmarks as well as on the large-scale ImageNet benchmark. The code is available at: `github.com/MarcLafon/heatood`.

Content

5.1 Introduction.

Out-of-distribution (OOD) detection is a major safety requirement for the deployment of deep learning models in critical applications, *e.g.* healthcare, autonomous driving, or defense [271]–[273]. Deployed machine learning systems must successfully perform a specific task, *e.g.* image classification, or object detection while being able to distinguish *in-distribution* (ID) from OOD samples, in order to abstain from making an arbitrary prediction when facing the latter. We also distinguish, near-OOD samples, that have classes more semantically close to the ID samples, from far-OOD with classes that are semantically further.

Post-hoc OOD detection. In recent years, there has been a raise of performant off-the-shelf models available to the deep learning community [6], [12], [34], [274]. To leverage state-of-the-art models for the main prediction task, recent OOD detection approaches follow a *post-hoc* strategy [85], [89], [90], [94], [211], [275], instead of explicitly enforcing OOD detection performances during training, *e.g.* with outlier exposure [209]. This strategy allows maintaining the performances of these off-the-shelf state-of-the-art models, while also relaxing the need for very demanding training processes, which can be prohibitive with huge deep neural networks and foundation models.

Density based OOD detection. State-of-the-art post-hoc methods exploit the feature space of pre-trained networks and attempt to estimate the density of ID features to address OOD detection. Existing ID density estimation methods include Gaussian Mixture Models (GMMs) [89], [91], k nearest neighbors distribution (kNN) [90], or the distribution derived from the energy logits (EL) [94]. However, these approaches tend to detect different types of OOD samples: for instance, GMMs' density explicitly decreases when moving away from training data, making them effective for far-OOD detection, while EL benefits from the classifier training to obtain strong results on near-OOD samples [211]. Because of their strong priors, *e.g.* the feature space is Gaussian for [89], it is difficult to overcome these biases. To address this issue, ResFlow [236] uses a normalizing flow (NF) [234], [235] to learn the residual of a Gaussian density for OOD detection. However, NFs require invertible mapping, which intrinsically limit their expressive power and make the learned residual less accurate, while also being computationally expensive.

Composition for OOD detection. Another direction, discussed in Sec. 2.4.4, is to combine several detectors to perform OOD detection. The topic of model ensembling or composition of scorers has been widely studied in OOD detection. Methods like [238] average the predictions

of several models in order to compute a confidence score. To limit the need for diverse models, methods like [91], [276] compute a confidence score from features at several layers of a DNN. Despite showing high detection rates, these methods induce a large computation overhead at inference time, and may require training several DNN, thus limiting their use in real world applications.

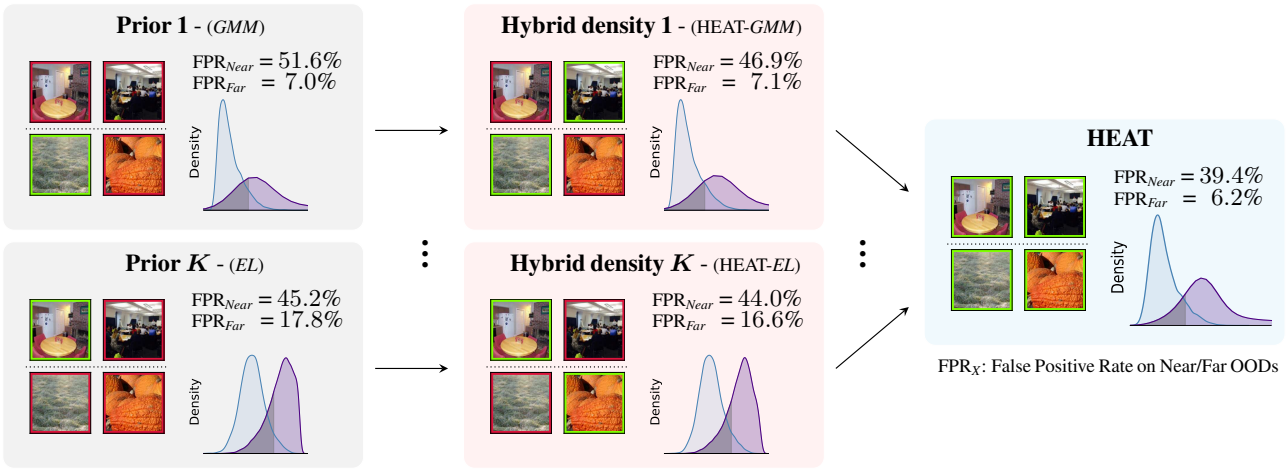


Figure 5.1: **Illustration of our HEAT model**. HEAT leverages a) K prior density estimators, such as GMM or EL, and overcomes their modeling biases by learning a residual term with an EBM b) leading to more accurate OOD scorers, *e.g.* HEAT-GMM or HEAT-EL. The second contribution is to combine the different refined scorers using an EBM energy composition function. The final HEAT prediction c) can thus leverage the strengths of the different OOD scorers, and be effective for both far and near-OOD detection.

In this chapter, we introduce **HEAT**, a new post-hoc density-based OOD detection method which estimates the density of ID samples using a **H**ybrid **E**nergy-based model in the fe**AT**ure space of a fixed pre-trained backbone, which provides strong OOD detection performances on both near and far-OOD data. HEAT leverages the energy-based model (EBM) framework [93] to build a powerful density estimation method relying on two main components:

1. **Energy-based correction** of prior OOD detectors (*e.g.* GMMs or EL) with a data-driven EBM, providing an accurate ID density estimation while benefiting from the strong generalization properties of the priors. The corrected model is carefully trained such that the prior and residual terms achieve optimal cooperation. Furthermore, EBMs do not require a specific architecture design, contrarily to NFs, and can be implemented in practice with a standard multi-layer perceptron.

2. **Hybrid density estimation** by combining several sources to improve OOD detection, using the principled energy functions composition. Which allows leveraging the different modeling biases of prior OOD detectors, *e.g.* GMMs and EL. The energy composition requires only a single hyperparameter. It also involves seamlessly no computational overhead since it is applied at a single layer of the network.

We illustrate HEAT in Fig. 5.1 using two prior OOD detectors from the literature: SSD+ which is based on GMMs [89] and EL [94], with CIFAR-10 dataset as ID dataset and with six OOD datasets, see Sec. 5.3. We can see in Fig. 5.1 that GMM is able to correctly detect far-OOD samples while struggling on near-OOD ones, while EL exhibits the opposite behavior. The energy-correction step enhances both priors, reducing the false positive rate (FPR↓) by -4.7 pts on near-OOD while being stable on far-OOD for GMM, and by -3.2 pts on near-OOD and -1.2 pts for EL. Finally, the energy-composition step produces a hybrid density estimator leading to a better ID density estimation which further improves the OOD detection performances, both for near and far OOD regimes.

We conduct an extensive experimental validation in Sec. 5.3, showing the importance of our two contributions. HEAT performs well *vs.* state-of-the-art OOD methods with CIFAR-10/-100 as ID data, and on the large-scale ImageNet dataset. HEAT is also agnostic to the prediction backbone (ResNet, ViT) and remains effective in low-data regimes.

5.2 HEAT for OOD detection.

In this section, we describe the proposed HEAT model to estimate the density of ID features using a hybrid energy-based model (EBM). We remind that we place ourselves in the difficult but realistic case where only ID samples are available, and we do not use any OOD samples for density estimation. Also, HEAT is a post-hoc approach estimating the density of the latent space of a pre-trained prediction model, as in [89]–[91], [211].

Let $p(\mathbf{x})$ be the probability of ID samples, where $\mathbf{x} \in \mathcal{X}$, and $\mathbf{z} = \phi(\mathbf{x}) \in \mathcal{Z}$ denotes the network's embedding of $\mathbf{x}$ with $\mathcal{Z}$ the d-dimensional latent space at the penultimate layer of a pre-trained prediction model f, *e.g.* a deep neural net for classification. We aim at estimating $p(\mathbf{z}|\mathcal{D})$ with $\mathcal{D} := \{\mathbf{x}_i\}_{i=1}^N$ the ID training dataset[1].

We illustrate the two main components at the core of HEAT in Fig. 5.2. Firstly, we introduce a hybrid density estimation to refine a set of prior densities $\{q_k(z)\}_{1 \leq k \leq K}$ by complementing

[1] we ignore the dependence to $\mathcal{D}$ in the following and denote the sought density as $p(\mathbf{z})$.

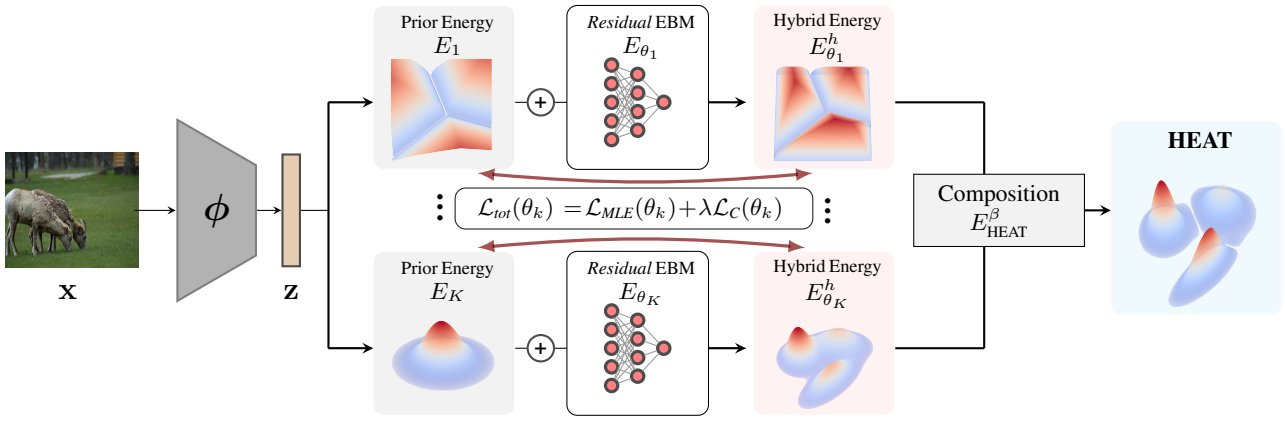


Figure 5.2: **Schematic view of the HEAT model for OOD detection**. Each selected prior density estimator q_k is expressed as an EBM, $q_k(\boldsymbol{z}) \propto \exp\left(-E_{q_k}(\boldsymbol{z})\right)$, and is refined with its own residual EBM parameterized with a neural network: The energy for each prior E_k (*e.g.* EL, GMM) is corrected by a residual energy E_{θ_k} to produce the hybrid energy $E^h_{\theta_k}$ (cf. Sec. 5.2.1). Then all hybrid energies are composed to produce HEAT's energy E^β_{HEAT} (cf. Sec. 5.2.2), which is used as uncertainty score for OOD detection.

each of them with a residual EBM. Secondly, we propose to compose several hybrid density estimations based on different priors, which capture different facets of ID density distributions.

5.2.1 Hybrid Energy-based density estimation.

The main motivation in hybrid EBM density estimation is to leverage existing models that rely on specific assumptions on the form of the density $p(\mathbf{z})$, *e.g.* EL [94], which captures class-specific information in the logit vector, or SSD [89] which uses a GMM. These approaches have appealing properties: GMM is a parametric model relying on few parameters thus exhibiting strong generalization performances, and EL benefits from classification training. However, their underlying modeling assumptions intrinsically limit their expressiveness which leads to coarse boundaries between ID and OOD, and they generally fail at discriminating between ambiguous data.

Hybrid EBM model. Formally, let $q_k(\mathbf{z})$ be a density estimator inducing an OOD-prior among a set of K priors $\{q_k(\mathbf{z})\}_{1 \leq k \leq K}$. We propose to refine its estimated density by learning a residual model $p^r_{\theta_k}(\mathbf{z})$, such that our hybrid density estimation is performed by $p^h_{\theta_k}(\mathbf{z})$ as follows:

$$p^h_{\theta_k}(\mathbf{z}) = \frac{1}{Z(\theta_k)} p^r_{\theta_k}(\mathbf{z}) q_k(\mathbf{z}), \tag{5.1}$$

with $Z(\theta_k) = \int p^r_{\theta_k}(\mathbf{z}) q_k(\mathbf{z}) \mathrm{d}\mathbf{z}$ the normalization constant. We propose to learn the residual density $p^r_{\theta_k}(\mathbf{z})$ with an EBM: $p^r_{\theta_k}(\mathbf{z}) \propto \exp\left(-E_{\theta_k}(\mathbf{z})\right)$. From Eq. (5.1), we can derive a hybrid energy $E^h_{\theta_k}(\mathbf{z}) = E_{q_k}(\mathbf{z}) + E_{\theta_k}(\mathbf{z})$ and express $p^h_{\theta_k}(\mathbf{z})$ as follows:

$$p^h_{\theta_k}(\mathbf{z}) = \frac{1}{Z(\theta_k)} \exp\left(-E^h_{\theta_k}(\mathbf{z})\right), \tag{5.2}$$

with $E_{q_k} = -\log q_k(\mathbf{z})$ the energy from the prior. The goal of the residual energy $E_{\theta_k}(\mathbf{z})$ is to compensate for the lack of accuracy of the energy of the prior density $q_k(\mathbf{z})$. We choose to parameterize it with a neural network, as shown in Fig. 5.2. This gives our EBM density estimation the required expressive power to approximate the residual term.

Hybrid EBM training. The hybrid model energy $E^h_{\theta_k}(\mathbf{z})$ can be learned via maximum likelihood estimation (MLE), which amounts to perform stochastic gradient descent with the following loss:

$$\mathcal{L}_{\text{MLE}}(\theta_k) = \mathbb{E}_{\mathbf{z}\sim p_{in}}\Big[E_{\theta_k}(\mathbf{z})\Big] - \mathbb{E}_{\mathbf{z}'\sim p^h_{\theta_k}}\Big[E_{\theta_k}(\mathbf{z}')\Big], \tag{5.3}$$

with $\mathbf{z} \sim p_{in}$ being the true distribution of the features from the dataset. Minimizing Eq. (5.3) has for effect to lower the energy of real samples while raising the energy of generated ones. To learn a residual model, we *must* sample $\mathbf{z}'$ from the hybrid model $p^h_{\theta_k}$. To do so, we follow previous works on EBM training [218] and exploit stochastic gradient Langevin dynamics (SGLD) [216]. SGLD sampling consists in gradient descent on the energy function:

$$\mathbf{z}_{t+1} = \mathbf{z}_t - \frac{\eta}{2}\nabla_{\mathbf{z}} E^h_{\theta_k}(\mathbf{z}_t) + \sqrt{\eta}\; w_t, \quad \text{with } w_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \tag{5.4}$$

where η is the step size, the chain being initiated with $\mathbf{z}_0 \sim q_k$. The residual energy corrects the prior density by raising (resp. lowering) the energies in areas where the prior over- (resp. under-) estimates p_{in}. It then does so for the current hybrid model $E^h_{\theta_k}$. The overall training hybrid EBM scheme is summarized Algorithm 2.

Controlling the residual. As our goal is to learn a residual model over q, we must prevent the energy-correction term E_{θ_k} to take too large values, thus canceling the benefit from the prior model q_k. Therefore, we introduce an additional loss term, preventing the hybrid model from deviating too much from the prior density:

$$\mathcal{L}_C(\theta_k) = \mathbb{E}_{p_{in}, p^h_{\theta_k}}\Big[\; (E^h_{\theta_k} - E_{q_k})^2 \;\Big]. \tag{5.5}$$

The final loss is then:

$$\mathcal{L}_{Tot}(\theta_k) = \mathcal{L}_{\text{MLE}}(\theta_k) + \lambda\; \mathcal{L}_C(\theta_k), \tag{5.6}$$

where λ is an hyperparameter balancing between the two losses. Although $\mathcal{L}_C(\theta_k)$ in Eq. (5.5) rewrites as $\mathbb{E}_{p_{in},p^h_{\theta_k}}\left[E^2_{\theta_k}\right]$, we point out that its objective goes beyond a standard ℓ_2-regularization used to stabilize training. It has the more fundamental role of balancing the prior and the residual energy terms in order to drive a proper cooperation.

Algorithm 2 Hybrid Energy Based Model Training

input : Features $\mathcal{D}_z$, ID-Prior (q_k, E_{q_k}), λ, α and η.
output: Hybrid EBM $E^h_{\theta_k} = E_{q_k}(\boldsymbol{z}) + E_{\theta_k}(\boldsymbol{z})$. // cf. Eq. (5.2)
while *not converged* **do**
  Sample $\boldsymbol{z} \in \mathcal{D}_z$ and $\boldsymbol{z}'_0 \sim q_k$
  for $0 \leq t \leq T-1$ **do**
    $w \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
    $\boldsymbol{z}'_{t+1} \leftarrow \boldsymbol{z}_{t'} - \frac{\eta}{2}\nabla_z E^h_{\theta_k}(\boldsymbol{z}'_t) + \sqrt{\eta}w$ // SGLD, Eq. (5.4)
  end
  $\mathcal{L}_{Tot}(\theta_k) = \mathcal{L}_{MLE}(\theta_k) + \lambda\mathcal{L}_c(\theta_k)$ // cf. Eq. (5.6)
  $\theta_k \leftarrow \theta_k - \alpha\nabla_{\theta_k}\mathcal{L}_{Tot}(\theta_k)$
end

5.2.2 Composition of refined prior density estimators.

In this section, we motivate the choice of prior OOD scorers that we correct, and how to efficiently compose them within our HEAT framework.

Selected OOD-Priors. As previously stated, EL and GMM show complementary OOD detection performances, EL being useful to discriminate class ambiguities while GMM is effective on far-OOD. Additionally, they can be directly interpreted as energy-based models and thus can easily be refined and composed with HEAT.

1. **Class Prior.** Based on the energy from the logits derived in [94] we express the hybrid energy HEAT-EL as $E^h_{\theta_l}(\mathbf{z}) = -\log\sum_c e^{f(\mathbf{z})[c]} + E^r_{\theta_l}(\mathbf{z})$ where $f(.)[c]$ denotes the logit associated to the class c.

2. **Feature prior.** For the GMM prior, we derive an energy from the Mahalanobis distances to each class centroid. Giving the following expression for our hybrid HEAT-GMM's energy $E^h_{\theta_g}(\mathbf{z}) = -\log\sum_c e^{-\frac{1}{2}(\mathbf{z}-\mu_c)^T\mathbf{\Sigma}^{-1}(\mathbf{z}-\mu_c)} + E^r_{\theta_g}(\mathbf{z})$ with Σ and μ_c being the empirical covariance matrix and mean feature for class c. HEAT-GMM's energy is computed on the

$\mathbf{z}$ vector in Fig. 5.2, which is obtained by average pooling from the preceding tensor in the network.

3. **Style prior.** Finally, inspired by Gram matrices [239], we aim at further exploiting the full feature volume before the pooling operation (*e.g.* average pooling) using the second-order moments. We hypothesize that the volume and its second order moments contains "style" information relevant to OOD detection. We compute the vector of second-order moments of the feature volume by using a std-pooling operator, *i.e.* we compute the standard deviation of each local feature map resulting in a single vector $\sigma \in \mathbb{R}^d$. We subsequently model the density of the second-order features with a GMM. This leads to a third hybrid EBM denoted as HEAT-GMM$_{std}$.

Composition strategy. The EBM framework offers a principled way to make a composition [241] of energy functions. Given K corrected energy functions $E^h_{\theta_k}$, such that: $p^h_{\theta_k} \propto \exp(-(E_{\theta_k}(\mathbf{z}) + E_{q_k}(\mathbf{z}))$, we introduce the following composition function:

$$E^{\beta}_{\text{HEAT}} = \frac{1}{\beta} \log \sum_{k=1}^{K} e^{\beta E^h_{\theta_k}} \tag{5.7}$$

Depending on β, E^{β}_{HEAT} can recover a sum of energies ($\beta = 0$), *i.e.* a product of probabilities. For $\beta = -1$, E^{β}_{HEAT} is equivalent to the *logsumexp* operator, *i.e.* a sum of probabilities. Moreover, unlike previous approaches that require learning a set of weights [91], [236], HEAT's composition only requires tuning a single hyperparameter, *i.e.* β which has a clear interpretation.

The composition strategy adopted in HEAT is also scalable since: i) we work in the feature space $\mathbf{z} = \phi(\mathbf{x}) \in \mathcal{Z}$ of controlled dimension (*e.g.* 1024 even for the CLIP foundation model [12]), and ii) our energy-based correction uses a relatively small model (we use a 6-layers MLP in practice).

OOD detection with HEAT. Finally, we use the learned and composed energy of HEAT, E^{β}_{HEAT} in Eq. (5.7), as an uncertainty score to detect OOD samples, as described in Eq. (2.12).

5.3 Experiments.

Datasets. We validate HEAT on several benchmarks. The two commonly used CIFAR-10 and CIFAR-100 [50] benchmarks, as in [89], [90]. We also conduct experiments on the large-scale ImageNet [43] dataset.

Evaluation metrics. We report the following standard metrics used in the literature [85]: the area under the receiver operating characteristic curve (AUC, higher is better ↑) and the false positive rate at a threshold corresponding to a true positive rate of 95% (FPR95, smaller is better ↓).

$$\text{AUC} = \frac{\sum_{x_{\text{in}} \in \mathcal{D}_{\text{in}}} \sum_{x_{\text{out}} \in \mathcal{D}_{\text{out}}} \mathbb{1}\left[E(x_{\text{in}}) < E(x_{\text{out}})\right]}{|\mathcal{D}_{\text{in}}| \cdot |\mathcal{D}_{\text{out}}|} \tag{5.8}$$

$$\text{FPR95} = \frac{1}{|\mathcal{D}_{\text{out}}|} \cdot \sum_{x_{\text{out}} \in \mathcal{D}_{\text{out}}} \mathbb{1}\left[E(x_{\text{out}}) < \lambda\right] \tag{5.9}$$
$$\text{with } \lambda \in \mathbb{R} \text{ such that for } 95\% \text{ of } x_{\text{in}} \in \mathcal{D}_{\text{in}}, \text{ we have } E(x_{\text{in}}) < \lambda$$

Implementation details. All experiments are conducted using `PyTorch` [277]. We use a ResNet-34 classifier from the `timm` library [249] for the CIFAR-10 and CIFAR-100 datasets and a ResNet-50 for the ImageNet experiments. HEAT consists in a 6 layers MLP trained for 20 epochs with Adam with learning rate $5e$-6. The network input dimension is 512 (which is the dimension of the penultimate layer of ResNet-34) for the CIFAR-10/100 benchmarks and 2048 (which is the dimension of the penultimate layer of ResNet-50) for the ImageNet benchmark. The hidden dimension is 1024 for CIFAR-10/100 and 2048 for ImageNet, and the output dimension is 1. For SGLD sampling, we use 20 steps with an initial step size of $1e$-4 linearly decayed to $1e$-5 and an initial noise scale of $5e$-3 linearly decayed to $5e$-4. We add a small Gaussian noise with std $1e$-4 to each input of the EBM network to stabilize training, as done in previous works [218], [219]. The L_2 coefficient is set to 10. We use temperature scaling on the mixture of Gaussian distributions' energy with temperature $T_{\mathcal{G}} = 1e3$. The hyperparameters for the CIFAR-10 and CIFAR-100 models are identical.

