On depth prediction for autonomous driving using self-supervised learning

Houssem Eddine Boulahbal

UNIVERSITÉ CÔTE D'AZUR

ÉCOLE DOCTORALE
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THÈSE DE DOCTORAT

Prédiction de la profondeur pour la conduite autonome à l'aide de l'apprentissage auto-supervisé

Houssem eddine BOULAHBAL

Renault Software Factory et Laboratoire d'Informatique, de Signaux et Systèmes de Sophia Antipolis (I3S), Université Côte d'Azur, CNRS

Présentée en vue de l'obtention du grade de docteur en Automatique et Traitement du Signal et des Images **d'**Université Côte d'Azur

Dirigée par : Andrew I. COMPORT, Tenured senior researcher, Université Côte d'Azur, I3S CNRS, France

Co-encadrée par : Adrian VOICILA, PhD research engineer, Renault software factory, France

Soutenue le : 29/09/2023

Devant le jury, composé de :

Tarek HAMEL, Professor, Université Côte d'Azur, I3S CNRS, France

Tom DRUMMOND, Professor, Monash University, Australia

Walterio W. MAYOL-CUEVAS, Professor, University of Bristol and Amazon, United Kingdom

Anne SPALANZANI, Professor, Université Grenoble Alpes, Inria, France

PRÉDICTION DE LA PROFONDEUR POUR LA CONDUITE AUTONOME À L'AIDE DE L'APPRENTISSAGE AUTO-SUPERVISÉ

On depth prediction for autonomous driving using self-supervised learning

Houssem eddine BOULAHBAL

Jury :

Président du jury

Tarek HAMEL, Professor, Université Côte d'Azur, I3S CNRS, France

Rapporteurs

Tom DRUMMOND, Professor, Monash University, Australia
Walterio W. MAYOL-CUEVAS, Professor, University of Bristol and Amazon, United Kingdom

Examinateurs

Anne SPALANZANI, Professor, Université Grenoble Alpes, Inria, France

Directeur de thèse

Andrew I. COMPORT, Tenured senior researcher, Université Côte d'Azur, I3S CNRS, France

Co-encadrant de thèse

Adrian VOICILA, PhD research engineer, Renault software factory, France

---

Université Côte d'Azur

الْحَمْدُ لِلَّهِ الَّذِي هَدَانَا لِهَذا وَمَا كُنَّا لِنَهْتَدِيَ لَوْلَا أَنْ هَدَانَا اللَّه

To my beloved family - my mother, father, wife, brothers and sister.

Résumé

La perception de l'environnement est un élément essentiel de la conduite autonome. Elle permet au véhicule de comprendre son environnement et de prendre des décisions informées. La prédiction de la profondeur joue un rôle central dans ce processus, car elle aide à comprendre la géométrie et le mouvement de l'environnement. Cette thèse se concentre sur le défi de la prédiction de la profondeur en utilisant des techniques d'apprentissage auto-supervisé en utilisant des cameras monoculaire. En premier lieu, le problème est abordé d'un point de vue plus large, en explorant les réseaux adversaires génératifs conditionnels (cGAN) en tant que technique potentielle pour obtenir une meilleure généralisation. Ce faisant, une contribution fondamentale aux GAN conditionnels, le cGAN ac, a été proposée. La deuxième contribution concerne une méthode auto-supervisée pour translater une image à une carte de profondeur, en proposant une solution pour les scènes rigides à l'aide d'une nouvelle méthode basée sur les transformeurs qui génère une pose pour chaque objet dynamique. Le troisième aspect important concerne l'introduction d'une approche de prévision du future de carte profondeur en utilisant la vidéo. Cette méthode sert d'extension aux techniques auto-supervisées pour prédire les profondeurs futures. Elle implique la création d'un nouveau modèle de transformateur capable de prédire la profondeur future d'une scène donnée. En outre, les diverses limitations des méthodes précédemment mentionnées ont été abordées et un modèle de cartes de profondeur vidéo-vidéo a été proposé. Ce modèle tire parti de la cohérence spatio-temporelle de la séquence d'entrée et de la séquence de sortie pour prédire une séquence de profondeur plus précise. Ces méthodes ont des applications significatives dans la conduite autonome et les systèmes avancés d'aide à la conduite. L'approche est auto-supervisée, ce qui élimine le besoin de labellisation manuelle des cartes de profondeur pendant la phase d'apprentissage, la rendant ainsi efficace et rentable. Dans l'ensemble, cette thèse apporte plusieurs contributions au domaine de la conduite autonome en développant une approche auto-supervisée de la prédiction de la profondeur. L'approche proposée est efficace, avec le potentiel d'améliorer la sécurité et la fiabilité des systèmes de conduite autonome. Les implications de ces résultats sont importantes pour la conception de systèmes avancés d'aide à la conduite et de véhicules autonomes, ce qui nous rapproche de l'objectif d'une conduite entièrement autonome.

Keywords Profondeur, apprentissage profond, auto-supervisé, prédiction, conduite autonome

Abstract

Perception of the environment is a critical component for enabling autonomous driving. It provides the vehicle with the ability to comprehend its surroundings and make informed decisions. Depth prediction plays a pivotal role in this process, as it helps the understanding of the geometry and motion of the environment. This thesis focuses on the challenge of depth prediction using monocular self-supervised learning techniques. The problem is approached from a broader perspective first, exploring conditional generative adversarial networks (cGANs) as a potential technique to achieve better generalization was performed. In doing so, a fundamental contribution to the conditional GANs, the *a contrario* cGAN was proposed. The second contribution entails a single image-to-depth self-supervised method, proposing a solution for the rigid-scene assumption using a novel transformer-based method that outputs a pose for each dynamic object. The third significant aspect involves the introduction of a video-to-depth map forecasting approach. This method serves as an extension of self-supervised techniques to predict future depths. This involves the creation of a novel transformer model capable of predicting the future depth of a given scene. Moreover, the various limitations of the aforementioned methods were addressed and a video-to-video depth maps model was proposed. This model leverages the spatio-temporal consistency of the input and output sequence to predict a more accurate depth sequence output. These methods have significant applications in autonomous driving (AD) and advanced driver assistance systems (ADAS). The approach is self-supervised, which eliminates the need for manual labeling of depth maps during training, making it efficient and cost-effective. Overall, this thesis makes several contributions to the field of autonomous driving by developing a self-supervised approach to depth prediction. The proposed approach is effective and efficient, with the potential to enhance the safety and reliability of autonomous driving systems. The implications of the findings are important for the design of advanced driver assistance systems and autonomous vehicles, bringing us one step closer to achieving the goal of fully autonomous driving.

Keywords Depth, self-supervision, prediction, autonomous driving, deep learning

Acknowledgments

I would like to sincerely express my gratitude to everyone who has supported me throughout this incredible journey. Firstly, I am deeply thankful to God for His guidance and blessings, as I wouldn't have reached this point without His steadfast support.

I am immensely thankful to my supervisor, Andrew Comport, for granting me a unique and exceptional opportunity. His invaluable feedback and guidance have played a pivotal role in shaping my thesis and fostering my growth as a researcher. Working under his supervision in such an exceptional