Baselines. We perform extensive validation of HEAT *vs.* several recent state-of-the-art baselines, including the maximum softmax probability (MSP) [85], ODIN [212], Energy-logits [94], SSD [89], KNN [90] and ViM [211]. We apply our energy-based correction of EL, GMM and GMM_{std} that we then denote as HEAT-EL, HEAT-GMM and HEAT-GMM_{std}. We choose those priors as they can naturally be written as energy models as described in Sec. 5.2.1, furthermore, they are strong baselines and combining them allows us to take advantage of their respective strengths (discussed in Sec. 5.2.2). All the baselines are compared using the same backbone trained with the standard cross-entropy loss.

5.3.1 HEAT improvements.

In this section, we study the different components of HEAT. In Tab. 5.1 we show that learning a residual correction term with HEAT improves the OOD detection performances of prior scorers. In Tab. 5.2 we show the interest of learning a residual model as described in Sec. 5.2.1 rather than a standard fully data-driven energy-based model. Finally, in Tab. 5.4 we show how using the energy composition improves OOD detection.

Table 5.1: Refinement of Energy-logits [94] (EL) and GMM, GMM with std-pooling (GMM_{std}) with our energy-based correction on CIFAR-10 and CIFAR-100 as in-distribution datasets. Results are reported with FPR95↓ / AUC ↑.

	Method	*Near-OOD*		*Mid-OOD*		*Far-OOD*		Average
		C-100/10	**TinyIN**	**LSUN**	**Places**	**Textures**	**SVHN**	
CIFAR-10	EL	48.4 / 86.9	41.9 / 88.2	33.7 / 92.6	35.7 / 91.0	30.7 / 92.9	4.9 / 99.0	32.6 / 91.8
	HEAT-EL	**47.3 / 88.0**	**40.7 / 88.9**	**30.8 / 93.4**	**33.8 / 91.8**	**28.8 / 93.9**	**4.5 / 99.1**	**31.0 / 92.5**
	GMM	52.6 / 89.0	50.9 / 89.5	47.1 / 92.4	46.4 / 91.2	**13.1 / 97.8**	**0.9 / 99.8**	35.1 / 93.3
	HEAT-GMM	**49.0 / 89.8**	**44.8 / 90.4**	**40.5 / 93.2**	**40.4 / 92.0**	13.4 / 97.7	**0.8 / 99.8**	**31.5 / 93.8**
	GMM_{std}	58.4 / 84.9	50.6 / 87.9	32.2 / 94.5	38.5 / 91.8	13.8 / **97.6**	**2.5 / 99.5**	32.7 / 92.7
	HEAT-GMM_{std}	**56.1 / 86.1**	**47.8 / 88.7**	**28.2 / 95.2**	**35.8 / 92.5**	**13.3** / **97.5**	2.7 / 99.4	**30.7 / 93.2**
CIFAR-100	EL	80.6 / 76.9	79.4 / 76.5	87.6 / 71.7	83.1 / 74.7	62.4 / 85.2	53.0 / 88.9	74.3 / 79.0
	HEAT-EL	**80.1 / 77.2**	**77.6 / 77.5**	**87.2 / 72.2**	**81.8 / 75.0**	**61.5 / 85.8**	**47.5 / 90.2**	**72.6 / 79.6**
	GMM	85.6 / 73.6	82.5 / 77.2	87.8 / 73.7	84.5 / 74.4	**36.7** / **92.4**	20.0 / 96.3	66.2 / 81.3
	HEAT-GMM	**84.2 / 74.8**	**80.5 / 78.5**	**86.4 / 74.8**	**82.7 / 75.9**	37.9 / **92.2**	**17.8 / 96.7**	**64.9 / 82.1**
	GMM_{std}	91.4 / 67.9	84.3 / 74.8	83.4 / 75.2	83.5 / 75.2	**40.6 / 91.3**	36.7 / 93.1	70.0 / 79.6
	HEAT-GMM_{std}	**89.1 / 70.3**	**82.2 / 76.2**	**82.3 / 76.1**	**81.4 / 76.7**	42.9 / 90.7	**32.9 / 93.8**	**68.5 / 80.6**

Correcting prior scorers. In Tab. 5.1 we demonstrate the effectiveness of energy-based correction to improve different prior OOD scorers on two ID datasets: CIFAR-10 and CIFAR-100. We show that across the two ID datasets and for all prior scorers, using a residual correction always improves the aggregated results, *e.g.* for GMM -3.6 pts FPR95 on CIFAR-10 and -1.3 pts FPR95 on CIFAR-100. Furthermore, on near-OOD and mid-OOD learning our correction always improves the prior scores, *e.g.* on LSUN with CIFAR-10 as ID dataset the correction improves EL by -2.9 pts FPR95, GMM by -6.6 pts FPR95 and -4 pts FPR95 for GMM_{std}. On far-OOD the corrected scorers performs at least on par with the base scorers, and can further improve it, *e.g.* on SVHN when CIFAR-100 is the ID datasets, the correction improves by -5.5pts FPR95,

-2.2 pts FPR95 and -3.8pts FPR95, EL, GMM, GMM_{std} respectively. Overall Tab. 5.1, clearly validates the relevance of correcting the modeling assumptions of prior scorers with our learned energy-based residual.

Learning a residual model. In Tab. 5.2 we compare learning an EBM *vs.* our residual training using a GMM prior (HEAT-GMM) of Sec. 5.2 on CIFAR-10 and CIFAR-100. The EBM is a fully data-driven approach, which learns the density of ID samples without any prior distribution model. On both datasets, our residual training leads to better performances than the EBM, *e.g.* +2.6 pts AUC on CIFAR-100. On near-OOD, both the residual training and the EBM perform on par. On far-OOD, our residual training takes advantage of the good performances of the prior scorer, *i.e.*GMM, and significantly outperforms the EBM, especially on CIFAR-100, with *e.g.* +7.7 pts AUC on Textures. Our residual training combines the strengths of GMM and EBMs: Gaussian modelization by design penalizes samples far away from the training dataset and thus eases far-OOD's detection, whereas EBM may overfit in this case. On the other hand, near-OOD detection requires a too complex density estimation for simple parametric distribution models such as GMMs.

Table 5.2: Comparison of learning a residual model, *i.e.* HEAT-GMM, *vs.* learning an EBM and GMM. Results reported with AUC ↑.

	Method	*Near-OOD*		*Mid-OOD*		*Far-OOD*		**Average**
		C-100/10	**TinyIN**	**LSUN**	**Places**	**Textures**	**SVHN**	
C-10	GMM	89.0	89.5	92.4	91.2	**97.7**	**99.8**	93.3
	EBM	89.4	89.9	**93.8**	91.8	96.2	99.0	93.3
	HEAT-GMM	**89.8**	**90.4**	93.2	**92.0**	**97.7**	**99.8**	**93.8**
C-100	GMM	73.6	77.0	73.8	74.5	**92.4**	96.4	81.3
	EBM	**74.8**	**79.7**	71.9	75.4	84.5	91.0	79.5
	HEAT-GMM	**74.8**	78.5	**74.8**	**75.9**	**92.2**	**96.7**	**82.1**

Comparison to ResFlow. In Tab. 5.3 we compare HEAT to ResFlow [236] using results reported in [89] and use a ResNet-50 trained with the supervised contrastive loss [278] on CIFAR-100. HEAT outperforms ResFlow both on far-OOD detection by +0.8 pts AUC on SVHN and +4.2 pts AUC. This shows the interest of using our HEAT models, which performs better and is more expressive even when having orders of magnitude less layers. Indeed, ResFlow uses for each layer (×4) of the network and for each class (×10) a 30 layers invertible neural network.

Table 5.3: Comparison to ResFlow using a ResNet-50 on CIFAR-10 (ID) using near (CIFAR-100) and far (SVHN) OOD datasets. Results are reported with AUC ↑.

In-distribution (Out-of-distribution)	CIFAR-10 (CIFAR-100)	CIFAR-10 (SVHN)
ResFlow [236]	89.4	99.1
HEAT	**93.6**	**99.9**

Composing energy-based scorers. In Tab. 5.4 we show that composing different energy-based scores (see Sec. 5.2.2), *i.e.* the selected OOD prior scorers with our energy-based correction as described in Sec. 5.2.1, improves overall performances on CIFAR-10 and CIFAR-100. For instance composing our HEAT-GMM and HEAT-GMM$_{std}$ leads to improvements of all reported results, *i.e.* on CIFAR-10 -5.1 pts FPR95 and +0.8 pts AUC and on CIFAR-100 -0.6 pts FPR95 and +0.6 pts AUC. Composing the three prior scorers leads to the best results, improving over the best single scorer performances by great margins on CIFAR-10 with -7.1 pts FPR95 and +1 AUC and with smaller margins on CIFAR-100 -0.8 pts FPR95 and +1.1 pts AUC on CIFAR-100. This shows the interest of composing different scorers as they detect different types of OOD. Note that while the composition has the best performances, our correction model (HEAT-GMM) already has competitive performances on CIFAR-10 and better performances on CIFAR-100 than state-of-the-art methods reported in Tab. 5.5.

Table 5.4: Aggregated performances on CIFAR-10 and CIFAR-100 for the energy composition of the refined OOD scorers of Tab. 5.1.

HEAT -GMM	HEAT -GMM$_{std}$	HEAT -EL	**CIFAR-10** FPR95↓	**CIFAR-10** AUC ↑	**CIFAR-100** FPR95↓	**CIFAR-100** AUC ↑
✓	✗	✗	31.5	93.8	64.9	82.1
✗	✓	✗	30.7	93.2	68.5	80.6
✗	✗	✓	31.0	92.5	72.6	79.6
✓	✓	✗	25.6	94.6	64.3	82.7
✓	✗	✓	28.0	94.1	65.5	82.4
✗	✓	✓	23.6	94.6	66.6	82.1
✓	✓	✓	**23.5**	**94.8**	**63.9**	**83.0**

5.3.2 Comparison to state-of-the-art.

In this section, we present the results of HEAT *vs.* state-of-the-art methods. In Tab. 5.5 we present our results with CIFAR-10, and CIFAR-100 as ID data, and in Tab. 5.6 we present our

results on the large and complex ImageNet dataset.

CIFAR-10 results. In Tab. 5.5 we compare HEAT *vs.* state-of-the-art methods when using CIFAR-10 as the ID dataset. First, we show that HEAT sets a new state-of-the-art on the aggregated results. It outperforms the prior scorers it corrects, *i.e.* SSD+ by -11.6 pts FPR95 and Energy-logits by -9.1 pts FPR95. It also outperforms the previous state-of-the-art methods ViM by -5.3 pts FPR95 and KNN by +1.1 pts AUC. Interestingly, we can see that HEAT outperforms other methods because it improves OOD detection on near-, mid-, and far-OOD. On near OOD, it outperforms KNN by -4.6 pts FPR95 on C-100 and Energy-logits by -6.1 pts FPR95 on TinyIN. On mid-OOD detection, it outperforms ViM by -9.8 pts FPR95 on LSUN and Energy-logits by -8.5 pts FPR95. Finally, on far-OOD, the performances are similar to SSD+ which is by far the best performing method on this regime.

Table 5.5: **Results on CIFAR-10 & CIFAR-100.** All methods are based on a pre-trained ResNet-34 trained on the ID dataset only. ↑ indicates larger is better, and ↓ the opposite. Best results are in bold, second best underlined. Results are reported with FPR95↓ / AUC ↑.

	Method	*Near-OOD*		*Mid-OOD*		*Far-OOD*		Average
		C-10/100	TinyIN	LSUN	Places	Textures	SVHN	
CIFAR-10	MSP [85]	58.0 / 87.9	55.9 / 88.2	50.5 / 91.9	52.7 / 90.2	52.3 / 91.7	19.7 / 97.0	48.2 / 91.2
	ODIN [212]	48.4 / 86.0	42.2 / 87.3	32.6 / 92.3	35.6 / 90.4	29.4 / 92.6	7.8 / 98.3	32.6 / 91.1
	KNN [90]	47.9 / **90.3**	43.1 / 90.6	36.1 / 94.1	37.9 / 92.7	24.9 / 96.0	8.1 / 98.6	33.0 / 93.7
	ViM [211]	44.8 / 89.2	40.1 / 89.8	32.0 / 93.8	34.3 / 92.2	17.9 / 96.4	3.6 / 99.2	28.8 / 93.4
	SSD+ [89]	52.6 / 89.0	50.9 / 89.5	47.1 / 92.4	46.4 / 91.2	13.1 / **97.8**	**0.9 / 99.8**	35.1 / 93.3
	EL [94]	48.4 / 86.9	41.9 / 88.2	33.7 / 92.6	35.7 / 91.0	30.7 / 92.9	4.9 / 99.0	32.6 / 91.8
	DICE [92]	51.0 / 85.7	44.3 / 87.0	33.3 / 92.3	35.6 / 90.5	29.3 / 92.8	3.6 / 99.2	32.8 / 91.3
	HEAT (ours)	**43.1 / 90.2**	**35.7 / 91.3**	**22.2 / 95.8**	**27.4 / 93.9**	**11.3 / 97.9**	**1.1 / 99.8**	**23.5 / 94.8**
CIFAR-100	MSP [85]	**80.0 / 76.6**	78.3 / 77.6	**83.5** / 74.7	81.0 / 76.4	72.1 / 81.0	62.0 / 86.4	76.1 / 78.8
	ODIN [212]	81.4 / 76.4	78.7 / 76.2	86.1 / 72.0	82.6 / 74.5	62.4 / 85.2	80.7 / 80.4	78.6 / 77.5
	KNN [90]	82.1 / 74.5	**76.7 / 80.2**	90.1 / 74.4	83.2 / 75.5	47.2 / 90.2	35.6 / 93.6	69.2 / 81.4
	ViM [211]	85.8 / 74.3	77.5 / 79.6	86.2 / 75.3	**79.8 / 77.6**	42.3 / 91.9	41.3 / 93.2	68.8 / 82.0
	SSD+ [89]	85.6 / 73.6	82.5 / 77.2	87.8 / 73.7	84.5 / 74.4	**36.7** / 92.4	**20.0 / 96.3**	66.2 / 81.3
	EL [94]	**80.6 / 76.9**	79.4 / 76.5	87.6 / 71.7	83.1 / 74.7	62.4 / 85.2	53.0 / 88.9	74.3 / 79.0
	DICE [92]	81.2 / 75.8	82.4 / 74.2	87.8 / 70.4	84.5 / 73.1	63.0 / 83.8	51.9 / 88.1	75.2 / 77.6
	HEAT (ours)	83.7 / 75.8	77.7 / 79.5	**83.4 / 76.3**	**80.0 / 77.8**	37.1 / **92.7**	21.7 / **96.0**	**63.9 / 83.0**

CIFAR-100 results. In Tab. 5.5 we compare HEAT *vs.* state-of-the-art method when using CIFAR-100 as the ID dataset. HEAT outperforms state-of-the-art methods on aggregated results, with -2.3 pts FPR95 and +1.7 pts AUC *vs.*SSD+. HEAT takes advantage of SSD+ on far-OOD and outperforms other methods (except SSD+) by large margins -13.9 pts FPR95 and +2.4 pts AUC on SVHN *vs.* the best non-parametric data-driven density estimation, *i.e.* KNN. Also, HEAT significantly outperforms SSD+ for near-OOD and mid-OOD, *e.g.*-4.8 pts FPR95 on TinyIN or -4.5 pts FPR95 on Places.

ImageNet results. In Tab. 5.6 we compare HEAT on the recently introduced [90] ImageNet OOD benchmark. HEAT sets a new state-of-the-art on this ImageNet benchmark for the aggregated results, with 34.4 FPR95 and 92.6 AUC which outperforms by -1.5 pts FPR95 and +1.7 pts AUC *vs.* the previous best performing method DICE. Furthermore, HEAT improves the aggregated results because it is a competitive method on each dataset. On far-OOD, *i.e.* Textures, it performs on par with SSD+, *i.e.* 5.7 FPR95, the best performing method on this dataset. On mid-OOD, it is the second-best method on SUN and on Places behind DICE. Finally, on near-OOD it performs on par with DICE. This shows that HEAT can be jointly effective on far-, mid-, and near-OOD detection, whereas state-of-the-art methods are competitive for a specific type of OOD only. For instance, the performance of DICE drops significantly on Textures. This also shows that HEAT performs well on larger scale and more complex datasets such as ImageNet. Finally, in Appendix C.2.1 we show that HEAT is also state-of-the-art when using another type of neural network, *i.e.* Vision Transformer [46].

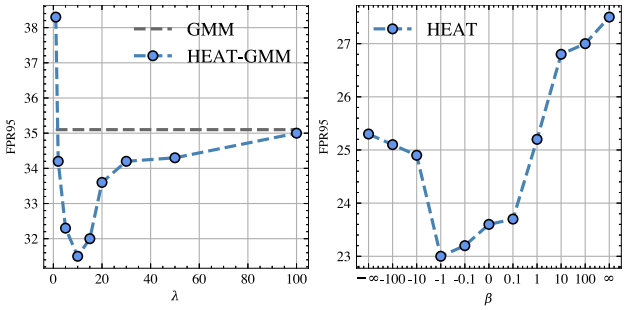
Table 5.6: **Results on ImageNet.** All methods use an ImageNet pre-trained ResNet-50. Results are reported with FPR95↓ / AUC ↑.

Method	iNaturalist	SUN	Places	Textures	Average
MSP [85]	52.8 / 88.4	69.1 / 81.6	72.1 / 80.5	66.2 / 80.4	65.1 / 82.7
ODIN [212]	41.1 / 92.3	56.4 / 86.8	64.2 / 84.0	46.5 / 87.9	52.1 / 87.8
ViM [211]	47.4 / 92.3	62.3 / 86.4	68.6 / 83.3	15.2 / 96.3	48.4 / 89.6
KNN [90]	60.0 / 86.2	70.3 / 80.5	78.6 / 74.8	11.1 / 97.4	55.0 / 84.7
SSD+ [89]	50.0 / 90.7	66.5 / 83.9	76.5 / 78.7	**5.8 / 98.8**	49.7 / 88.0
EL [94]	53.7 / 90.6	58.8 / 86.6	66.0 / 84.0	52.4 / 86.7	57.7 / 87.0
DICE [92]	**26.6** / 94.5	**36.5 / 90.8**	**47.9 / 87.5**	32.6 / 90.4	35.9 / 90.9
HEAT (ours)	28.1 / **94.9**	44.6 / **90.7**	58.8 / 86.3	**5.9** / **98.7**	**34.4** / **92.6**

5.3.3 Model analysis.

In this section, we show how HEAT works in a wide range of settings. We show in Fig. 5.3 the impact of λ and β and in Fig. 5.4 that HEAT performs well in low data regimes.

Robustness to λ. We show in Fig. 5.3a the impact of λ on the FPR95 for CIFAR-10 as the ID dataset. We can observe that for a wide range of λ, *e.g.* $[2, 50]$, our energy-based correction improves the OOD detection of the prior scorer, *i.e.* GMM, with ideal values close to ~ 10. λ controls the cooperation between the prior scorer and the learned residual term, which can be observed on Fig. 5.3a. When setting λ to a value that is too low, there is no control over the energy. The prior density is completely disregarded, which will eventually lead to optimization issues resulting in poor detection performances. On the other hand, setting λ to a value too high (*e.g.* 100) will constrain the energy too much, resulting in performances closer to that of GMM. On CIFAR-10 as the ID dataset, we observe similar trends in Appendix C.2.2.



(a) λ *vs.*FPR95↓ (b) β *vs.*FPR95↓

Figure 5.3: On CIFAR-10 ID: (a) impact of λ in Eq. (5.6) *vs.* FPR95 and (b) analysis of β in Eq. (5.7) *vs.* FPR95.

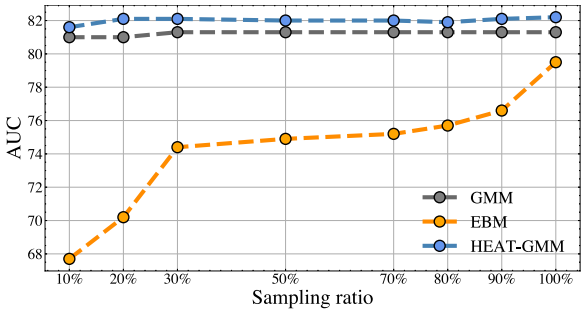


Figure 5.4: Impact on performances (AUC↑ on CIFAR-100) *vs.* the number of training data for GMM density, fully data-driven EBM, and HEAT. Our hybrid approach maintains strong performances in low-data regimes, in contrast to the fully data-driven EBM.

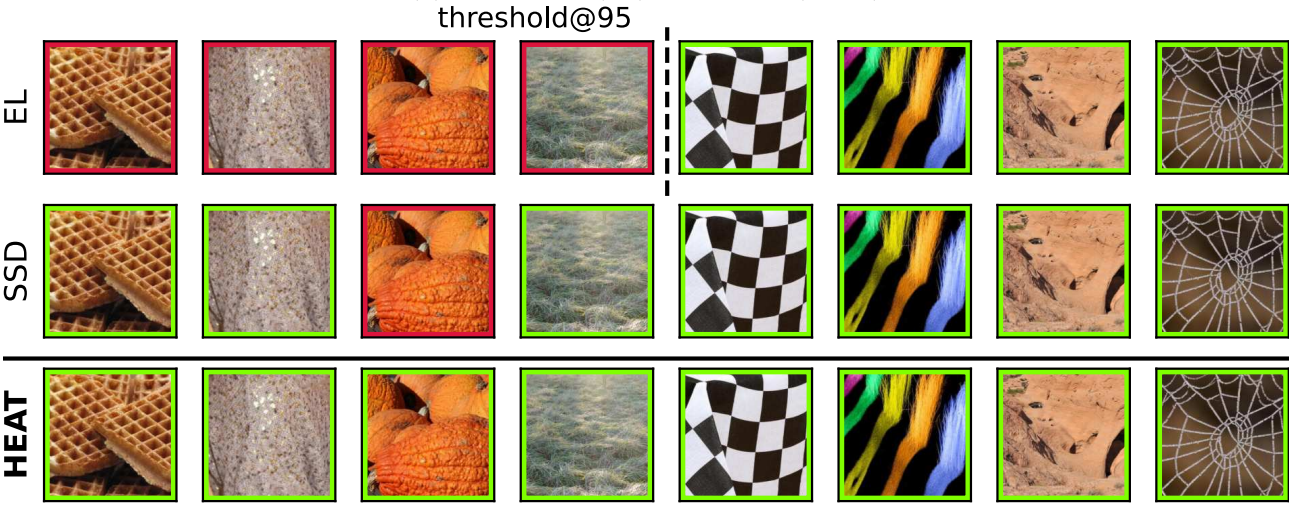
Robustness to β. We show in Fig. 5.3b that HEAT is robust wrt. β in Eq. (5.7). We remind that $\beta \to 0$ is equivalent to the mean, $\beta \to -\infty$ is equivalent to the minimum and $\beta \to \infty$ is equivalent to the maximum. We show that HEAT is stable to different values of β, and performs best with values close to 0. Note that we used $\beta = 0$ for HEAT in Tab. 5.6 and Tab. 5.5 but using a lower value, *i.e.*-1, leads to better results. We hypothesize that using a more advanced $\boldsymbol{\beta}$ selection methods could further improve performances.

Low data regime. We study in Fig. 5.4 the stability of HEAT on low data regimes. Specifically, we restrict the training of HEAT to a subset of the ID dataset, *i.e.* CIFAR-100. We compare HEAT to a fully data-driven EBM and to a GMM. The EBM is very sensitive to the lack of training data, with a gap of 12 pts AUC between 10% of data and 100%. On the other hand, GMM is quite robust to low data regimes, with a minor gap of 0.3 pts AUC between 10% and 100%. HEAT builds on this stability and is able to improve the performance of GMM for all tested sampling ratios. HEAT is very stable to low data regimes which makes it easier to use than a standard EBM, it is also able to improve GMM even when few training data are available.

5.3.4 Qualitative results.

We show qualitative results of HEAT *vs.* EL [94] and SSD [89] for CIFAR-10 (ID) on Fig. 5.5 with LSUN as OOD dataset on Fig. 5.5a and Textures as OOD dataset on Fig. 5.5b. We display in red OOD samples incorrectly identified as ID samples, *i.e.* below the threshold at 95% of ID samples, and in green OOD samples correctly detected, *i.e.* are above the 95% threshold. On Fig. 5.5a, we can see that EL and SSD detect different OOD samples. HEAT is able through the correction and composition to recover those mis-detected OOD samples. On Fig. 5.5b we can see that SSD performs well on Textures, a far-OOD dataset, however HEAT is able to recover a mis-detected OOD sample. Fig. 5.5a and Fig. 5.5b qualitatively show how HEAT is able to better mis-detect OOD samples.

(a) CIFAR-10 (ID) *vs.* LSUN (OOD)



(b) CIFAR-10 (ID) *vs.* Textures (OOD).

Figure 5.5: Qualitative comparison of HEAT *vs.* EL [94] and SSD [89] with CIFAR-10 (ID) *vs.* LSUN (OOD) Fig. 5.5a and Textures (OOD) Fig. 5.5b. Samples in green are correctly detected as OOD (above the 95% of ID threshold), samples in red are incorrectly predicted as ID, *i.e.* an energy lower than the threshold.

5.4 Conclusion.

We have introduced HEAT which leverages the versatility of the EBM framework to provide a strong post-hoc OOD detection method effective on both far and near-OODs. HEAT i) corrects prior OOD detectors to boost their detection performances and ii) naturally combines the corrected detectors to take advantage of their strengths. We perform extensive experiments

to validate HEAT on several benchmarks, highlighting the importance of the correction and the composition, and showing that HEAT sets new state-of-the-art performances on CIFAR-10, CIFAR-100, and on the large-scale ImageNet dataset. HEAT is also applicable to different backbones, and remains efficient in low-data regimes. HEAT can also be extended to K prior scorers, provided that they can write as an EBM and that they are differentiable in order to perform SGLD sampling. Interesting extensions would include adapting the approach to other state-of-the-art OOD detectors, such as a soft-KNN [90] or ViM [211]

HEAT tackles another aspect of model robustness that is involved post-training, which was not the case in Chapter 4. It can also be used to detect images that should not be classified at all, again this was not the case in Chapter 4

Chapter 6

Conclusion and perspectives

We first summarize the contributions that we proposed in this thesis. We then discuss perspective for future works, starting with ongoing works, including adaptation of HAPPIER and ROADMAP to the hierarchical and multi-label setting and local prompt learning for OOD detection. We also talk about long term perspectives, with adaptation of foundation models to image retrieval and image retrieval with human preferences.

Content

6.1 Contributions.

In this thesis, we improved robustness in deep learning from three perspectives: in optimization, in mistake severity and at inference time. In Chapters 3 and 4 we studied robustness during training. More precisely, in Chapter 3 we ensured that the training objectives are aligned with the evaluation metrics, in Chapter 4 we trained models using hierarchical semantic relations that significantly reduced the severity of their mistakes. Finally, in Chapter 5 we investigated post-hoc OOD detection, by using an ensemble of OOD detectors to remove outliers and avoid the processing of uncertain images.

Optimization of Ranking Losses for Image retrieval In this chapter, we first introduce a framework to address the issues of non-differentiability and non-decomposability of ranking losses. It consists of a smooth and differentiable approximation of the rank that has sound theoretical properties and corrects shortcomings of previous rank approximations. We use an additional training objective that supports the decomposability of the target metric. It does not entail computational overhead and can be trained with mini-batches. This framework is general it can be used to optimize losses for fine-grained image retrieval such as AP or R@k, and also to optimize non-binary metrics for hierarchical image retrieval, *e.g.* $\mathcal{H}$-AP or NDCG. We show that this framework leads to very competitive results on image retrieval benchmarks.

Hierarchical Image Retrieval for Robust Ranking In this chapter, we address the severity of mistakes made by traditional image retrieval models. We leverage the hierarchical semantic relations between labels as a proxy for the severity of mistakes. We then integrate these relations during training to make image retrieval systems more robust. We extend the average precision to the hierarchical settings, $\mathcal{H}$-AP, and use the framework from Chapter 3 to optimize graded ranking losses such as $\mathcal{H}$-AP and the NDCG. The models optimized with hierarchical ranking losses perform on par with state-of-the-art standard image retrieval methods results on fine-grained metrics. We then show quantitatively and qualitatively that these models produce more robust rankings. We also propose a semi-automatic pipeline to annotate a fine-grained dataset with hierarchical labels, which we apply to the GLDv2 landmark retrieval dataset. This results in the first hierarchical landmark retrieval dataset, $\mathcal{H}$-GLDv2.

Post-hoc out-of-distribution detection In this chapter, we address DNN robustness at inference time by introducing HEAT, a new method for out-of-distribution (OOD) detection. Specifically, we look at post-hoc OOD detection. Post-hoc methods allow using off-the-shelf

models with strong predictive performances without the need for cumbersome fine-tuning. We use the energy-based model framework to learn a residual term to correct less-expressive prior OOD scorers of the literature. Corrected OOD scorers better approximate the density of in-distribution features. We then show that combining corrected prior OOD scorers allows leveraging their different modeling biases, thus improving overall OOD detection performances. We conduct extensive experimental validation to both show the interest in the different components of HEAT and that HEAT compares well against state-of-the-art methods.

6.2 Perspectives for futures works.

6.2.1 Ongoing work.

(a) Stock logo*.

CLASSIFICATION	DESCRIPTION	AUXILIARY OF
3.6.3	Paws, feet, pawprints, footprints	3.6.1
26.1.15	Circles or ellipses containing representations of animals, parts of animals' bodies or plants	26.1.1, 26.1.2, 26.1.4, 26.1.5, 26.1.6, 26.1.7, 26.1.8, 26.1.9, 26.1.10, 26.1.11, 26.1.12
29.1.1	Red, pink, orange	-

(b) Annotations of WIPO's AI based Vienna classification assistant.

Figure 6.1: The logo of Fig. 6.1a can be annotated with at least three labels from the Vienna classification, as seen on Fig. 6.1b.

*Image taken from: https://stock.adobe.com/

**WIPO's Vienna classification AI: https://vienna-assistant.branddb.wipo.int/

Adaptation of ROADMAP and HAPPIER to multi-label settings. Coexya's data are both multi-labeled and hierarchical. Trademark logos are annotated using Vienna's classification[1], a hierarchical standardized classification for trademark logos. Each trademark logo can be annotated with several labels, as seen on Fig. 6.1. We successfully adapted ROADMAP to the multi-label by computing an average precision for each label, as can be done in multi-label classification to evaluate models [279]. Computing an AP per label converts the multi-label classification task to multiple binary problems. However, this adaptation does not take into account the relations between labels. For instance, it will penalize in the same manner

[1] https://www.wipo.int/classifications/vienna/en/index.html

retrieving an instance of "3.9: AQUATIC ANIMALS, SCORPIONS" when querying with an image of "1.3: SUN" than retrieving an image of "1.1: STARS, COMETS", while both 1.3 and 1.1 are subcategories of "Category 1: CELESTIAL BODIES, NATURAL PHENOMENA, GEOGRAPHICAL MAPS" and semantically closer. We are working on designing a relevance function for HAPPIER to address both the multi-label setting and the hierarchical relations between labels. Although the first tests are encouraging, designing a relevance function that aligns with user requirements will take a bit more work, notably to adjust the relative weights given to the hierarchical relations. More broadly, working on adapting the relevance function of HAPPIER to diverse settings will allow verifying that our framework for the direct optimization works on settings outside traditional retrieval, and confirm the interest of the $\mathcal{H}$-AP objective on a broad set of tasks.

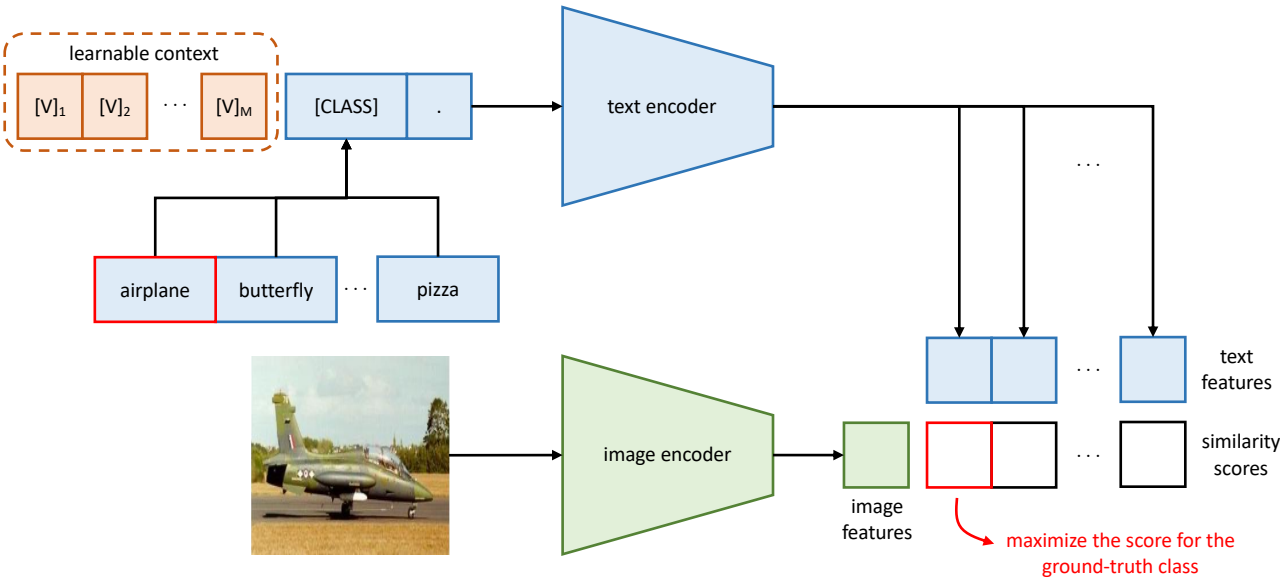


Figure 6.2: Image taken from [280]. The context used to prompt CLIP is learned and allows efficient adaptation to downstream datasets.

Robustness of vision-language models. Vision-language models (VLMs) like CLIP [12] are now broadly used to perform classification on downstream datasets. There is an active research area to study their robustness, *e.g.* their OOD detection capabilities. For instance, in [281] the authors use zero-shot CLIP for OOD detection. They specify the concepts that are considered in-distribution using their names, without CLIP being trained specifically on them. This is a new paradigm for OOD detection, indeed OOD samples are not defined based on the training data, but rather based on the downstream task. Furthermore, methods have emerged to adapt

VLMs efficiently to downstream datasets. For instance, one direction is to learn the prompt used to query CLIP's language model as illustrated on Fig. 6.2, *e.g.* CoOp [280] or MaPLe [282]. The OOD detection capabilities of these methods has been discussed in recent papers, *e.g.* LoCoOp [283] or Catex [284]. Recent methods like [285] leverage both local and global features of CLIP in a zero-shot manner, optimizing this criterion during prompt learning could make text-based classification both more accurate and more robust for OOD detection.

6.2.2 Long-term perspectives.

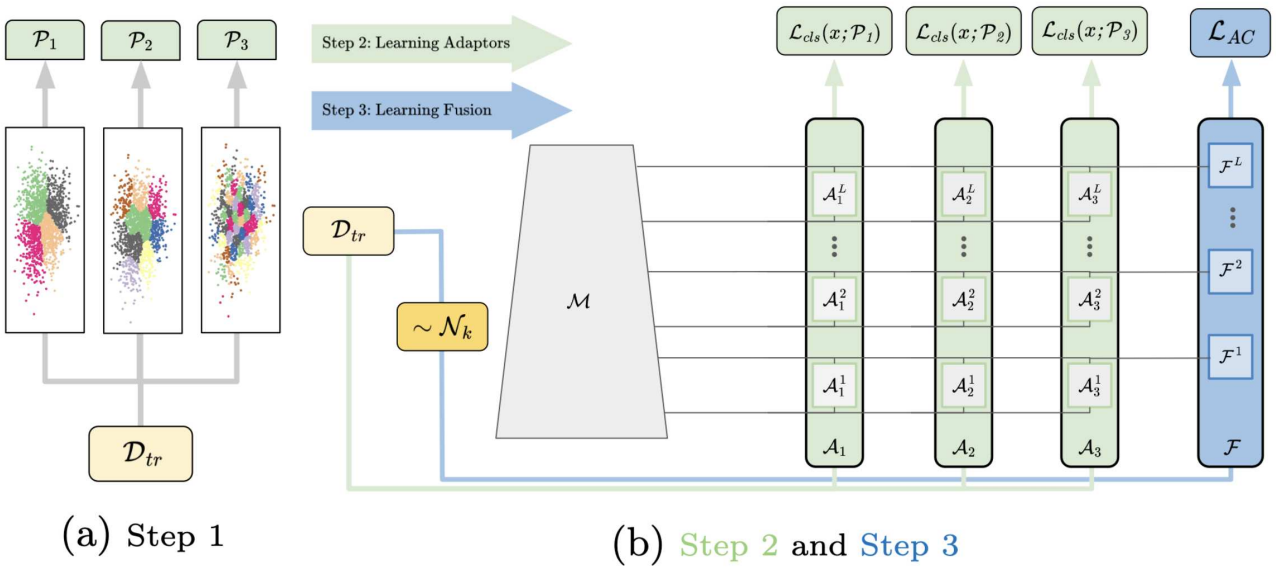


Figure 6.3: Image taken from [150]. The authors of [150] propose to adapt the foundation model Dino [147] to image retrieval. They do so by proposing a multitask framework inspired from [151] to optimize the parameter efficient adapters inspired by the NLP literature [286].

Efficient adaptation for image retrieval. While foundation models exhibit impressive generalization capabilities, they often fall short in terms of performance when compared to expert models that are fine-tuned on specific downstream tasks, such as in image retrieval [36]. Consequently, the adaptation of foundation models to image retrieval has emerged as a significant research challenge. Researchers have explored numerous methods to adapt these models to specialized tasks. These approaches encompass prompt learning [280], [287], where only specific tokens are learned, as well as adapters [150], [288] see Fig. 6.3, which interleave new parameters within the model's existing layers. Robust fine-tuning has also been employed, involving full fine-tuning of foundation models followed by weight averaging [289]. Recent methods have

leveraged the mixture of experts layer [290] principle to interleave a "mixture of adapters" in foundation models. It allows the adaptation to multiple datasets in [291] or to fine-grained classification in [292]. Leveraging this style of adaptation may help adapt foundation models to the diverse domains present in the recent universal image retrieval benchmarks [36], [150].

Image retrieval with human preferences. One desired property of deep learning is to match human expectations. For instance, annotations used to train and evaluate recommender systems in information retrieval can rely on human ratings [293]. With the development of Large-Language-Models (LLM) [2]–[5] embedded in mainstream products, *e.g.* ChatGPT, the domain of natural language processing [48], [294] has been at the forefront of the "alignment" of DNN with human preferences. It relies on a new paradigm that was introduced to fine-tune models with human preferences, namely Reinforcement Learning with Human Feedback (RLHF) [183] that is based on Reinforcement Learning principle, *e.g.* PPO [184] in [183] or Reinforce [295] in [185]. The final pipeline involves multiple steps, starting from pre-training on large unlabeled data, supervised fine-tuning on more curated data, and finally RLHF with policy trained on human-annotated datasets. Similar approaches have been designed in computer vision to optimize a task reward in [185] or in image editing [186]. Image retrieval is a task where human preferences are also the end goal, *e.g.* for Coexya's Acsepto. To optimize human preferences, the first step would be to collect a dataset, *e.g.* by having human annotators rank several retrieval results for a query. Then, because [183] and [185] require stochastic models, we would need to adapt these methods for deterministic image retrieval models. This fine-tuning process would allow image based search engines to more closely match humans' expectations.

Bibliography

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] T. Brown, B. Mann, N. Ryder, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[3] R OpenAI, "Gpt-4 technical report," *arXiv*, pp. 2303–08 774, 2023.

[4] H. Touvron, T. Lavril, G. Izacard, *et al.*, "Llama: open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[5] H. Touvron, L. Martin, K. Stone, *et al.*, "Llama 2: open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[6] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 10 684–10 695.

[7] A. Ramesh, M. Pavlov, G. Goh, *et al.*, "Zero-shot text-to-image generation," in *International Conference on Machine Learning*, PMLR, 2021, pp. 8821–8831.

[8] M. Oquab, T. Darcet, T. Moutakanni, *et al.*, "Dinov2: learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[9] K. He, X. Chen, S. Xie, Y. Li, P. Doll'ar, and R. B. Girshick, "Masked autoencoders are scalable vision learners.," in *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[10] A. Kirillov, E. Mintun, N. Ravi, *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

[11] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International Conference on Machine Learning*, PMLR, 2023, pp. 28 492–28 518.

[12] A. Radford, J. W. Kim, C. Hallacy, *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*, 2021.

[13] R. Girdhar, A. El-Nouby, Z. Liu, *et al.*, "Imagebind: one embedding space to bind them all," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 15 180–15 190.

[14] G. Gemini Team, "Gemini: a family of highly capable multimodal models," *technical report*, 2023.

[15] D. Silver, A. Huang, C. J. Maddison, *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[16] E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Müller, V. Koltun, and D. Scaramuzza, "Champion-level drone racing using deep reinforcement learning," *Nature*, vol. 620, no. 7976, pp. 982–987, 2023.

[17] A. Hu, L. Russell, H. Yeo, *et al.*, "Gaia-1: a generative world model for autonomous driving," *arXiv preprint arXiv:2309.17080*, 2023.

[18] R. Lam, A. Sanchez-Gonzalez, M. Willson, *et al.*, "Graphcast: learning skillful medium-range global weather forecasting," *arXiv preprint arXiv:2212.12794*, 2022.

[19] J. Degrave, F. Felici, J. Buchli, *et al.*, "Magnetic control of tokamak plasmas through deep reinforcement learning," *Nature*, vol. 602, no. 7897, pp. 414–419, 2022.

[20] J. Jumper, R. Evans, A. Pritzel, *et al.*, "Highly accurate protein structure prediction with alphafold," *Nature*, vol. 596, no. 7873, pp. 583–589, 2021.

[21] K. Takahashi and S. Yamanaka, "Induction of pluripotent stem cells from mouse embryonic and adult fibroblast cultures by defined factors," *cell*, vol. 126, no. 4, pp. 663–676, 2006.

[22] Z. Ramzi, G. Chaithya, J.-L. Starck, and P. Ciuciu, "Nc-pdnet: a density-compensated unrolled network for 2d and 3d non-cartesian mri reconstruction," *IEEE Transactions on Medical Imaging*, vol. 41, no. 7, pp. 1625–1638, 2022.

[23] L. Themyr, C. Rambour, N. Thome, T. Collins, and A. Hostettler, "Memory transformers for full context and high-resolution 3d medical segmentation," in *International Workshop on Machine Learning in Medical Imaging*, Springer, 2022, pp. 121–130.

[24] F. Ghasemi, A. Mehridehnavi, A. Pérez-Garrido, and H. Pérez-Sánchez, "Neural network and deep-learning algorithms used in qsar studies: merits and drawbacks," *Drug discovery today*, vol. 23, no. 10, pp. 1784–1790, 2018.

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: a large-scale hierarchical image database," in *CVPR*, 2009.

[26] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, "Microsoft coco: common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, Springer, 2014, pp. 740–755.

[27] B. Zhou, H. Zhao, X. Puig, *et al.*, "Semantic understanding of scenes through the ade20k dataset," *International Journal of Computer Vision*, vol. 127, pp. 302–321, 2019.

[28] T. Weyand, A. Araujo, B. Cao, and J. Sim, "Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval," in *CVPR*, 2020.

[29] G. Van Horn, O. Mac Aodha, Y. Song, *et al.*, "The inaturalist species classification and detection dataset," in *CVPR*, 2018.

[30] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 843–852.

[31] C. Schuhmann, R. Beaumont, R. Vencu, *et al.*, "LAION-5b: an open large-scale dataset for training next generation image-text models," in *NeurIPS Datasets and Benchmarks Track*, 2022.

[32] R. Roy, J. Raiman, N. Kant, *et al.*, "Prefixrl: optimization of parallel prefix circuits using deep reinforcement learning," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*, IEEE, 2021, pp. 853–858.

[33] N. Jouppi, G. Kurian, S. Li, *et al.*, "Tpu v4: an optically reconfigurable supercomputer for machine learning with hardware support for embeddings," in *Proceedings of the 50th Annual International Symposium on Computer Architecture*, 2023, pp. 1–14.

[34] R. Bommasani, D. A. Hudson, E. Adeli, *et al.*, "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021.

[35] G. Ilharco, M. Wortsman, R. Wightman, *et al.*, *Openclip*, 2021.

[36] N.-A. Ypsilantis, K. Chen, B. Cao, *et al.*, "Towards universal image embeddings: a large-scale dataset and challenge for generic image representations," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 11 290–11 301.

[37] J. Zhang, K. Ma, S. Kapse, *et al.*, "Sam-path: a segment anything model for semantic segmentation in digital pathology," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2023, pp. 161–170.

[38] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," *Technologies*, vol. 9, no. 1, p. 2, 2020.

[39] H. Hu, Y. Luan, Y. Chen, *et al.*, "Open-domain visual entity recognition: towards recognizing millions of wikipedia entities," *arXiv preprint arXiv:2302.11154*, 2023.

[40] Y. Lecun, L Eon Bottou, Y. Bengio, and P. H. Abstract|, "Gradient-Based Learning Applied to Document Recognition," in *IEEE*, 1998.

[41] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *2012 IEEE conference on computer vision and pattern recognition*, IEEE, 2012, pp. 3642–3649.

[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *NeurIPS*, 2012. [Online]. Available: http://code.google.com/p/cuda-convnet/.

[43] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: a large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.

[44] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations*, 2015.

[45] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition. corr abs/1512.03385 (2015)*, 2015.

[46] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, "An image is worth 16x16 words: transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[47] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," *arXiv preprint*, 2020.

[48] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[49] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, Ieee, vol. 2, 1999, pp. 1150–1157.

[50] A. Krizhevsky, "Learning multiple layers of features from tiny images," *Master's thesis, Department of Computer Science, University of Toronto*, pp. 32–33, 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

[51] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[52] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[53] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*, Springer, 2020, pp. 213–229.

[54] V. Badrinarayanan, A. Handa, and R. Cipolla, "Segnet: a deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling," *arXiv preprint arXiv:1505.07293*, 2015.

[55] O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, Springer, 2015, pp. 234–241.

[56] T. Mensink, J. Uijlings, L. Castrejon, *et al.*, "Encyclopedic vqa: visual questions about detailed properties of fine-grained categories," *arXiv preprint arXiv:2306.09224*, 2023.

[57] Y. LeCun, C. Cortes, and C. Burges, "Mnist handwritten digit database," *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, vol. 2, 2010.

[58] M. Rzeszotarski, F. Royer, and G. Gilmore, "Introduction to two-dimensional fourier analysis," *Behavior Research Methods & Instrumentation*, vol. 15, pp. 308–318, 1983.

[59] V. N. Mahajan, "Zernike annular polynomials for imaging systems with annular pupils," *JOSA*, vol. 71, no. 1, pp. 75–85, 1981.

[60] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, Ieee, vol. 1, 2005, pp. 886–893.

[61] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: speeded up robust features," in *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*, Springer, 2006, pp. 404–417.

[62] S. A. Chatzichristofis and Y. S. Boutalis, "Fcth: fuzzy color and texture histogram-a low level feature for accurate image retrieval," in *2008 ninth international workshop on image analysis for multimedia interactive services*, IEEE, 2008, pp. 191–196.

[63] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[64] F. Radenović, G. Tolias, and O. Chum, "@articleradenovic2018fine, title=Fine-tuning CNN image retrieval with no human annotation, author=Radenović, Filip and Tolias, Giorgos and Chum, Ondřej, journal=IEEE transactions on pattern analysis and machine intelligence, volume=41, number=7, pages=1655–1668, year=2018, publisher=IEEE with no human annotation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 7, pp. 1655–1668, 2018.

[65] C.-Y. Wu, R Manmatha, A. J. Smola, and P. Krahenbuhl, "Sampling matters in deep embedding learning," in *ICCV*, 2017.

[66] A. Zhai and H. Wu, "Classification is a strong baseline for deep metric learning," *BMVC*, 2018.

[67] J. Revaud, J. Almazán, R. S. Rezende, and C. R. d. Souza, "Learning with average precision: training image retrieval with a listwise loss," in *ICCV*, 2019.

[68] M. Rolínek, V. Musil, A. Paulus, M. Vlastelica, C. Michaelis, and G. Martius, "Optimizing rank-based metrics with blackbox differentiation," in *CVPR*, 2020.

[69] A. Brown, W. Xie, V. Kalogeiton, and A. Zisserman, "Smooth-ap: smoothing the path towards large-scale image retrieval," in *ECCV*, 2020.

[70] Y. Patel, G. Tolias, and J. Matas, "Recall@ k surrogate loss with large batches and similarity mixup," in *CVPR*, 2022.

[71] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.

[72] T. Sorensen, "A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons," *Biologiske skrifter*, vol. 5, pp. 1–34, 1948.

[73] X. Wang, H. Zhang, W. Huang, and M. R. Scott, "Cross-batch memory for embedding learning," in *CVPR*, 2020.

[74] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6572.

[75] L. Bertinetto, R. Mueller, K. Tertikas, S. Samangooei, and N. A. Lord, "Making better mistakes: leveraging class hierarchies with deep networks," in *CVPR*, 2020.

[76] R. Geirhos, J.-H. Jacobsen, C. Michaelis, *et al.*, "Shortcut learning in deep neural networks," *Nature Machine Intelligence*, vol. 2, no. 11, pp. 665–673, 2020.

[77] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *2010 IEEE computer society conference on computer vision and pattern recognition*, IEEE, 2010, pp. 3304–3311.

[78] G. A. Miller, "Wordnet: a lexical database for english," *Commun. ACM*, 1995.

[79] T. Qin, T.-Y. Liu, and H. Li, "A general approximation framework for direct optimization of information retrieval measures," *Information Retrieval*, 2009.

[80] S. E. Robertson, E. Kanoulas, and E. Yilmaz, "Extending average precision to graded relevance judgments," in *SIGIR*, 2010.

[81] Y. Sun, Y. Zhu, Y. Zhang, *et al.*, "Dynamic metric learning: towards a scalable metric space to accommodate multiple semantic scales," in *CVPR*, 2021.

[82] W. Zheng, Y. Huang, B. Zhang, J. Zhou, and J. Lu, "Dynamic metric learning with cross-level concept distillation," in *ECCV*, 2022.

[83] O. Tursun, C. Aker, and S. Kalkan, "A large-scale dataset and benchmark for similar trademark retrieval," *arXiv preprint arXiv:1701.05766*, 2017.

[84] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proceedings of the International Conference on Machine Learning*, 2017.

[85] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *Proceedings of International Conference on Learning Representations*, 2017.

[86] K. Lee, H. Lee, K. Lee, and J. Shin, "Training confidence-calibrated classifiers for detecting out-of-distribution samples," in *ICLR*, 2018. arXiv: 1711.09325v3.

[87] A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks," in *Advances in Neural Information Processing Systems*, 2018.

[88] S. Pidhorskyi, R. Almohsen, and G. Doretto, "Generative probabilistic novelty detection with adversarial autoencoders," *Advances in neural information processing systems*, vol. 31, 2018.

[89] V. Sehwag, M. Chiang, and P. Mittal, "SSD: A unified framework for self-supervised outlier detection," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, OpenReview.net, 2021. [Online]. Available: https://openreview.net/forum?id=v5gjXpmR8J.

[90] Y. Sun, Y. Ming, X. Zhu, and Y. Li, "Out-of-distribution detection with deep nearest neighbors," in *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, and S. Sabato, Eds., ser. Proceedings of Machine Learning Research, vol. 162, PMLR, 2022, pp. 20 827–20 840. [Online]. Available: https://proceedings.mlr.press/v162/sun22d.html.

[91] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in *Advances in Neural Information Processing Systems*, 2018.

[92] Y. Sun and Y. Li, "Dice: leveraging sparsification for out-of-distribution detection," in *European Conference on Computer Vision*, 2022.

[93] Y. LeCun, S. Chopra, R. Hadsell, and F. J. Huang, "A tutorial on energy-based learning," in *Predicting Structured Data*, MIT Press, 2006.

[94] W. Liu, X. Wang, J. D. Owens, and Y. Li, "Energy-based Out-of-distribution Detection," in *Advances in Neural Information Processing Systems*, 2020. arXiv: 2010.03759v4. [Online]. Available: https://github.com/wetliu/energy_ood.

[95] E. Ramzi, N. Thome, C. Rambour, N. Audebert, and X. Bitot, "Robust and decomposable average precision for image retrieval," *NeurIPS*, 2021.

[96] E. Ramzi, N. Audebert, N. Thome, C. Rambour, and X. Bitot, "Hierarchical average precision training for pertinent image retrieval," in *ECCV*, 2022.

[97] M. Lafon, E. Ramzi, C. Rambour, and N. Thome, "Hybrid energy based model in the feature space for out-of-distribution detection," *arXiv preprint arXiv:2305.16966*, 2023.

[98] E. Ramzi, N. Audebert, C. Rambour, A. Araujo, X. Bitot, and N. Thome, "Optimization of rank losses for image retrieval," *arXiv preprint arXiv:2309.08250*, 2023.

[99] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: ideas, influences, and trends of the new age," *ACM Computing Surveys (Csur)*, vol. 40, no. 2, pp. 1–60, 2008.

[100] A. Miech, J.-B. Alayrac, I. Laptev, J. Sivic, and A. Zisserman, "Thinking fast and slow: efficient text-to-visual retrieval with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9826–9836.

[101] C. Jia, Y. Yang, Y. Xia, *et al.*, "Scaling up visual and vision-language representation learning with noisy text supervision," in *International conference on machine learning*, PMLR, 2021, pp. 4904–4916.

[102] G. Delmas, R. S. de Rezende, G. Csurka, and D. Larlus, "Artemis: attention-based retrieval with text-explicit matching and implicit similarity," *arXiv preprint arXiv:2203.08101*, 2022.

[103] A. Baldrati, L. Agnolucci, M. Bertini, and A. Del Bimbo, "Zero-shot composed image retrieval with textual inversion," *arXiv preprint arXiv:2303.15247*, 2023.

[104] H. Liu, Y. Tian, Y. Yang, L. Pang, and T. Huang, "Deep relative distance learning: tell the difference between similar vehicles," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2167–2175.

[105] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, pp. 91–110, 2004.

[106] J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo, "Evaluating bag-of-visual-words representations in scene classification," in *Proceedings of the international workshop on Workshop on multimedia information retrieval*, 2007, pp. 197–206.

[107] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *International journal of computer vision*, vol. 60, pp. 63–86, 2004.

[108] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *2007 IEEE conference on computer vision and pattern recognition*, IEEE, 2007, pp. 1–8.

[109] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[110] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Oakland, CA, USA, vol. 1, 1967, pp. 281–297.

[111] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, Ieee, vol. 2, 2006, pp. 2161–2168.

[112] R Baeza-Yates, "Modern information retrieval," *Addison Wesley google schola*, vol. 2, pp. 127–136, 1999.

[113] G. Tolias, R. Sicre, and H. Jégou, "Particular object retrieval with integral max-pooling of cnn activations," *arXiv preprint arXiv:1511.05879*, 2015.

[114] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII 13*, Springer, 2014, pp. 392–407.

[115] A. Babenko and V. Lempitsky, "Aggregating deep convolutional features for image retrieval," *arXiv preprint arXiv:1510.07493*, 2015.

[116] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, Springer, 2014, pp. 584–599.

[117] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, "End-to-end learning of deep visual representations for image retrieval," *Int. J. Comput. Vis.*, 2017.

[118] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *CVPR*, 2006.

[119] J. L. Schonberger, F. Radenovic, O. Chum, and J.-M. Frahm, "From single image query to detailed 3d reconstruction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5126–5134.

[120] F. Radenovic, J. L. Schonberger, D. Ji, J.-M. Frahm, O. Chum, and J. Matas, "From dusk till dawn: modeling in the dark," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5488–5496.

[121] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, "Deep image retrieval: learning global representations for image search," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14*, Springer, 2016, pp. 241–257.

[122] C. Liao, T. Tsiligkaridis, and B. Kulis, "Supervised metric learning to rank for retrieval via contextual similarity optimization," in *International Conference on Machine Learning*, PMLR, 2023, pp. 20 906–20 938.

[123] Y. Zhu, X. Gao, B. Ke, R. Qiao, and X. Sun, "Coarse-to-fine: learning compact discriminative representation for single-stage image retrieval," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 11 260–11 269.

[124] J. Yan, L. Luo, C. Deng, and H. Huang, "Unsupervised hyperbolic metric learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 465–12 474.

[125] A. Ermolov, L. Mirvakhabova, V. Khrulkov, N. Sebe, and I. Oseledets, "Hyperbolic vision transformers: combining improvements in metric learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7409–7419.

[126] S. Kim, B. Jeong, and S. Kwak, "Hier: metric learning beyond class labels via hierarchical regularization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 19 903–19 912.

[127] V. Khrulkov, L. Mirvakhabova, E. Ustinova, I. Oseledets, and V. Lempitsky, "Hyperbolic image embeddings," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6418–6428.

[128] S. Shao, K. Chen, A. Karpur, Q. Cui, A. Araujo, and B. Cao, "Global features are all you need for image retrieval and reranking," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 11 036–11 046.

[129] J. Revaud, P. Weinzaepfel, C. De Souza, *et al.*, "R2d2: repeatable and reliable detector and descriptor," *arXiv preprint arXiv:1906.06195*, 2019.

[130] P. Weinzaepfel, T. Lucas, D. Larlus, and Y. Kalantidis, "Learning super-features for image retrieval," *arXiv preprint arXiv:2201.13182*, 2022.

[131] C. H. Song, J. Yoon, S. Choi, and Y. Avrithis, "Boosting vision transformers for image retrieval," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 107–117.

[132] R. Arandjelović and A. Zisserman, "Three things everyone should know to improve object retrieval," in *2012 IEEE conference on computer vision and pattern recognition*, IEEE, 2012, pp. 2911–2918.

[133] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman, "Total recall: automatic query expansion with a generative feature model for object retrieval," in *2007 IEEE 11th International Conference on Computer Vision*, IEEE, 2007, pp. 1–8.

[134] S. Kim, D. Kim, M. Cho, and S. Kwak, "Self-taught metric learning without labels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7431–7441.

[135] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Large-scale image retrieval with attentive deep local features," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3456–3465.

[136] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[137] S. Lee, H. Seong, S. Lee, and E. Kim, "Correlation verification for image retrieval," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5374–5384.

[138] F. Tan, J. Yuan, and V. Ordonez, "Instance-level image retrieval using reranking transformers," in *proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 12 105–12 115.

[139] A. Shabanov, A. Tarasov, and S. Nikolenko, "Stir: siamese transformer for image retrieval postprocessing," *arXiv preprint arXiv:2304.13393*, 2023.

[140] P. Turcot and D. G. Lowe, "Better matching with fewer features: the selection of useful features in large database recognition problems," in *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, IEEE, 2009, pp. 2109–2116.

[141] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 1, pp. 117–128, 2010.

[142] H. Liu, R. Wang, S. Shan, and X. Chen, "Deep supervised hashing for fast image retrieval," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2064–2072.

[143] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of ir techniques," *ACM TOIS*, 2002.

[144] W. B. Croft, D. Metzler, and T. Strohman, *Search engines: Information retrieval in practice*. Addison-Wesley Reading, 2010.

[145] J.-B. Grill, F. Strub, F. Altché, *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *Advances in neural information processing systems*, vol. 33, pp. 21 271–21 284, 2020.

[146] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738.

[147] M. Caron, H. Touvron, I. Misra, *et al.*, "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 9650–9660.

[148] S. Kim, D. Kim, M. Cho, and S. Kwak, "Embedding transfer with label relaxation for improved metric learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 3967–3976.

[149] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, PMLR, 2020, pp. 1597–1607.

[150] J. Almazán, B. Ko, G. Gu, D. Larlus, and Y. Kalantidis, "Granularity-aware adaptation for image retrieval over multiple tasks," in *European Conference on Computer Vision*, Springer, 2022, pp. 389–406.

[151] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 132–149.

[152] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," *Advances in neural information processing systems*, vol. 33, pp. 9912–9924, 2020.

[153] H. Touvron, A. Sablayrolles, M. Douze, M. Cord, and H. Jégou, "Grafit: learning fine-grained image representations with coarse labels," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 874–884.

[154] C. Li, X. Ma, B. Jiang, *et al.*, "Deep speaker: an end-to-end neural speaker embedding system," *arXiv preprint arXiv:1705.02304*, 2017.

[155] E. Xing, M. Jordan, S. J. Russell, and A. Ng, "Distance metric learning with application to clustering with side-information," in *NeurIPS*, 2003.

[156] F. Radenovic, G. Tolias, and O. Chum, "CNN image retrieval learns from bow: unsupervised fine-tuning with hard examples," in *ECCV*, 2016.

[157] H. Xuan, A. Stylianou, X. Liu, and R. Pless, "Hard negative examples are hard, but useful," in *ECCV*, 2020.

[158] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *NeurIPS*, 2016.

[159] M. T. Law, N. Thome, and M. Cord, "Learning a distance metric from relative comparisons between quadruplets of images," *Int. J. Comput. Vis.*, 2017.

[160] X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott, "Multi-similarity loss with general pair weighting for deep metric learning," in *CVPR*, 2019.

[161] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh, "No fuss distance metric learning using proxies," in *ICCV*, 2017.

[162] H. Wang, Y. Wang, Z. Zhou, *et al.*, "Cosface: large margin cosine loss for deep face recognition," in *CVPR*, 2018.

[163] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: additive angular margin loss for deep face recognition," in *CVPR*, 2019.

[164] E. W. Teh, T. DeVries, and G. W. Taylor, "Proxynca++: revisiting and revitalizing proxy neighborhood component analysis," in *ECCV*, 2020.

[165] Y. Yue, T. Finley, F. Radlinski, and T. Joachims, "A support vector method for optimizing average precision," in *SIGIR*, 2007.

[166] B. Mcfee and G. Lanckriet, "Metric learning to rank," in *ICML*, 2010.

[167] P. Mohapatra, M. Rolínek, C. Jawahar, V. Kolmogorov, and M. P. Kumar, "Efficient optimization for rank-based loss functions," in *CVPR*, 2018.

[168] T. Durand, N. Thome, and M. Cord, "Exploiting negative evidence for deep latent structured models," *TPAMI*, 2019.

[169] M. Vlastelica, A. Paulus, V. Musil, G. Martius, and M. Rolínek, "Differentiation of blackbox combinatorial solvers," in *ICLR*, 2020.

[170] K. He, F. Cakir, S. A. Bargal, and S. Sclaroff, "Hashing as tie-aware learning to rank," in *CVPR*, 2018.

[171] K. He, Y. Lu, and S. Sclaroff, "Local descriptors optimized for average precision," in *CVPR*, 2018.

[172] E. Ustinova and V. Lempitsky, "Learning deep embeddings with histogram loss," in *NeurIPS*, 2016.

[173] F. Cakir, K. He, X. Xia, B. Kulis, and S. Sclaroff, "Deep metric learning to rank," in *CVPR*, 2019.

[174] M. Engilberge, L. Chevallier, P. Perez, and M. Cord, "Sodeep: a sorting deep net to learn ranking loss surrogates," in *CVPR*, 2019.

[175] J. Geiping, M. Goldblum, P. E. Pope, M. Moeller, and T. Goldstein, "Stochastic training is not necessary for generalization," *arXiv preprint arXiv:2109.14119*, 2021.

[176] B. Harwood, V. Kumar B G, G. Carneiro, I. Reid, and T. Drummond, "Smart mining for deep metric learning," in *ICCV*, 2017.

[177] W. Ge, "Deep metric learning with hierarchical triplet loss," in *ECCV*, 2018.

[178] Y. Suh, B. Han, W. Kim, and K. M. Lee, "Stochastic class-based hard example mining for deep metric learning," in *CVPR*, 2019.

[179] F. Faghri, D. J. Fleet, J. R. Kiros, and S. Fidler, "VSE++: improving visual-semantic embeddings with hard negatives," in *BMVC*, 2018.

[180] M. Carvalho, R. Cadène, D. Picard, L. Soulier, N. Thome, and M. Cord, "Cross-modal retrieval in the cooking context: learning semantic text-image embeddings," in *SIGIR*, 2018.

[181] Y. You, I. Gitman, and B. Ginsburg, "Large batch training of convolutional networks," *arXiv preprint arXiv:1708.03888*, 2017.

[182] J. Davis, T. Liang, J. Enouen, and R. Ilin, "Hierarchical classification with confidence using generalized logits," in *2020 25th International Conference on Pattern Recognition (ICPR)*, IEEE, 2021, pp. 1874–1881.

[183] L. Ouyang, J. Wu, X. Jiang, *et al.*, "Training language models to follow instructions with human feedback," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27 730–27 744, 2022.

[184] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[185] A. S. Pinto, A. Kolesnikov, Y. Shi, L. Beyer, and X. Zhai, "Tuning computer vision models with task rewards," *arXiv preprint arXiv:2302.08242*, 2023.

[186] S. Zhang, X. Yang, Y. Feng, *et al.*, "Hive: harnessing human feedback for instructional visual editing," *arXiv preprint arXiv:2303.09618*, 2023.

[187] J. Goodman, "Classes for fast maximum entropy training," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, IEEE, vol. 1, 2001, pp. 561–564.

[188] F. Morin and Y. Bengio, "Hierarchical probabilistic neural network language model," in *International workshop on artificial intelligence and statistics*, PMLR, 2005, pp. 246–252.

[189] J. Deng, A. C. Berg, K. Li, and L. Fei-Fei, "What does classifying more than 10,000 image categories tell us?" In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part V 11*, Springer, 2010, pp. 71–84.

[190] B. Zhao, F. Li, and E. Xing, "Large-scale category structure aware image categorization," in *Advances in Neural Information Processing Systems*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds., vol. 24, Curran Associates, Inc., 2011. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2011/file/d5cfead94f5350c12c322b5b664544c1-Paper.pdf.

[191] N. Verma, D. Mahajan, S. Sellamanickam, and V. Nair, "Learning hierarchical similarity metrics," in *2012 IEEE conference on computer vision and pattern recognition*, IEEE, 2012, pp. 2280–2287.

[192] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[193] A. Bilal, A. Jourabloo, M. Ye, X. Liu, and L. Ren, "Do convolutional neural networks learn class hierarchy?" *IEEE transactions on visualization and computer graphics*, vol. 24, no. 1, pp. 152–162, 2017.

[194] A. Dhall, A. Makarova, O. Ganea, D. Pavllo, M. Greeff, and A. Krause, "Hierarchical image classification using entailment cone embeddings," in *CVPR Workshops*, 2020.

[195] D. Chang, K. Pang, Y. Zheng, Z. Ma, Y.-Z. Song, and J. Guo, "Your" flamingo" is my" bird": fine-grained, or not," in *CVPR*, 2021.

[196] A. Garg, D. Sani, and S. Anand, "Learning hierarchy aware features for reducing mistake severity," in *European Conference on Computer Vision*, Springer, 2022, pp. 252–267.

[197] B. Hjørland, "The foundation of the concept of relevance," *Journal of the American Society for Information Science and Technology*, 2010.

[198] J. Kekäläinen and K. Järvelin, "Using graded relevance assessments in ir evaluation," *Journal of the American Society for Information Science and Technology*, 2002.

[199] C. Burges, T. Shaked, E. Renshaw, *et al.*, "Learning to rank using gradient descent," in *ICML*, 2005.

[200] C. Burges, R. Ragno, and Q. Le, "Learning to rank with nonsmooth cost functions," in *NeurIPS*, 2006.

[201] M. Taylor, J. Guiver, S. Robertson, and T. Minka, "Softrank: optimizing non-smooth rank metrics," in *WSDM*, 2008.

[202] S. Bruch, M. Zoghi, M. Bendersky, and M. Najork, "Revisiting approximate metric optimization in the age of deep neural networks," in *SIGIR*, 2019.

[203] G. Dupret and B. Piwowarski, "Model based comparison of discounted cumulative gain and average precision," *Journal of Discrete Algorithms*, 2013.

[204] G. Dupret and B. Piwowarski, "A user behavior model for average precision and its generalization to graded judgments," in *SIGIR*, 2010.

[205] S. Kim, M. Seo, I. Laptev, M. Cho, and S. Kwak, "Deep metric learning beyond binary supervision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2288–2297.

[206] Z. Yang, M. Bastan, X. Zhu, D. Gray, and D. Samaras, "Hierarchical proxy-based loss for deep metric learning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 1859–1868.

[207] D. Zhang, Y. Li, and Z. Zhang, "Multi-scale similarity aggregation for dynamic metric learning," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 125–134.

[208] J. Yang, K. Zhou, Y. Li, and Z. Liu, "Generalized out-of-distribution detection: a survey," *arXiv preprint arXiv:2110.11334*, 2021.

[209] D. Hendrycks, M. Mazeika, and T. Dietterich, "Deep Anomaly Detection with Outlier Exposure," in *ICLR*, 2019. arXiv: 1812.04606v3. [Online]. Available: https://github.com/hendrycks/outlier-exposure..

[210] B. Charpentier, D. Zügner, and S. Günnemann, "Posterior network: uncertainty estimation without ood samples via density-based pseudo-counts," in *Advances in Neural Information Processing Systems*, 2020.

[211] H. Wang, Z. Li, L. Feng, and W. Zhang, "Vim: out-of-distribution with virtual-logit matching," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, IEEE, 2022, pp. 4911–4920. DOI: 10.1109/CVPR52688.2022.00487. [Online]. Available: https://doi.org/10.1109/CVPR52688.2022.00487.

[212] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," in *Proceedings of International Conference on Learning Representations*, 2018.

[213] C. S. Sastry and S. Oore, "Detecting out-of-distribution examples with gram matrices," in *International Conference on Machine Learning*, PMLR, 2020, pp. 8491–8501.

[214] J. Li, P. Chen, Z. He, S. Yu, S. Liu, and J. Jia, "Rethinking out-of-distribution (ood) detection: masked image modeling is all you need," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[215] R. Gao, C. Zhao, L. Hong, and Q. Xu, "Diffguard: semantic mismatch-guided out-of-distribution detection using pre-trained diffusion models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 1579–1589.

[216] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient langevin dynamics," in *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, 2011, pp. 681–688. [Online]. Available: https://icml.cc/2011/papers/398_icmlpaper.pdf.

[217] R. M. Neal, "MCMC using Hamiltonian dynamics," *Handbook of Markov Chain Monte Carlo*, vol. 54, pp. 113–162, 2010.

[218] Y. Du and I. Mordatch, "Implicit generation and modeling with energy-based models," in *Advances in Neural Information Processing Systems*, vol. 32, 2019. arXiv: 1903.08689v6. [Online]. Available: https://sites.google.com/view/igebm.

[219] W. Grathwohl, K.-C. Wang, J.-H. Jacobsen, D. Duvenaud, M. Norouzi, and K. Swersky, "Your Classifier is Secretly an Energy Based Model and You Should Treat it Like One," in *8th International Conference on Learning Representations*, 2020. arXiv: 1912.03263. [Online]. Available: http://arxiv.org/abs/1912.03263.

[220] D. J. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *32nd International Conference on Machine Learning, ICML 2015*, vol. 2, 2015, pp. 1530–1538, ISBN: 9781510810587. arXiv: 1505.05770.

[221] J. Xie, Y. Lu, S. Zhu, and Y. N. Wu, "A theory of generative convnet," in *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, M. Balcan and K. Q. Weinberger, Eds., ser. JMLR Workshop and Conference Proceedings, vol. 48, JMLR.org, 2016, pp. 2635–2644. [Online]. Available: http://proceedings.mlr.press/v48/xiec16.html.

[222] S. Elflein, B. Charpentier, D. Zügner, and S. Günnemann, "On Out-of-distribution Detection with Energy-based Models," in *ICML Workshop*, 2021. [Online]. Available: https://github.com/selflein/.

[223] A. Zeng, S. Song, J. Lee, A. Rodriguez, and T. A. Funkhouser, "Tossingbot: learning to throw arbitrary objects with residual physics," *IEEE Transactions on Robotics*, vol. 36, pp. 1307–1319, 2020.

[224] Y. Yin, V. Le Guen, J. Dona, *et al.*, "Augmenting physical models with deep networks for complex dynamics forecasting," in *Ninth International Conference on Learning Representations ICLR 2021*, 2021.

[225] A. Bakhtin, Y. Deng, S. Gross, M. Ott, M. A. Ranzato, and A. Szlam, "Residual Energy-based Models for Text," *Journal of Machine Learning Research*, vol. 22, pp. 1–18, 2021, ISSN: 15337928. arXiv: 2004.10188.

[226] V. Le Guen and N. Thome, "Disentangling physical dynamics from unknown factors for unsupervised video prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[227] V. Le Guen, C. Rambour, and N. Thome, "Complementing brightness constancy with deep networks for optical flow prediction," in *European Conference on Computer Vision (ECCV)*, 2022.

[228] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, Y. W. Teh and D. M. Titterington, Eds., ser. JMLR Proceedings, vol. 9, JMLR.org, 2010, pp. 297–304. [Online]. Available: http://proceedings.mlr.press/v9/gutmann10a.html.

[229] J. Xie, Y. Lu, R. Gao, and Y. N. Wu, "Cooperative learning of energy-based model and latent variable model via MCMC teaching," *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pp. 4292–4301, 2018.

[230] J. Xie, Z. Zheng, X. Fang, S. Zhu, and Y. N. Wu, "Cooperative training of fast thinking initializer and slow thinking solver for conditional learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 8, pp. 3957–3973, 2022. DOI: 10.1109/TPAMI.2021.3069023. [Online]. Available: https://doi.org/10.1109/TPAMI.2021.3069023.

[231] B. Pang, T. Han, E. Nijkamp, S. C. Zhu, and Y. N. Wu, "Learning latent space energy-based prior model," in *Advances in Neural Information Processing Systems*, vol. 2020-Decem, 2020. arXiv: 2006.08205.

[232] J. Xie, Z. Zheng, and P. Li, "Learning energy-based model with variational auto-encoder as amortized sampler," in *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, AAAI Press, 2021, pp. 10 441–10 451. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/17250.

[233] Z. Xiao, K. Kreis, J. Kautz, and A. Vahdat, "VAEBM: A Symbiosis between Variational Autoencoders and Energy-based Models," in *Proceedings of International Conference on Learning Representations*, 2021, ISBN: 2010.00654v2. arXiv: 2010.00654. [Online]. Available: http://arxiv.org/abs/2010.00654.

[234] E. Nijkamp, R. Gao, P. Sountsov, *et al.*, "MCMC should mix: learning energy-based model with neural transport latent space MCMC," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, OpenReview.net, 2022. [Online]. Available: https://openreview.net/forum?id=4C93Qvn-tz.

[235] J. Xie, Y. Zhu, J. Li, and P. Li, "A tale of two flows: cooperative learning of langevin flow and normalizing flow toward energy-based model," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, OpenReview.net, 2022. [Online]. Available: https://openreview.net/forum?id=31d5RLCUuXC.

[236] E. Zisselman and A. Tamar, "Deep Residual Flow for Out-of-Distribution Detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 991–14 000. DOI: 10.1109/CVPR42600.2020.01401. arXiv: 2001.05419.

[237] N. Lang, N. Kalischek, J. Armston, K. Schindler, R. Dubayah, and J. D. Wegner, "Global canopy height regression and uncertainty estimation from gedi lidar waveforms with deep ensembles," *Remote Sensing of Environment*, vol. 268, p. 112 760, 2022.

[238] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems*, 2017.

[239] C. S. Sastry and S. Oore, "Detecting out-of-distribution examples with gram matrices," *37th International Conference on Machine Learning, ICML 2020*, vol. PartF16814, pp. 8449–8459, 2020. [Online]. Available: https://github.com/.

[240] O. Laurent, A. Lafage, E. Tartaglione, *et al.*, "Packed-ensembles for efficient uncertainty estimation," *arXiv preprint arXiv:2210.09184*, 2022.

[241] Y. Du, S. Li, and I. Mordatch, "Compositional Visual Generation with Energy Based Models," *Advances in Neural Information Processing Systems*, vol. 2020-Decem, 2020, ISSN: 10495258. arXiv: 2004.06030v3. [Online]. Available: https://energy-based-model.github.io/.

[242] Y. Du, S. Li, Y. Sharma, J. B. Tenenbaum, and I. Mordatch, "Unsupervised Learning of Compositional Energy Concepts," *Advances in Neural Information Processing Systems*, 2021. arXiv: 2111.03042. [Online]. Available: https://energy-based-model.github.io/comet/http://arxiv.org/abs/2111.03042.

[243] Y. Zhu, M. Yang, C. Deng, and W. Liu, "Fewer is more: a deep graph metric learning perspective using fewer proxies," in *NeurIPS*, 2020.

[244] I. Amir, T. Koren, and R. Livni, "Sgd generalizes better than gd (and regularization doesn't help)," in *Conference on Learning Theory*, PMLR, 2021, pp. 63–92.

[245] Z. Li, W. Min, J. Song, *et al.*, "Rethinking the optimization of average precision: only penalizing negative instances before positive ones is enough," in *AAAI*, 2022.

[246] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *CVPR*, 2016.

[247] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," California Institute of Technology, Tech. Rep., 2011.

[248] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, "An image is worth 16x16 words: transformers for image recognition at scale," *arXiv preprint*, 2020.

[249] R. Wightman, *Pytorch image models*, https://github.com/rwightman/pytorch-image-models, 2019. DOI: 10.5281/zenodo.4414861.

[250] K. Musgrave, S. Belongie, and S.-N. Lim, "A metric learning reality check," in *European Conference on Computer Vision*, Springer, 2020, pp. 681–699.

[251] D. Zhang, Y. Li, and Z. Zhang, "Deep metric learning with spherical embedding," in *NeurIPS*, 2020.

[252] P. Jacob, D. Picard, A. Histace, and E. Klein, "Metric learning with horde: high-order regularizer for deep embeddings," in *ICCV*, 2019.

[253] M. Boudiaf, J. Rony, I. M. Ziko, *et al.*, "A unifying mutual information view of metric learning: cross-entropy vs. pairwise losses," in *ECCV*, 2020.

[254] A. El-Nouby, N. Neverova, I. Laptev, and H. Jégou, "Training vision transformers for image retrieval," *arXiv preprint*, 2021.

[255] F. Radenović, A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, "Revisiting oxford and paris: large-scale image retrieval benchmarking," in *CVPR*, 2018.

[256] T. Qin and T. Liu, "Introducing LETOR 4.0 datasets," *CoRR*, 2013.

[257] O. Chapelle and Y. Chang, "Yahoo! learning to rank challenge overview," in *Proceedings of the learning to rank challenge*, 2011.

[258] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *3dRR Workshop*, 2013.

[259] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "Deepfashion: powering robust clothes recognition and retrieval with rich annotations," in *CVPR*, 2016.

[260] E. W. Wilt and A. V. Harrison, "Creating a semantic hierarchy of SUN database object labels using WordNet," in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications III*, vol. 11746, 2021.

[261] D. Chen, G. Baatz, K. Koser, *et al.*, "City-Scale Landmark Identification on Mobile Devices," in *CVPR*, 2011.

[262] Y. Avrithis, G. Tolias, and Y. Kalantidis, "Feature Map Hashing: Sub-linear Indexing of Appearance and Global Geometry," in *Proc. ACM MM*, 2010.

[263] A. Torii, R. Arandjelović, J. Sivic, M. Okutomi, and T. Pajdla, "24/7 place recognition by view synthesis," in *Proc. CVPR*, 2015.

[264] S. Yokoo, K. Ozaki, E. Simo-Serra, and S. Iizuka, "Two-stage discriminative re-ranking for large-scale landmark retrieval," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 1012–1013.

[265] N. Reimers and I. Gurevych, "Sentence-bert: sentence embeddings using siamese bert-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[266] R. Fagin, R. Kumar, and D. Sivakumar, "Comparing top k lists," *SIAM Journal on discrete mathematics*, 2003.

[267] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[268] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[269] L. van der Maaten and G. Hinton, "Visualizing high-dimensional data using t-sne," *Journal of Machine Learning Research*, 2008.

[270] D. M. Chan, R. Rao, F. Huang, and J. F. Canny, "Gpu accelerated t-distributed stochastic neighbor embedding," *Journal of Parallel and Distributed Computing*, 2019.

[271] A. Bendale and T. E. Boult, "Towards open world recognition," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, IEEE Computer Society, 2015, pp. 1893–1902. DOI: 10.1109/CVPR.2015.7298799. [Online]. Available: https://doi.org/10.1109/CVPR.2015.7298799.

[272] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete Problems in AI Safety," pp. 1–29, 2016. arXiv: 1606.06565. [Online]. Available: http://arxiv.org/abs/1606.06565.

[273] J. Janai, F. Güney, A. Behl, and A. Geiger, "Computer vision for autonomous vehicles: problems, datasets and state of the art," *Found. Trends Comput. Graph. Vis.*, vol. 12, no. 1-3, pp. 1–308, 2020. DOI: 10.1561/0600000079. [Online]. Available: https://doi.org/10.1561/0600000079.

[274] J.-B. Alayrac, J. Donahue, P. Luc, *et al.*, "Flamingo: a Visual Language Model for Few-Shot Learning," in *Advances in Neural Information Processing Systems*, 2022. arXiv: 2204.14198. [Online]. Available: http://arxiv.org/abs/2204.14198.

[275] S. Liang, Y. Li, and R Srikant, "Enhancing The Reliability of Out-Of-Distribution Image Detection in Neural Networks," in *ICLR*, 2018. arXiv: 1706.02690v5. [Online]. Available: https://github.com/facebookresearch/odin.

[276] C. Shama Sastry and S. Oore, "Detecting out-of-distribution examples with in-distribution examples and gram matrices," *arXiv e-prints*, arXiv–1912, 2019.

[277] A. Paszke, S. Gross, F. Massa, *et al.*, "PyTorch: an imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., Curran Associates, Inc., 2019, pp. 8024–8035.

[278] P. Khosla, P. Teterwak, C. Wang, *et al.*, "Supervised contrastive learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 18 661–18 673, 2020.

[279] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, *The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results*, http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[280] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, "Learning to prompt for vision-language models," *International Journal of Computer Vision*, vol. 130, no. 9, pp. 2337–2348, 2022.

[281] Y. Ming, Z. Cai, J. Gu, Y. Sun, W. Li, and Y. Li, "Delving into out-of-distribution detection with vision-language representations," *Advances in Neural Information Processing Systems*, vol. 35, pp. 35 087–35 102, 2022.

[282] M. U. Khattak, H. Rasheed, M. Maaz, S. Khan, and F. S. Khan, "Maple: multi-modal prompt learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 19 113–19 122.

[283] A. Miyai, Q. Yu, G. Irie, and K. Aizawa, "Locoop: few-shot out-of-distribution detection via prompt learning," *arXiv preprint arXiv:2306.01293*, 2023.

[284] K. Liu, Z. Fu, C. Chen, *et al.*, "Category-extensible out-of-distribution detection via hierarchical context descriptions," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[285] A. Miyai, Q. Yu, G. Irie, and K. Aizawa, "Zero-shot in-distribution detection in multi-object settings using vision-language foundation models," *arXiv preprint arXiv:2304.04521*, 2023.

[286] N. Houlsby, A. Giurgiu, S. Jastrzebski, *et al.*, "Parameter-efficient transfer learning for nlp," in *International Conference on Machine Learning*, PMLR, 2019, pp. 2790–2799.

[287] C.-H. Tu, Z. Mai, and W.-L. Chao, "Visual query tuning: towards effective usage of intermediate representations for parameter and memory efficient transfer learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7725–7735.

[288] E. J. Hu, Y. Shen, P. Wallis, *et al.*, "Lora: low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[289] S. Goyal, A. Kumar, S. Garg, Z. Kolter, and A. Raghunathan, "Finetune like you pretrain: improved finetuning of zero-shot vision models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 19 338–19 347.

[290] N. Shazeer, A. Mirhoseini, K. Maziarz, *et al.*, "Outrageously large neural networks: the sparsely-gated mixture-of-experts layer," *arXiv preprint arXiv:1701.06538*, 2017.

[291] Y. Liu, J. Yan, Y. Chen, J. Liu, and H. Wu, "Smoa: sparse mixture of adapters to mitigate multiple dataset biases," *arXiv preprint arXiv:2302.14413*, 2023.

[292] Q. Zhang, B. Zou, R. An, J. Liu, and S. Zhang, "Split & merge: unlocking the potential of visual adapters via sparse training," *arXiv preprint arXiv:2312.02923*, 2023.

[293] F. M. Harper and J. A. Konstan, "The Movielens datasets: history and context," *ACM TIIS*, 2015.

[294] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[295] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, pp. 229–256, 1992.

[296] G. E. Hinton, "Training Products of Experts by Minimizing Contrastive Divergence," *Neural Computation*, vol. 1800, no. 14, pp. 1771–1800, 2002.

[297] T. Tieleman, "Training restricted boltzmann machines using approximations to the likelihood gradient," *Proceedings of the 25th International Conference on Machine Learning*, pp. 1064–1071, 2008. DOI: 10.1145/1390156.1390290.

[298] Y. Song and D. P. Kingma, "How to train your energy-based models," *CoRR*, vol. abs/2101.03288, 2021. arXiv: 2101.03288. [Online]. Available: https://arxiv.org/abs/2101.03288.

Appendix A

Supplementary material: Optimization of Ranking Losses for Image retrieval

A.1 Theoretical analysis.

A.1.1 Contradictory gradient flow for positives samples.

In the theoretical analysis of the main manuscript Sec. 3.4.1, we write that:

$$\boxed{\frac{\partial \mathcal{L}_{\text{SmoothAP}}}{\partial s_1} = -\frac{\partial \mathcal{L}_{\text{SmoothAP}}}{\partial s_2}}$$

To see this, we write:

$$\begin{aligned}\frac{\partial \mathcal{L}_{\text{SmoothAP}}}{\partial s_1} =& \frac{\partial \mathcal{L}_{\text{SmoothAP}}}{\partial \operatorname{rank}^+(x_1)} \cdot \frac{\partial \operatorname{rank}^+(x_1)}{\partial s_1} + \frac{\partial \mathcal{L}_{\text{SmoothAP}}}{\partial \operatorname{rank}^+(x_2)} \cdot \frac{\partial \operatorname{rank}^+(x_2)}{\partial s_1} \\ &+ \frac{\partial \mathcal{L}_{\text{SmoothAP}}}{\partial \operatorname{rank}^-(x_1)} \cdot \frac{\partial \operatorname{rank}^-(x_1)}{\partial s_1} + \frac{\partial \mathcal{L}_{\text{SmoothAP}}}{\partial \operatorname{rank}^-(x_2)} \cdot \frac{\partial \operatorname{rank}^-(x_2)}{\partial s_1}\end{aligned}$$

Because $\operatorname{rank}^-(x_2) = \sigma(\frac{s_3 - s_2}{\tau})$, we have $\frac{\partial \operatorname{rank}^-(x_2)}{\partial s_1} = 0$ and $\frac{\partial \operatorname{rank}^-(x_1)}{\partial s_1} = 0$ in the example of Fig. 3.8, because $\operatorname{rank}^-(x_1) = \sigma(\frac{s_3 - s_1}{\tau})$ and $s_3 - s_1$ falls into the saturation regime of the sigmoid. We get a similar result for the derivative of $\mathcal{L}_{\text{SmoothAP}}$ wrt. s_2 :

$$\frac{\partial \mathcal{L}_{\text{SmoothAP}}}{\partial s_2} = \frac{\partial \mathcal{L}_{\text{SmoothAP}}}{\partial \operatorname{rank}^+(x_1)} \cdot \frac{\partial \operatorname{rank}^+(x_1)}{\partial s_2} + \frac{\partial \mathcal{L}_{\text{SmoothAP}}}{\partial \operatorname{rank}^+(x_2)} \cdot \frac{\partial \operatorname{rank}^+(x_2)}{\partial s_2}$$

Furthermore, we have :

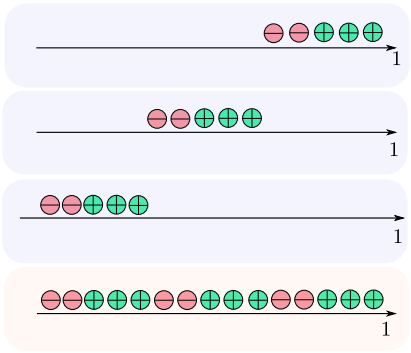


Figure A.1: The worst case when computing the global AP would be that each batch is juxtaposed.

$$\frac{\partial \operatorname{rank}^+(x_1)}{\partial s_1} = -\frac{\partial \operatorname{rank}^+(x_1)}{\partial s_2}$$

Indeed $\operatorname{rank}^+(x_1) = 1 + \sigma(\frac{s_2 - s_1}{\tau})$, such that $\frac{\partial \operatorname{rank}^+(x_1)}{\partial s_1} = -\tau \cdot \sigma(\frac{s_2-s_1}{\tau}) \left(1 - \sigma(\frac{s_2-s_1}{\tau})\right)$ and $\frac{\partial \operatorname{rank}^+(x_1)}{\partial s_2} = \tau \cdot \sigma(\frac{s_2-s_1}{\tau}) \left(1 - \sigma(\frac{s_2-s_1}{\tau})\right)$. Similarly, the derivatives of $\operatorname{rank}^+(x_2)$ wrt. s_1 and s_2 also have opposite signs: $\frac{\partial \operatorname{rank}^+(x_2)}{\partial s_1} = -\frac{\partial \operatorname{rank}^+(x_2)}{\partial s_2}$.
It concludes the proof that $\frac{\partial \mathcal{L}_{\text{SmoothAP}}}{\partial s_1} = -\frac{\partial \mathcal{L}_{\text{SmoothAP}}}{\partial s_2}$. □

A.1.2 Upper bounds on the decomposability gap.

A.1.3 Proof of Eq. (3.18): Upper bound on the DG_{AP} with no $\mathcal{L}_{\text{DG}}$.

We choose a setting for the proof of the upper bound similar to the one used for training, *i.e.* all the batch have the same size, and the number of positive instances per batch (*i.e.* Ω_b^+) is the same.

Eq. (3.18) gives an upper bound for DG. This upper bound is given in the worst case: when the AP has the lowest value guaranteed by the AP on each batch. We illustrate this case in Fig. A.1.

In Eq. (3.18) the 1 in the right-hand term comes from the average of AP over all batches:

$$\frac{1}{K} \sum_{b=1}^{K} \text{AP}_i^b(\theta) = 1$$

We then justify the term in the parenthesis of Eq. (3.18), which is the lower bound of the AP. In the global ordering the positive instances are ranked after all the positive instances

from previous batches giving the following rank$^+$: $j + |\Omega_1^+| + \cdots + |\Omega_{b-1}^+|$, with j the rank$^+$ in the batch, Positive instances are also ranked after all negative instances from previous batches giving rank$^-$: $|\Omega_1^-| + \cdots + |\Omega_{b-1}^-|$. $\square$

A.1.4 Proof of Eq. (3.19) Upper bound on the DG with $\mathcal{L}_{\mathbf{DG}}$.

We now write that each positive instance that respects the constraint of $\mathcal{L}_{\text{DG}}$ is ranked after the positive instances of previous batches that respect the constraint giving the following rank$^+$: $j + G_1^+ + \cdots + G_{b-1}^+$, with j the rank$^+$ in the current batch. Positive instances are also ranked after the negative instances of previous batches that do not respect the constraints, yielding rank$^-$: $E_1^- + \cdots + E_{b-1}^-$.

We then write that positive instances that do not respect the constraints are ranked after all positive instances from previous batches and the positive instances respecting the constraints of the current batch, giving rank$^+$: $j + G_b^+|\Omega_1^+| + \cdots + |\Omega_{b-1}^+|$. They also are ranked after all the negative instances from previous batches giving rank$^-$: $|\Omega_1^-| + \cdots + |\Omega_{b-1}^-|$. $\square$

Appendix B

Supplementary material: Hierarchical Image Retrieval for Robust Ranking

B.1 Proof of Property 1.

Denoting $\Sigma w\text{AP} := \sum_{l=1}^{L} w_l \cdot AP^{(l)}$, we obtain from Eq. (4.8):

$$\Sigma w\text{AP} = \sum_{l=1}^{L} w_l \cdot \frac{1}{|\Omega^{+,l}|} \sum_{k \in \Omega^{+,l}} \frac{\text{rank}^{+,l}(k)}{\text{rank}(k)} \tag{B.1}$$

We define $\hat{w}_l = \frac{w_l}{|\Omega^{+,l}|}$ to ease notations, so:

$$\Sigma w\text{AP} = \sum_{l=1}^{L} \hat{w}_l \sum_{k \in \Omega^{+,l}} \frac{\text{rank}^{+,l}(k)}{\text{rank}(k)} \tag{B.2}$$

We define $\mathbb{1}(k,l) = \mathbb{1}\left[k \in \Omega^{+,l}\right]$ so that we can sum over Ω^{+} instead of $\Omega^{+,l}$ and inverse the summations. Note that rank does not depend on l, contrary to $\text{rank}^{+,l}$.

$$\Sigma w\text{AP} = \sum_{l=1}^{L} \sum_{k \in \Omega^{+}} \frac{\hat{w}_l \cdot \mathbb{1}(k,l) \cdot \text{rank}^{+,l}(k)}{\text{rank}(k)} \tag{B.3}$$

$$= \sum_{k \in \Omega^{+}} \sum_{l=1}^{L} \frac{\hat{w}_l \cdot \mathbb{1}(k,l) \cdot \text{rank}^{+,l}(k)}{\text{rank}(k)} \tag{B.4}$$

$$= \sum_{k \in \Omega^{+}} \frac{\sum_{l=1}^{L} \mathbb{1}(k,l) \cdot \hat{w}_l \cdot \text{rank}^{+,l}(k)}{\text{rank}(k)} \tag{B.5}$$

We replace $\text{rank}^{+,l}$ in Eq. (B.5) with its definition from Eq. (4.8):

$$\Sigma w\text{AP} = \sum_{k\in\Omega^+} \frac{\sum_{l=1}^{L} \mathbb{1}(k,l)\cdot\hat{w}_l\cdot\left(1+\sum_{j\in\Omega^{+,l}} H(s_j-s_k)\right)}{\text{rank}(k)} \tag{B.6}$$

$$= \sum_{k\in\Omega^+} \frac{\sum_{l=1}^{L} \mathbb{1}(k,l)\cdot\hat{w}_l + \sum_{l=1}^{L}\sum_{j\in\Omega^{+,l}} \mathbb{1}(k,l)\cdot\hat{w}_l\cdot H(s_j-s_k)}{\text{rank}(k)} \tag{B.7}$$

$$= \sum_{k\in\Omega^+} \frac{\sum_{l=1}^{L} \mathbb{1}(k,l)\cdot\hat{w}_l + \sum_{l=1}^{L}\sum_{j\in\Omega^{+}} \mathbb{1}(j,l)\cdot\mathbb{1}(k,l)\cdot\hat{w}_l\cdot H(s_j-s_k)}{\text{rank}(k)} \tag{B.8}$$

$$= \sum_{k\in\Omega^+} \frac{\sum_{l=1}^{L} \mathbb{1}(k,l)\cdot\hat{w}_l + \sum_{j\in\Omega^{+}}\sum_{l=1}^{L} \mathbb{1}(j,l)\cdot\mathbb{1}(k,l)\cdot\hat{w}_l\cdot H(s_j-s_k)}{\text{rank}(k)} \tag{B.9}$$

We define the following relevance function:

$$\text{rel}(k) = \sum_{l=1}^{L} \mathbb{1}(k,l)\cdot\hat{w}_l \tag{B.10}$$

By construction of $\mathbb{1}(\cdot,l)$:

$$\sum_{l=1}^{L} \mathbb{1}(j,l)\cdot\mathbb{1}(k,l)\cdot\hat{w}_l = \min(\text{rel}(k),\text{rel}(j)) \tag{B.11}$$

Using the definition of the relevance function from Eq. (B.10) and Eq. (B.11), we can rewrite Eq. (B.9) with $\mathcal{H}$-rank:

$$\Sigma w\text{AP} = \sum_{k\in\Omega^+} \frac{\text{rel}(k) + \sum_{j\in\Omega^+}\min(\text{rel}(j),\text{rel}(k))\cdot H(s_j-s_k)}{\text{rank}(k)} \tag{B.12}$$

$$= \sum_{k\in\Omega^+} \frac{\mathcal{H}\text{-rank}(k)}{\text{rank}(k)} \tag{B.13}$$

Eq. (B.13) lacks the normalization constant $\sum_{k\in\Omega^+}\text{rel}(k)$ in order to have the same shape

as $\mathcal{H}$-AP in Eq. (4.3). So we must prove that $\sum_{k\in\Omega^+} \text{rel}(k) = 1$:

$$\sum_{k\in\Omega^+} \text{rel}(k) = \sum_{k\in\Omega^+} \sum_{l=1}^{L} \mathbb{1}(k,l) \cdot \hat{w}_l \tag{B.14}$$

$$= \sum_{l=1}^{L} |\Omega^{(l)}| \sum_{p=1}^{l} \hat{w}_p \tag{B.15}$$

$$= \sum_{l=1}^{L} |\Omega^{(l)}| \sum_{p=1}^{l} \frac{w_p}{|\Omega^{+,p}|} \tag{B.16}$$

$$= \sum_{l=1}^{L} |\Omega^{(l)}| \sum_{p=1}^{l} \frac{w_p}{|\bigcup_{q=p}^{L} \Omega^{(q)}|} \tag{B.17}$$

$$= \sum_{l=1}^{L} |\Omega^{(l)}| \sum_{p=1}^{l} \frac{w_p}{\sum_{q=p}^{L} |\Omega^{(q)}|} \tag{B.18}$$

$$= \sum_{l=1}^{L} \sum_{p=1}^{l} \frac{|\Omega^{(l)}| \cdot w_p}{\sum_{q=p}^{L} |\Omega^{(q)}|} \tag{B.19}$$

$$= \sum_{p=1}^{L} \sum_{l=p}^{L} \frac{|\Omega^{(l)}| \cdot w_p}{\sum_{q=p}^{L} |\Omega^{(q)}|} \tag{B.20}$$

$$= \sum_{p=1}^{L} w_p \cdot \frac{\sum_{l=p}^{L} |\Omega^{(l)}|}{\sum_{q=p}^{L} |\Omega^{(q)}|} \tag{B.21}$$

$$= \sum_{p=1}^{L} w_p = 1 \tag{B.22}$$

We have proved that $\Sigma w\text{AP} = \mathcal{H}$-AP with the relevance function of Eq. (B.10):

$$\Sigma w\text{AP} = \frac{1}{\sum_{k\in\Omega^+} \text{rel}(k)} \sum_{k\in\Omega^+} \frac{\mathcal{H}\text{-rank}(k)}{\text{rank}(k)} = \mathcal{H}\text{-AP} \tag{B.23}$$

Finally, we show, for an instance $k \in \Omega^{(l)}$, :

$$\text{rel}(k) = \sum_{p=1}^{L} \mathbb{1}(k,p) \cdot \hat{w}_p = \sum_{p=1}^{l} \cdot \hat{w}_p = \sum_{p=1}^{l} \frac{w_p}{|\Omega^{+,p}|} \tag{B.24}$$

i.e. the relevance of Eq. (B.10) is the same as the relevance of Property 1. This concludes the proof of Property 1. □

Appendix C

Supplementary material: Post-hoc out-of-distribution detection

C.1 Energy-based models.

An energy-based model (EBM) is an unnormalized density model defined via its energy function $E_\theta : \mathbb{R}^m \to \mathbb{R}$ parameterized by a neural network with parameters θ. For $\mathbf{z} \in \mathbb{R}^m$, its probability density is given by the Boltzmann distribution

$$p_\theta(\mathbf{z}) = \frac{1}{Z_\theta} \exp\left(-E_\theta(\mathbf{z})\right), \tag{C.1}$$

where Z_θ is the partition function which is intractable in high dimension. We can train EBMs via maximum likelihood estimation:

$$\arg\max_\theta \ \log p_\theta(\mathcal{D}) = \arg\min_\theta \ \mathbb{E}_{\mathbf{z}\sim p_{in}}[-\log p_\theta(\mathbf{z})] \tag{C.2}$$

which can be approximated via stochastic gradient descent :

$$\theta_{i+1} = \theta_i - \lambda \nabla_\theta(-\log p_{\theta_i}(\mathbf{z})) \quad \text{with} \quad \mathbf{z} \sim p_{in} \tag{C.3}$$

Interestingly, $\nabla_\theta(-\log p_{\theta_i}(\mathbf{z}))$ can be computed without computing the intractable normalization constant Z_θ.

We have:

$$\begin{aligned}
\nabla_\theta(-\log p_\theta(\mathbf{z})) &= \nabla_\theta E_\theta(\mathbf{z}) + \nabla_\theta \log Z_\theta \\
&= \nabla_\theta E_\theta(\mathbf{z}) + \frac{1}{Z_\theta}\nabla_\theta Z_\theta \\
&= \nabla_\theta E_\theta(\mathbf{z}) + \frac{1}{Z_\theta}\nabla_\theta \int_{\mathbf{z}} \exp(-E_\theta(\mathbf{z}))d\mathbf{z} \\
&= \nabla_\theta E_\theta(\mathbf{z}) + \frac{1}{Z_\theta}\int_{\mathbf{z}} \nabla_\theta \exp(-E_\theta(\mathbf{z}))d\mathbf{z} \\
&= \nabla_\theta E_\theta(\mathbf{z}) + \int_{\mathbf{z}} -\nabla_\theta E_\theta(\mathbf{z})\frac{\exp(-E_\theta(\mathbf{z}))}{Z_\theta}d\mathbf{z} \\
&= \nabla_\theta E_\theta(\mathbf{z}) - \mathbb{E}_{\mathbf{z}'\sim p_\theta}[\nabla_\theta E_\theta(\mathbf{z}')].
\end{aligned}$$

Therefore, training EBMs via maximum likelihood estimation (MLE) amounts to perform stochastic gradient descent with the following loss:

$$\mathcal{L}_{\text{MLE}} = \mathbb{E}_{\mathbf{z}\sim p_{in}}\Big[E_\theta(\mathbf{z})\Big] - \mathbb{E}_{\mathbf{z}'\sim p_\theta}\Big[E_\theta(\mathbf{z}')\Big]. \tag{C.4}$$

Intuitively, this loss amounts to diminishing the energy for samples from the true data distribution $p(x)$ and to increasing the energy for synthesized examples sampled according to the current model. Eventually, the gradients of the energy function will be equivalent for samples from the model and the true data distribution and the loss term will be zero.

The expectation $\mathbb{E}_{\mathbf{z}'\sim p_\theta}\Big[E_\theta(\mathbf{z}')\Big]$ can be approximated through MCMC sampling, but we need to sample z' from the model, p_θ, which is an unknown moving density. To estimate the expectation under p_θ in the right hand-side of equation (C.4) we must sample according to the energy-based model p_θ. To generate synthesized examples from p_θ, we can use gradient-based MCMC sampling, such as Stochastic Gradient Langevin Dynamics (SGLD) [216] or Hamiltonian Monte Carlo (HMC) [217]. In this work, we use SGLD sampling following [218], [219]. In SGLD, initial features are sampled from a proposal distribution p_0 and are updated for T steps with the following iterative rule:

$$\mathbf{z}_{t+1} = \mathbf{z}_t - \frac{\eta}{2}\nabla_{\mathbf{z}} E^h_{\theta_k}(\mathbf{z}_t) + \sqrt{\eta}\; w_t, \quad \text{with } w_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \tag{C.5}$$

where η is the step size. Therefore, sampling from p_θ does not require computing the normalization constant Z_θ.

Many variants of this training procedure have been proposed including Contrastive Divergence (CD) [296] where $p_0 = p_{\text{data}}$, or Persistent Contrastive Divergence (PCD) [297] which uses a buffer to extend the length of the MCMC chains. We refer the reader to [298] for more details on EBM training with MLE as well as other alternative training strategies (score-matching, noise contrastive estimation, Stein discrepancy minimization, etc.).

C.2 Experimental results

C.2.1 ViT results

In Tab. C.1 we compare HEAT using a Vision Transformer[1] (ViT), on the ImageNet benchmark introduced in [211]. We show that on the aggregated results, HEAT outperforms the previous best method, ViM [211], by -1.7 pts FPR95. Importantly, HEAT outperforms other method on three datasets of the benchmark, *i.e.* OpenImage-O, Textures, ImageNet-O, and is competitive on iNaturalist. Tab. C.1 demonstrates the ability of HEAT to adapt to architectures of neural networks, *i.e.* Vision Transformer [46], other than the convolutional networks (*i.e.* ResNet-34 & ResNet-50) tested in Sec. 5.3.2.

Table C.1: **Results on Imagenet.** All methods are based on an ImageNet pre-trained **Vision Transformer** (ViT) model. ↑ indicates larger is better, and ↓ the opposite.

Method	OpenImage-O	Textures	iNaturalist	Imagenet-O	Average
	FPR95↓ / AUC ↑	FPR95↓ / AUC ↑	FPR95↓ / AUC ↑	FPR95↓ / AUC ↑	FPR95↓ / AUC ↑
MSP	34.2 / 92.5	48.6 / 87.1	19.0 / 96.1	64.8 / 81.9	41.7 / 89.4
EL	14.0 / 97.1	28.2 / 93.4	6.2 / 98.7	41.3 / 90.5	22.4 / 94.9
ODIN	15.7 / 96.9	30.6 / 93.0	6.6 / 98.6	44.2 / 89.9	24.3 / 94.6
MaxLogit	15.7 / 96.9	30.6 / 93.0	6.6 / 98.6	44.2 / 89.9	24.3 / 94.6
KL Matching	28.5 / 93.9	44.1 / 88.8	14.8 / 96.9	55.7 / 84.1	35.8 / 90.9
KNN	45.8 / 91.7	28.9 / 93.2	52.3 / 91.1	52.9 / 88.4	45.0 / 91.1
Residual	32.6 / 92.7	33.8 / 92.2	6.6 / 98.6	47.9 / 88.2	30.2 / 92.9
ReAct	13.5 / 97.4	28.5 / 93.3	4.3 / 99.0	42.6 / 90.7	22.2 / 95.1
Mahalanobis	13.5 / 97.5	25.2 / 94.2	**2.1** / **99.5**	37.0 / 92.8	19.5 / 96.0
ViM	12.6 / 97.6	20.3 / 95.3	2.6 / **99.4**	36.8 / 92.6	18.1 / 96.2
HEAT	**11.2** / **97.8**	**12.8** / **96.9**	6.9 / 98.2	**34.8** / **93.1**	**16.4** / **96.5**

C.2.2 Model analysis

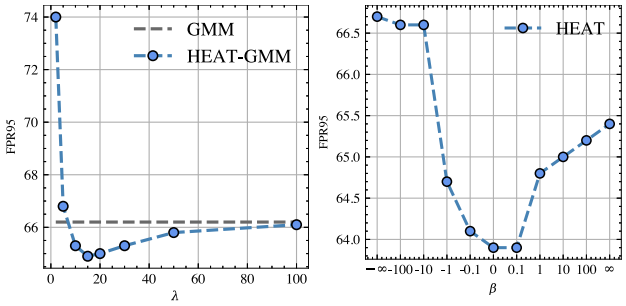
In Fig. C.1 we show the impact of λ in Eq. (5.6) and β *vs.*FPR95 on CIFAR-100, we study in Fig. C.2 how HEAT behaves on low data regimes with CIFAR-10 as ID dataset.

Robustness to λ In Fig. C.1a we can see that we have similar trends to Fig. 5.3a. For values of λ too high, *i.e.* when the expressivity of the energy-based correction is limited, HEAT-GMM

[1]The model used can be found at https://github.com/haoqiwang/vim

has the same performances as GMM. For values of λ too low, the energy-based correction is not controlled and disregards the prior scorer, *i.e.*GMM. Finally, for a wide range of λ values, HEAT-GMM improves the OOD detection performances of GMM.

Robustness to β In Fig. C.1b we show that HEAT is stable wrt. β on CIFAR-100 similarly to Fig. 5.3b.



(a) λ *vs.*FPR95↓ (b) β *vs.*FPR95↓

Figure C.1: On CIFAR-100 ID: (a) impact of λ in Eq. (5.6) *vs.* FPR95 and (b) analysis of β in Eq. (5.7) *vs.* FPR95.

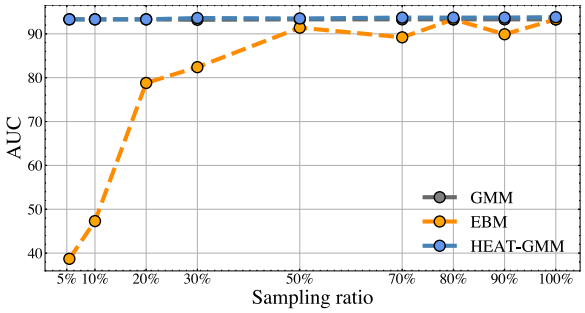


Figure C.2: Impact on performances (AUC↑ on CIFAR-10) *vs.* the number of training data for GMM density, fully data-driven EBM, and HEAT. Our hybrid approach maintains strong performances in low-data regimes, in contrast to the fully data-driven EBM.

Low data regime Similarly to Fig. 5.4, we can see that training solely an EBM is very unstable when the number of data is low. On the other hand, HEAT-GMM is stable to the lack of data and improves GMM even with few ID samples available.

A L'été de l'IA.

Ces dernières années, le domaine de l'intelligence artificielle (IA) a connu une transformation remarquable, propulsée par le renouveau de l'apprentissage profond (DL) [1]. Il a révolutionné de nombreux domaines. L'IA générative a rencontré un énorme succès avec les grands modèles de langage (LLM), tels que ChatGPT 3 & 4 [2], [3] (Fig. C.3f), Llama 1 & 2 [4], [5], et la génération d'images, par exemple la génération de texte vers image avec Stable Diffusion [6] (Fig. C.3e) ou DALL-E [7]. Cela a changé notre manière de représenter le contenu multimédia, comme les images avec DINOv2 [8], MAE [9], SAM [10] (Fig. C.3c), le son avec Whisper [11] (Fig. C.3b), et plus récemment les données multimodales avec CLIP [12] (Fig. C.3a), ImageBind [13] et Gemini [14]. Son application en apprentissage par renforcement a permis de maîtriser le jeu de Go avec AlphaGo [15], battant par la suite son champion du monde (Fig. C.3d), et a permis en robotique de battre les champions de drones de course [16], ou pour la conduite autonome [17]. Ses applications sont diverses, et elle a été appliquée avec succès à la recherche en physique, par exemple la prévision météorologique avec GraphCast [18] ou la stabilisation du plasma dans la fusion nucléaire [19]. Elle a eu un impact considérable sur la biologie avec la sortie d'AlphaFold [20], et est au cœur d'entreprises telles qu'Altos lab qui utilise l'IA pour rajeunir les cellules avec la réaction de Shinya Yamanaka [21]. Un autre domaine prolifique est l'application de l'IA à la médecine, par exemple la reconstruction plus rapide d'IRM [22], la segmentation des organes sur des radios [23], ou la découverte de médicaments [24].

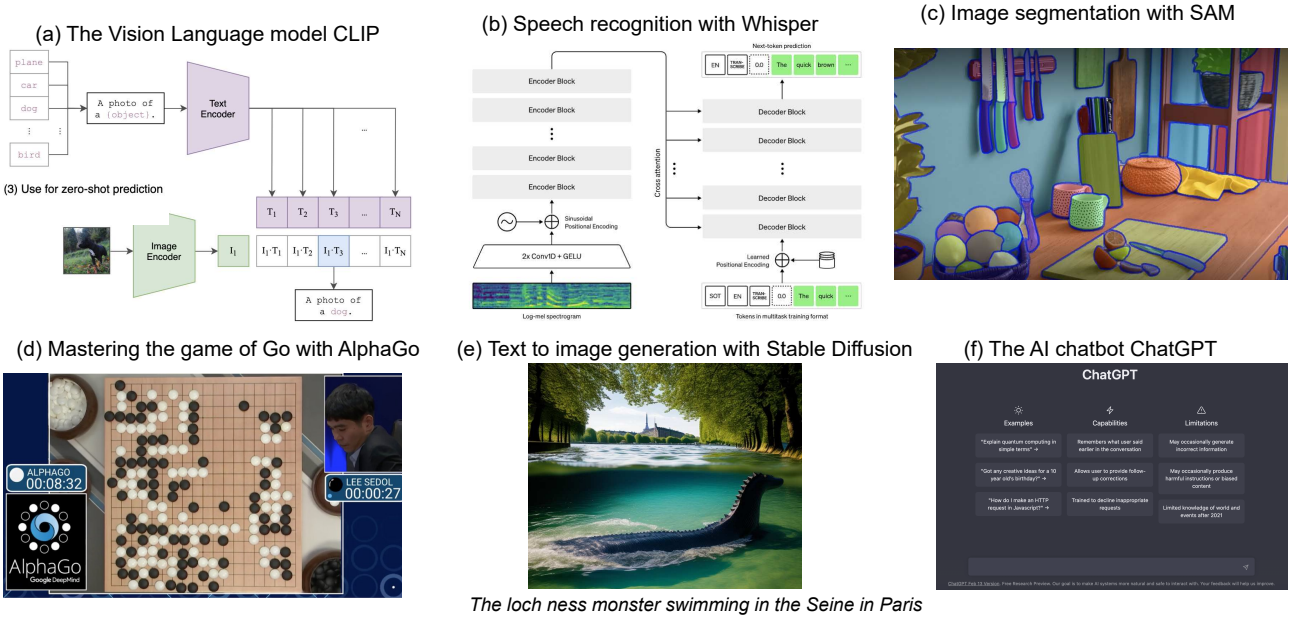


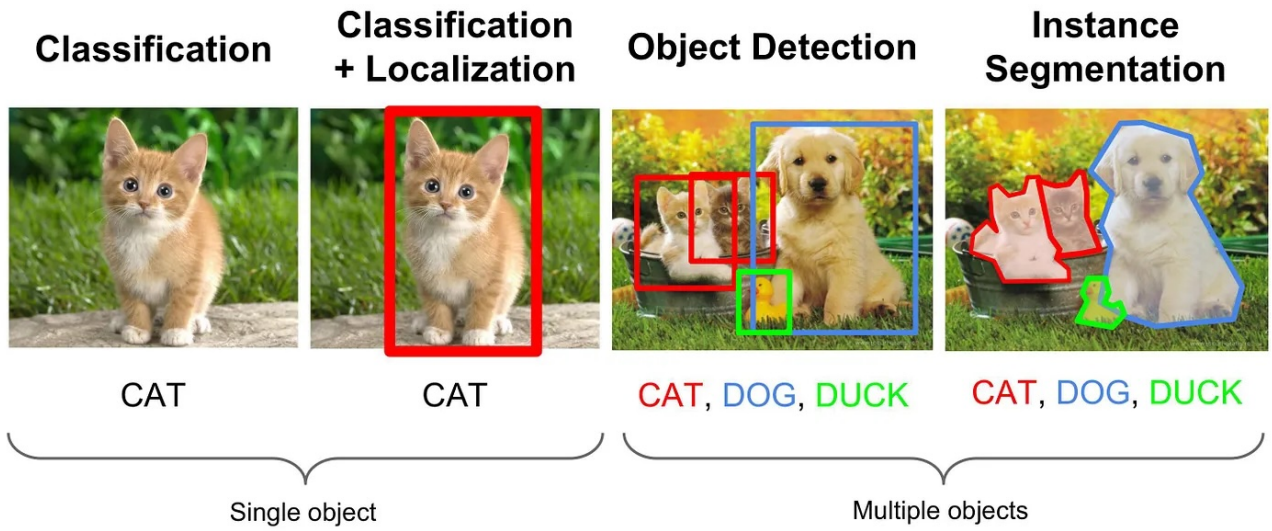
Figure C.3: Exemples de recherches en apprentissage profond ayant un impact élevé sur la communauté de l'IA et sur un large public. ChatGPT (f) a eu 1,7 *milliard* de visites en octobre 2023*.

* source : https://explodingtopics.com/blog/chatgpt-users

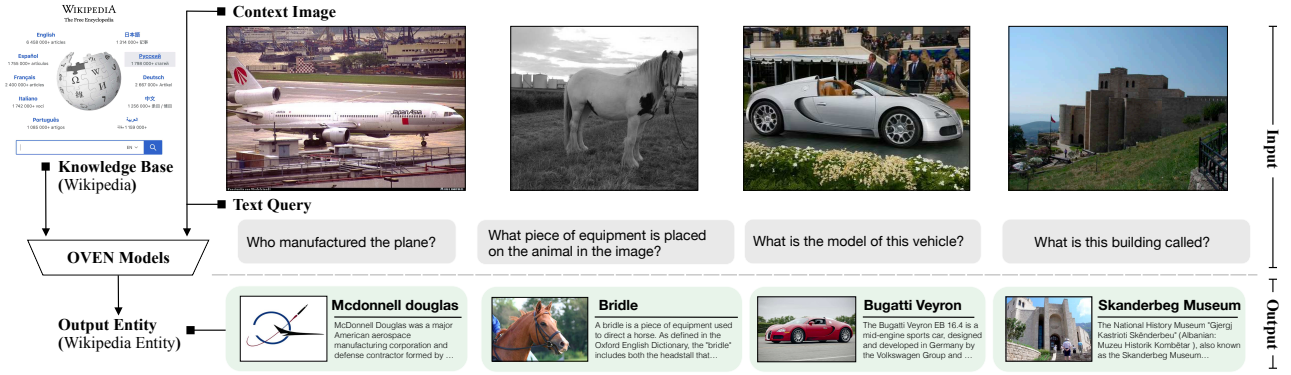
Dans les principaux domaines de l'IA, tels que la vision par ordinateur, le traitement du langage naturel ou la reconnaissance vocale, l'apprentissage profond est devenu le paradigme dominant, voire le seul. Cette dominance de l'apprentissage profond découle de deux facteurs majeurs : 1) la large disponibilité de grands ensembles de données tels que ImageNet [25], COCO [26], ADE20K [27], le jeu de données Google Landmarks v2 [28] ou iNaturalist [29] et ces dernières années la collecte de jeux de données massifs de plusieurs ordres de grandeurs plus grands, tels que JFM [30] (privé), LAION-5B [31], LVD-142M [8]; 2) la quantité de calcul de plus en plus importante disponible pour les laboratoires industriels et les chercheurs, avec une recherche dédiée au développement des meilleures puces [32], [33]. Ces deux facteurs permettent d'entraîner des modèles plus grands, pendant plus longtemps et sur des données très diverses.

En augmentant continuellement la taille des ensembles de données et la puissance de calcul, l'apprentissage profond entre dans l'ère des modèles de fondation [34]. Ces modèles qui ont été entraînés sur de grandes quantités de données, par exemple deux *milliards* de paires texte-image pour OpenClip [35] ou deux *trillions* de tokens pour Llama 2 [5]. En raison de l'ampleur de leur ensemble d'apprentissage, ces modèles sont dits ayant "vu le monde", c'est-à-dire qu'ils ont vu des données diverses et ont une compréhension globale du monde. Ces modèles généralistes peuvent être utilisés de manière "zero-shot", c'est-à-dire sans nécessiter d'entraînement, sur un large éventail de tâches. Ils sont conçus pour avoir une compréhension générale des données sans être experts dans des tâches spécifiques. Ainsi, en raison de leur généralité, ils peuvent sous-performer par rapport aux modèles experts sur des tâches ou des ensembles de données spécifiques, par exemple CLIP dans la recherche d'images [36], ou SAM sur des images médicales [37]. Bien que ces modèles soient censés être à la base de nombreux systèmes d'IA, les adapter reste le moyen le plus efficace pour des tâches spécifiques.

Vision par ordinateur moderne. En vision par ordinateur, l'utilisation de l'apprentissage profond redéfinit la manière dont les images sont traitées et représentées. Les premières applications réussies de l'apprentissage profond en vision informatique étaient basées sur les réseaux de neurones convolutifs modernes [40], avant de devenir la base de toutes les méthodes de vision par ordinateur après leurs succès à grande échelle avec MCDNN [41] et le célèbre AlexNet [42] qui a remporté le défi ILSVRC 2012 [43]. Les architectures ont ensuite évolué avec les bien connus VGG [44] et ResNets [45]. Récemment, les transformers vision, ViTs [46], [47], ont été développés en vision par ordinateur en adaptant le Transformer [48] issue du traitement du langage naturel. Le succès de l'apprentissage profond en vision par ordinateur vient notamment du fait qu'il apprend des représentations, les "embeddings" ou "deep features", plutôt que de s'appuyer sur des "features expertes" ingénierées telles que SIFT [49]. En effet, les features



(a) Image tirée de [38]. Exemple de différentes tâches de vision par ordinateur résolues à l'aide de réseaux de neurones profonds.



(b) Image tirée de [39]. Système de réponse à des questions visuelles basé sur la recherche d'images.

Figure C.4: Illustration des applications des DNN en vision par ordinateur.

expertes ont été conçues par les chercheurs en utilisant des notions de traitement du signal et notre compréhension des aspects importants des images, par exemple les gradients de couleur. Ces caractéristiques ingénierées saisissent principalement des indices de bas niveau, et elles peuvent donc manquer d'expressivité pour le contenu sémantique, tandis que les deep features sont basées sur les données et peuvent représenter différents niveaux d'abstraction pour s'attaquer aux tâches à résoudre. Les embeddings sont des vecteurs de grande dimension qui peuvent représenter des images complexes de manière compacte et permettent la comparaison à l'aide d'outils simples tels que la distance euclidienne. Ces représentations complexes ont permis aux systèmes d'apprentissage profond d'effectuer de nombreuses tâches telles que la classification d'images [44], [45], [50], la détection d'objets [51]–[53], la segmentation d'images [10], [54], [55],

qui sont illustrées sur la Fig. C.4a, ou la réponse à des questions visuelles [39], [56], illustrée sur la Fig. C.4b, *etc.*

B Contexte et motivations.

Cette thèse découle de la collaboration entre le Cnam et Coexya. Coexya[2] est une entreprise privée qui, entre autres, édite des solutions logicielles dédiées à la gestion de la propriété intellectuelle (PI). Parmi celles-ci figure Acsepto[3], une suite logicielle pour la recherche et la surveillance de marques, utilisée dans 16 institutions de propriété intellectuelle dans le monde. L'un des moteurs de recherche d'Acsepto est dédié à la recherche de logos de marques (TMs). En effet, lorsqu'une personne, une organisation ou une entreprise souhaite protéger un logo de marque, elle doit soumettre une demande de marque à l'office de la PI du pays concerné, tel que l'INPI[4] en France, pour s'assurer que le TM candidat n'est pas similaire à un logo déjà existant.

En raison de l'ampleur de leur base de données, par exemple 3.2 millions de TMs enregistrés en France, les offices de la PI ont dû automatiser le processus de recherche. Ce processus de recherche est illustré par la Fig. C.5. Étant donné un logo, appelé "requête", qu'un demandeur souhaiterait enregistrer, par exemple le logo ICML ici, Acsepto récupère une liste de logos dans la base de données d'un client qui sont les plus similaires à la requête.

Requêter une base de données avec une image est une tâche de vision par ordinateur appelée recherche d'images basée sur le contenu (CBIR). Le CBIR est basé sur la *représentation d'image*. Cela consiste à construire des représentations d'images qui permettent leur comparaison. En effet, comparer deux images en fonction de leurs pixels bruts, par exemple en calculant la distance L_2 pixel à pixel, n'est pas précis et est très sensible aux petites variations d'une image. Cela est illustré sur la Fig. C.6, en traduisant une image du jeu de données MNIST [57] de quelques pixels, la distance L_2 devient plus grande qu'avec une autre image complètement différente. Acsepto était donc basé dans ses versions précédentes sur des représentations d'images ingéniérées; sa version finale avant l'apprentissage profond était basée sur un mélange de plusieurs algorithmes, comprenant la transformée de Fourier 2D [58], les polynômes de Zernike 2D [59], les features HOG [60], les features SURF [61], les features FCTH [62] et un algorithme interne. Les représentations issues de ces différents algorithmes permettaient de se concentrer sur différents aspects des images, par exemple les formes ou les gradients de couleur.

[2]Site web de Coexya : https://www.coexya.eu/
[3]Description d'Acsepto : Fiches-produits-Acsepto.pdf
[4]Site web de l'INPI : https://www.inpi.fr/

Figure C.5: Exemple d'une requête dans Acsepto avec le logo ICML. Contrairement au cadre de recherche par images standard en milieu académique, pour la plupart des requêtes de logos, il n'y a pas nécessairement de résultats positifs. Certains logos sont plus pertinents que d'autres.

Coexya a depuis adopté les embeddings suite à l'avènement de l'apprentissage profond en vision par ordinateur. L'une des forces des DNN est qu'ils sont capables d'apprendre des représentations des images basées sur les données : on dit que les DNN sont "data-driven". Cela permet de créer des espaces d'embeddings où les distances sont perceptuelles. Cela signifie que deux images qui sont visuellement ou sémantiquement similaires seront proches dans le sens de la distance euclidienne. Les premiers modèles profonds de Coexya reposent sur le fine-tuning des DNN pré-entraînés sur ImageNet, par exemple le ResNet-50 [45], sur leur base de données interne annotées avec la classification de Vienne[5], une classification multi-étiquettes standardisée des logos de marques établie par l'Organisation mondiale de la propriété intellectuelle[6]. Les caractéristiques profondes extraites de ces DNN fine-tunés sont ensuite utilisées pour comparer les logos de marque les uns avec les autres. Le fine-tuning est important, car il permet d'adapter le modèle à un domaine différent, par exemple les logos de marques pour Coexya. Il permet également d'apprendre des représentations qui peuvent distinguer les différences subtiles dans les images, en effet les ensembles de données en recherche par images sont "fine-grained", ce qui n'est pas le cas pour les ensembles de données généralistes tels qu'ImageNet. Avec cette

[5]La classification de Vienne : https://www.wipo.int/classifications/vienna/en/index.html
[6]WIPO : https://www.wipo.int/portal/en/index.html

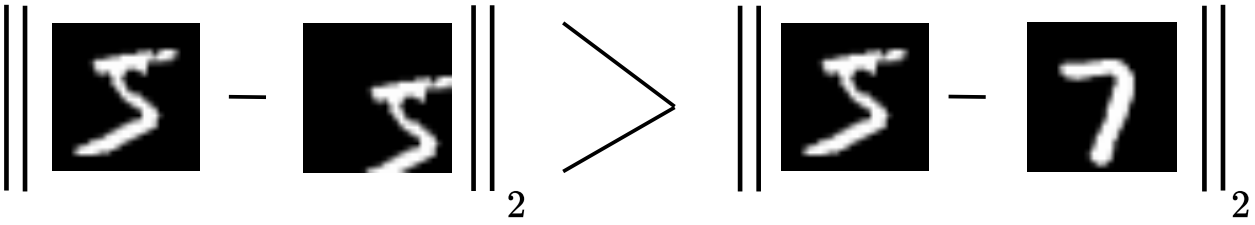


Figure C.6: La distance L_2 pixel à pixel entre une image et une version translatée d'elle-même est plus grande qu'avec une autre image complètement différente. Cela illustre la nécessité de concevoir des représentations plus puissantes pour comparer des images.

collaboration, Coexya a cherché à améliorer leurs modèles utilisés dans leur logiciel Acsepto, en améliorant leurs performances prédictives et en les rendant plus fiables.

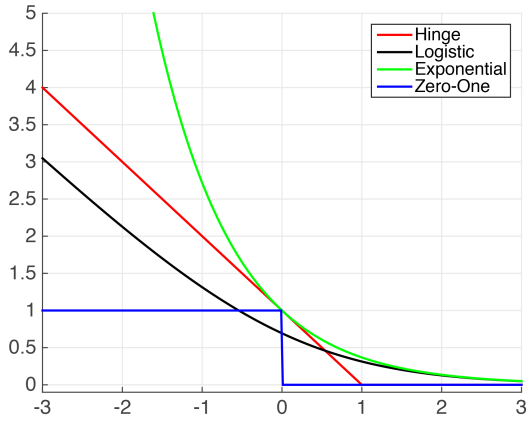
Ce riche contexte nous amène à aborder la notion de robustesse des DNN selon trois perspectives différentes :

1. Défi 1 : Robustesse en optimisation, où nous concevons une fonction de coût théoriquement justifiée, conduisant à de meilleures performances sur les métriques d'évaluation.
2. Défi 2 : Robustesse des classements, pour atténuer la *sévérité des erreurs* et garantir l'alignement du classement avec les préférences humaines en s'appuyant sur des annotations hiérarchiques.
3. Défi 3 : Robustesse des modèles, en détectant les images hors distribution en utilisant des modèles "data-driven" pour estimer la densité des images d'entraînement.

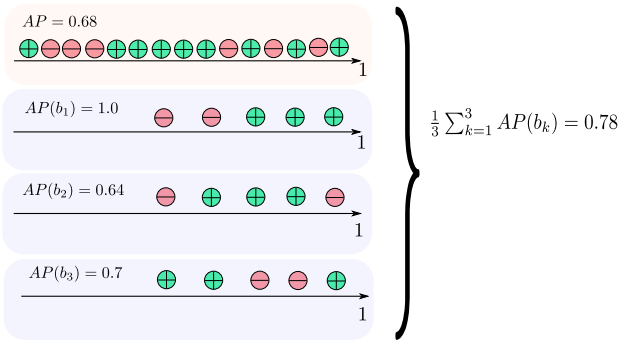
B.1 Défi 1 : Optimisation de métriques non lisses et non décomposables.

Pour apprendre des représentations, les DNN sont entraînés sur un ensemble de données en minimisant une fonction de coût. Les gradients sont calculés à partir des sorties des DNN. En utilisant l'algorithme de rétro-propagation [63], un gradient est calculé pour chaque couche du DNN. Les poids sont ensuite mis à jour en utilisant la descente de gradient stochastique (SGD). Ce paradigme d'entraînement repose sur des fonctions de coût qui sont différentiables, c'est-à-dire que le gradient de la fonction de coût par rapport à la sortie du DNN peut être calculé et est informatif. Pour avoir un DNN qui est entraîné pour une tâche spécifique, le meilleur scénario est de pouvoir optimiser les métriques d'évaluation pendant l'entraînement. Par exemple, c'est possible pour les métriques de régression standard : l'erreur quadratique moyenne (MSE). Cependant, pour plusieurs fonctions de coût, ce n'est pas possible, par exemple pour la fonction de coût 0/1 utilisée en classification, illustrée en noir sur la Fig. C.7a. En effet,

c'est une fonction en escalier et ses gradients sont soit nuls, soit indéfinis, ce qui les rend non informatifs pour la SGD. Cela nécessite donc l'utilisation d'une fonction de coût de substitution qui est différentiable, comme la fonction de coût "Hinge" ou l'entropie croisée, qui est la fonction de coût utilisée pour la classification en pratique. Ces fonctions de coût sont illustrées sur la Fig. C.7a.



(a) Pour optimiser la fonction de coût 0/1 (en bleu), une fonction de coût substitutive est nécessaire, par exemple la fonction de coût logistique.*



(b) Les métriques d'évaluation de la recherche par images, par exemple AP, ne sont pas décomposables. La valeur AP moyenne estimée sur les lots bleus est de 0.78, tandis que les valeurs globales réelles en jaune sont de 0.68.

Figure C.7: La descente de gradient stochastique repose sur des fonctions de coût qui sont différentiables Fig. C.7a, et qui sont décomposables, ce qui n'est pas le cas pour AP Fig. C.7b.
*Image tirée de www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote10.html

Les systèmes de recherche par images sont évalués avec des métriques basées sur le classement, telles que la précision moyenne (AP), le rappel à k (R@k) ou le gain cumulatif décroissant normalisé (NDCG). Ces métriques sont utilisées car la recherche par images est une tâche fortement déséquilibrée, c'est-à-dire qu'il y a beaucoup plus de négatifs que de positifs. En effet, étant donné une image requête, la plupart des images de la base de données seront non pertinentes. Par exemple, sur la Fig. C.4b, lors de la requête de l'image de l'avion, la plupart des images dans la base de données ne sont pas un "McDonnell Douglas", et sont donc non pertinentes. Ces métriques sont basées sur la fonction de classement, qui peut être décrit avec la fonction d'Heaviside, comme cela sera détaillé dans la Sec. 2.2. Comme elles sont basées sur des fonctions d'Heaviside, ces métriques souffrent des mêmes problèmes que la fonction de coût 0/1 : elles ne sont pas différentiables, donc fine-tuner des modèles de recherche par images nécessite la conception de fonction de coût de substitution. Ce problème a été longuement étudié et a été abordé soit en utilisant des bornes supérieures grossières, par exemple la perte fonction de coût contrastive [64], par triplet [65], par proxy [66] ou en utilisant des approx-

imations du rang qui permettent une approximation fine des métriques cibles [67]–[70]. Par exemple, Coexya s'est appuyé sur un entraînement basé sur la classification qui ne correspond pas aux métriques d'évaluation de la recherche d'images, ce qui conduit à des performances sous-optimales. Concevoir des fonctions de coût de substitution qui approximent correctement les métriques d'évaluation, tout en conservant d'importantes propriétés de robustesse telles que les bornes supérieures, est un problème difficile.

En outre, ces métriques sont "list-wise", c'est-à-dire que la valeur de la métrique pour une requête dépend d'autres exemples. Par conséquent, elles ne sont pas séparables linéairement entre les exemples. Cela les rend "non décomposables". Leurs valeurs estimées sur un sous-ensemble ou des mini-batch de données sont biaisées. Cela est illustré sur la Fig. C.7b, où la moyenne de l'AP sur chaque batch (de la deuxième à la dernière ligne) est supérieure à l'AP global (première ligne). Comme mentionné précédemment, les DNN sont optimisés à l'aide de SGD, qui est utilisée en pratique pour des raisons à la fois computationnelles et de performance. D'autres fonctions de coût, par exemple l'entropie croisée, ne rencontrent pas ce problème et peuvent être estimées à l'aide de mini-batch de données. La non-décomposabilité est également un problème pour d'autres métriques, telles que le score de Dice [71], [72]. Bien que la non-décomposabilité soit un problème connu, elle a été moins étudiée que le problème de la non-différentiabilité. Les approches qui abordent la non-décomposabilité sont moins courantes et utilisent des méthodes *ad hoc* et brute force, par exemple en augmentant la taille des batchs au détriment de l'efficacité computationnelle dans [67] ou en stockant des batchs précédents [68], [73].

B.2 Défi 2 : Fragilité des sorties des DNN et gravité des erreurs.

Alors que les DNN sont très puissants pour représenter des images et effectuer des tâches spécifiques, ils peuvent être étonnamment fragiles face à différents facteurs et produisent des sorties instables. Un aspect notoire de leur fragilité et de leur instabilité est les *attaques adversaires* [74], où la sortie des DNN peut changer radicalement tandis que l'entrée change légèrement. Une autre instabilité est que les DNN ont peu de contrôle sur la gravité des erreurs qu'ils commettent en termes de compréhension humaine. Cela a été notamment observé dans [75], où il est montré que, tandis que les performances prédictives d'AlexNet [42] à ResNet-50 [45] ont évolué pour la classification sur ImageNet [43], la gravité des erreurs n'a pas diminué. Cela peut s'expliquer en partie par l'apprentissage par "raccourcis" des DNN [76]. En effet, les DNN tendent à apprendre des tâches en utilisant des raccourcis, par exemple en regardant l'arrière-plan dans la classification d'images plutôt que l'objet principal. Cela peut impliquer qu'au lieu

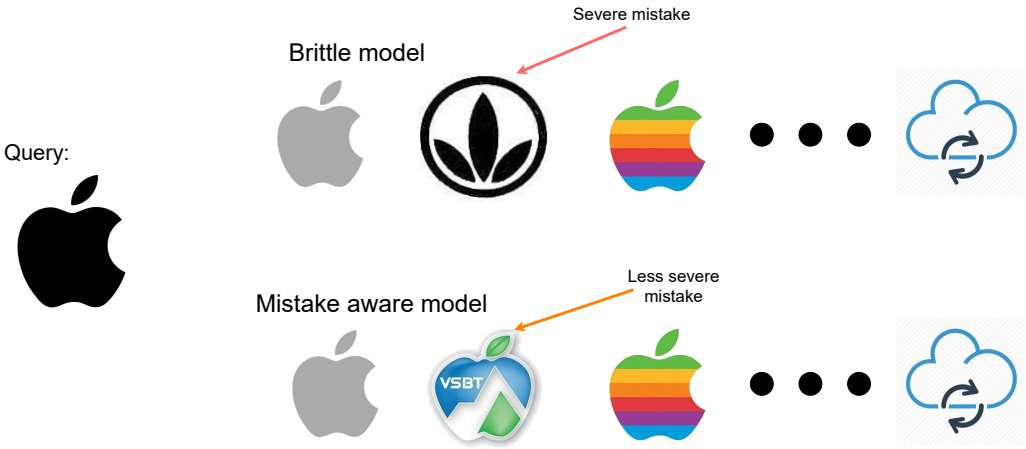


Figure C.8: Pour une requête, nous illustrons deux résultats. Le premier commet une erreur plus grave que le second.

d'apprendre une sémantique des images à reconnaître, ils s'appuient sur des caractéristiques non identifiables par les humains, ce qui peut conduire à des erreurs graves lorsqu'ils en commettent. De même, les systèmes CBIR peuvent présenter des cas d'échec, où ils commettent des erreurs graves lorsqu'ils récupèrent par erreur certains faux positifs. Cela est illustré sur la Fig. C.8, où, étant donné un logo "Apple", deux modèles peuvent commettre des erreurs plus ou moins graves. Contrairement aux features ingéniérées utilisées pour représenter des images, telles que SIFT [49], HOG [60] ou VLAD [77], les représentations des DNN manquent d'interprétabilité. Ce problème rend la compréhension de ces instabilités plus difficile à comprendre et à corriger en pratique.

La définition de la *gravité des erreurs* est difficile. Elle est liée aux préférences humaines et à leur compréhension des tâches. Comme les "préférences humaines"sont difficiles à définir en pratique, un domaine de recherche intéressant utilise les relations hiérarchiques entre les étiquettes d'images comme proxy. Pour le célèbre jeu de données ImageNet, la gravité des erreurs peut être déduite des relations hiérarchiques de la base de données syntaxique WordNet [78]. En recherche d'information, les chercheurs utilisent des "pertinences graduées"qui modélisent l'importance des instances récupérées pour une requête donnée. Par la suite, ils les utilisent dans des mesures graduées telles que le NDCG [79] ou une AP gradué [80]. De même, différents niveaux de pertinence peuvent être créés pour la recherche CBIR de logos de marque, comme illustré sur la Fig. C.9, où pour une requête donnée, les logos récupérés peuvent être plus ou moins pertinents. Cette tâche a également été abordée en utilisant des fonctions de coût de substitution hiérarchiques dans [81], [82]. Réduire la gravité des erreurs est important pour les moteurs de recherche, y compris le logiciel Acsepto de Coexya, afin de convaincre les utilisateurs

Figure C.9: Pour une requête donnée (*e.g.* le logo Apple), il existe plusieurs groupes plus ou moins pertinents : anciens logos Apple, logos représentant des pommes, des fruits, et enfin des logos n'ayant aucune similarité. Remarquez qu'Apple a engagé des poursuites pour des logos dans les groupes 2 et 3*.

* sources : www.huffpost.com/entry/apple-sues-woolworths-ove_n_309450
www.techspot.com/news/99131-apple-wants-trademark-images-apples.html
www.wired.com/2008/10/apple-takes-on/

de les utiliser et de leur faire confiance.

B.3 Défi 3 : Détection des échantillons hors distribution.

Bien que les performances prédictives des DNN aient augmenté, comme discuté précédemment, détecter quand les échantillons sont hors distribution (OOD) reste une tâche difficile. La tâche de détection d'OOD est une question difficile et a été longuement étudiée. La détection d'OOD est un autre défi important pour rendre les DNN plus robustes dans leurs prédictions. Cela donne la capacité de détecter s'ils doivent traiter une image ou non. Dans les applications critiques, il est important qu'un DNN sache quand il ne sait pas, par exemple pour des applications à la médecine, à la défense ou pour la conduite autonome, un système devrait rendre le contrôle à un décideur humain s'il ne sait pas quelle action prendre. La détection d'OOD est également une direction de recherche difficile pour la CBIR. C'est par exemple intéressant pour Coexya. En effet, un avantage industriel de Coexya est ses bases de données privées de logos de marque. Pour rester compétitif, une direction pourrait être de scraper le web ou de créer un sous-ensemble de très grands ensembles de données [31] pour constituer de nouveaux ensembles d'entraînement. Cela nécessite de pouvoir détecter si une image est un logo ou non. Cela est illustré sur Fig. C.10, un modèle a été entraîné sur l'ensemble d'entraînement, et doit détecter à l'inférence si une image est "in-distribution" (ID), *i.e.* elle provient de l'ensemble de

test, ou “out-of-distribution”, *i.e.* c’est une donnée OOD.

Figure C.10: Détection des logos de marque. Un modèle doit décider si les images sont “in-distribution” (ID), *e.g.* du jeu de données METU [83], ou hors distribution (OOD), *e.g.* du jeu de données ImageNet [43].

La difficulté de la détection OOD vient notamment de la surconfiance des modèles profonds. Cela a été identifié notamment dans [84], où les auteurs montrent comment les modèles profonds souffrent de surconfiance. Par exemple, en classification, cela signifie que les DNN donneront une forte probabilité à une mauvaise classe. Cela rend les approches naïves, telles que la probabilité softmax maximale (MSP) bien connue utilisée dans [85], échouent dans certains cas. En effet, MSP utilise la probabilité maximale d’un DNN comme mesure de confiance, ce qui n’est pas suffisant en pratique. Il y a eu plusieurs tentatives pour résoudre la détection d’OOD. Les auteurs de [86], [87] ont essayé de renforcer la détection d’OOD en intégrant des échantillons OOD dans l’ensemble d’entraînement. D’autres méthodes reposent sur des autoencodeurs pour accéder à une vraisemblance sur une image [88]. Les méthodes à l’état de l’art essaient d’estimer la densité de l’ensemble d’entraînement des DNN, par exemple [89] estime la densité ID en utilisant un modèle de mélange gaussien ou, plus récemment, les auteurs de [90] estiment la densité en utilisant la densité des k plus proches voisins. La disponibilité récente de gros modèles “sur l’étagère” qui ont de fortes performances prédictives a conduit à changement de paradigme dans la littérature sur la détection d’OOD. Ainsi, les méthodes récentes pour la détection d’OOD suivent le paradigme *post-hoc*, où elles exploitent des réseaux neuronaux pré-entraînés [89], [91], [92].

C Résumé et contributions.

Dans cette thèse, nous abordons plusieurs aspects de la robustesse des réseaux de neurones profonds. Plus précisément, nous introduisons une méthode pour l’optimisation robuste des métriques de classement utilisées dans la recherche par images, en abordant à la fois les prob-

lèmes de non-différentiabilité et de non-décomposabilité (Chapter 3). Nous montrons qu'en utilisant les relations hiérarchiques entre les étiquettes, nous pouvons entraîner des réseaux de neurones plus robustes par rapport à leurs erreurs dans Chapter 4. Nous examinons également la robustesse post-hoc des DNN en ce qui concerne leurs performances de détection hors distribution et comment les améliorer en utilisant des modèles basés énergie dans Chapter 5.

Plan. Nos contributions pour adresser les défis mentionnés ci-dessus sont les suivantes :

- Chapitre 3: Optimization of Ranking Losses for Image retrieval.

Dans ce chapitre, nous abordons les deux limitations de l'optimisation des métriques basées sur le classement identifiées dans le Défi 1 : la non-différentiabilité et la non-décomposabilité. Nous définissons un nouveau cadre d'entraînement qui aborde ces deux problèmes. Il utilise une approximation de la fonction de classement, SupRank, pour fournir des fonctions de coût de substitution lisses et qui sont des bornes supérieures. SupRank est une approximation précise du classement et présente des propriétés mathématiques solides ainsi que des performances expérimentales convaincantes. Nous montrons également les avantages théoriques que SupRank présente par rapport aux approximations lisses de [69], [70]. Nous optimisons une deuxième fonction de coût pendant l'entraînement, afin de garantir la décomposabilité des fonctions de coût basées sur le classement lors de l'entraînement par mini-batchs. Celle-ci entraîne un faible surcoût computationnel et rend l'optimisation du classement réalisable avec de petits batchs. Nous montrons dans une analyse théorique comment cet objectif supplémentaire aide à la décomposabilité durant l'optimisation de fonction de coût de classement. Ce cadre est général et peut être appliqué à de nombreuses fonctions de coût de classement. Dans ce premier chapitre, nous nous concentrons sur le cadre standard de recherche par images et appliquons ce cadre à deux métriques basées sur le classement : la précision moyenne (AP) et le rappel à k (R@k) pour optimiser DNN pour la recherche par images. Nous montrons lors de vastes validations expérimentales l'intérêt de notre cadre. Nous montrons d'abord qu'il se compare favorablement à des méthodes récentes de la littérature qui optimisent des métriques basées sur le classement. Nous montrons que notre cadre permet l'optimisation de métriques basées sur le classement avec de petits batchs. Nous montrons ensuite que notre cadre est robuste aux hyperparamètres. Enfin, nous comparons notre méthode à des méthodes de l'état de l'art et montrons qu'elle surpasse la concurrence sur plusieurs ensembles de données, de petites à grandes échelles, et validons notre méthode sur l'un des ensembles de données interne de Coexya.

- Chapitre 4: Hierarchical Image Retrieval for Robust Ranking.

Dans ce chapitre, nous remettons en question la définition de la similarité utilisée dans la recherche par images afin de traiter la fragilité des DNN par rapport à la gravité de leurs erreurs. Nous exposons les limitations de la similarité binaire communément utilisée dans la recherche par images en examinant la robustesse du modèle lorsqu'il commet des erreurs. Pour atténuer la gravité des erreurs, nous proposons d'utiliser des relations hiérarchiques entre les étiquettes pour définir une définition plus riche de la similarité entre deux images. Pour intégrer cette similarité lors de l'entraînement et de l'évaluation, nous introduisons une extension de la précision moyenne (AP), la précision moyenne hiérarchique ou $\mathcal{H}$-AP. Pour illustrer l'intérêt d'utiliser des relations hiérarchiques, nous optimisons deux métriques hiérarchiques différentes en utilisant le cadre de Chapter 3: $\mathcal{H}$-AP avec HAPPIER, et NDCG avec ROD-NDCG. L'utilisation de ce cadre nous permet d'avoir un entraînement plus robuste que les approximations utilisées en recherche d'informations [79], [80]. De plus, l'optimisation de métriques d'évaluation conduit à de meilleures performances que d'autres fonctions de coût de substituts utilisées en recherche par images hiérarchiques telles que [81], [82]. Nous discutons ensuite de l'hypothèse d'accès à des étiquettes hiérarchiques. Nous montrons comment annoter en pratique un ensemble de données de recherche par images avec des étiquettes hiérarchiques. Nous utilisons un pipeline semi-automatique pour étendre un jeu de données bien connu de recherche de landmarks, Google-Landmarks v2 [28] (GLDv2), avec des étiquettes hiérarchiques. Nous montrons dans une validation expérimentale qu'à la fois HAPPIER et ROD-NDCG i) sont au niveau des méthodes de l'état de l'art pour la recherche par images standard ii) surpassent largement les méthodes de recherche par images standard sur les métriques hiérarchiques, iii) surpassent d'autres méthodes hiérarchiques sur les métriques hiérarchiques et la recherche par images standard. Nos résultats sont valables pour six ensembles de données hiérarchiques et notre version hiérarchique de GLDv2. Nous montrons également l'intérêt de HAPPIER pour la recherche de logos sur deux ensembles de données internes de Coexya. Nous menons des études d'ablation de notre méthode pour montrer sa robustesse aux hyperparamètres. Enfin, nous montrons qualitativement que HAPPIER crée un espace d'embeddings mieux organisé que les méthodes non hiérarchiques, et montrons qualitativement la gravité moindre des erreurs de HAPPIER *vs.* des méthodes non hiérarchiques.

- Chapitre 5: Post-hoc out-of-distribution detection.

Dans ce chapitre, nous étudions un autre aspect de la robustesse des DNN: la détection hors distribution (OOD) post-hoc, comme décrit dans le Défi 3. Nous exploitons le cadre des modèles

basés énergie (EBM) [93] pour introduire une nouvelle méthode de détection d'OOD post-hoc : HEAT. HEAT est basé sur deux composantes : l'apprentissage résiduel et la composition des fonctions d'énergie. Nous utilisons d'abord les EBM pour apprendre une fonction résiduelle pour différentes méthodes de la littérature sur la détection d'OOD. En effet, plusieurs méthodes de la littérature sont basées sur l'approximation de la densité de l'ensemble de données d'entraînement, par exemple [91] utilise un modèle de mélange gaussien pour approximer la densité ID ou [94] utilise un score d'énergie dérivé des logits de sortie d'un DNN. Cependant, en raison de leurs forts biais de modélisation, ces méthodes manquent d'expressivité pour approximer correctement la distribution ID. Apprendre un terme résiduel avec un EBM permet plus d'expressivité. Un autre aspect des différents biais de modélisation de ces méthodes est qu'elles sont capables de détecter différents types d'échantillons OOD. Nous montrons qu'en utilisant la composition de fonctions d'énergie, nous sommes en mesure de combiner efficacement plusieurs types de scorers corrigés pour améliorer les performances globales de détection d'OOD. Enfin, HEAT est une méthode post-hoc, ce qui lui permet d'être utilisé sur pratiquement n'importe quel modèle profond disponible sur étagère. Nous nous concentrons sur la classification d'images car c'est un benchmark standard dans la littérature sur la détection d'OOD. Dans nos expériences, nous montrons comment les deux composantes de HEAT améliorent les performances de détection d'OOD. Nous comparons HEAT aux méthodes post-hoc de détection d'OOD de l'état de l'art sur deux benchmarks standard CIFAR-10 et CIFAR-100 et sur le grand jeu de données ImageNet. Nous montrons également que HEAT fonctionne avec plusieurs architectures, y compris les CNN et les Vision Transformers. Enfin, nous montrons que HEAT est robuste aux régimes de faibles données, et par rapport à ses hyperparamètres.

D Publications connexes.

Cette thèse est basée sur le contenu des articles suivants :

Publication	Chapitre
[95] **Elias Ramzi**, Nicolas Thome, Clément Rambour, Nicolas Audebert, and Xavier Bitot. "Robust and Decomposable Average Precision for Image Retrieval." Advances in Neural Information Processing Systems, 34$^{\text{th}}$ (NeurIPS, 2021). online: https://arxiv.org/abs/2110.01445	3
[96] **Elias Ramzi**, Nicolas Audebert, Nicolas Thome, Clément Rambour, and Xavier Bitot. "Hierarchical Average Precision Training for Pertinent Image Retrieval." in Proceedings of the 17$^{\text{th}}$ European Conference on Computer Vision (ECCV, 2022). online: https://arxiv.org/abs/2207.04873	4
[97] Marc Lafon, **Elias Ramzi**, Clément Rambour, Nicolas Thome. "Hybrid Energy Based Model in the Feature Space for Out-of-Distribution Detection." in Proceedings of the 40$^{\text{th}}$ International Conference on Machine Learning (ICML, 2023). online: https://arxiv.org/abs/2305.16966	5
[98] **Elias Ramzi**, Nicolas Audebert, Clément Rambour, André Araujo, Xavier Bitot and Nicolas Thome. "Optimization of Rank Losses for Image Retrieval." Under review, IEEE Transactions on Pattern Analysis and Machine Intelligence (under-review – TPAMI). online: https://arxiv.org/pdf/2309.08250.pdf	3,4

le cnam

Elias RAMZI

Robust image retrieval with deep learning

HESAM UNIVERSITÉ

Résumé : Cette thèse traite de la recherche par image robuste afin de rendre les systèmes de recherche d'images profonds performants et fiables. Nous commençons par discuter de l'écart entre leur entraînement et leur évaluation. Nous définissons ensuite une famille de fonction de coût, ROADMAP, qui sont plus étroitement alignées sur les métriques d'évaluation. Puis, nous étudions la gravité des erreurs commises par les systèmes de recherche d'images. Nous nous appuyons sur les relations hiérarchiques entre les catégories pour modéliser la gravité des erreurs. En se basant sur ROADMAP, nous optimisons une nouvelle extension de la précision moyenne pour le contexte hiérarchique, $\mathcal{H}$-AP avec HAPPIER. Nous montrons ensuite quantitativement et qualitativement que les modèles entraînés avec HAPPIER sont plus robustes et commettent des erreurs moins graves. Enfin, nous introduisons HEAT, une nouvelle méthode post-hoc pour la détection d'exemples hors distribution (OOD). HEAT est basée sur les modèles à énergies et s'appuie sur deux composants : un terme résiduel pour corriger des détecteurs OOD antérieurs et une composition des détecteurs corrigés afin d'exploiter leurs différents biais de modélisation.

Mots clés : Deep learning, Computer vision, Image retrieval, Robustness.

Abstract : In this thesis, we discuss robust image retrieval, to make deep image retrieval systems both performant and reliable. We first address the discrepancy between the training objective of image retrieval systems and how they are evaluated. We design a new training framework, ROADMAP, that more closely aligns with the evaluation metrics. Then, we investigate the severity of mistakes that image retrieval systems make. We rely on hierarchical relations between categories to model mistake severity. Using the ROADMAP framework, we optimize a novel extension of average precision to the hierarchical setting, $\mathcal{H}$-AP with HAPPIER. We show quantitatively and qualitatively that training with HAPPIER leads to more robust models with less severe mistakes. Finally, we address out-of-distribution (OOD) detection, a task where models must recognize if inputs relate to what they were trained for. We introduce a new post-hoc method, HEAT, that is based on energy-based models. It has two components: a residual term to correct prior OOD detectors, and a composition of corrected priors to leverage their different modeling biases.

Keywords: Deep learning, Computer vision, Image retrieval, Robustness.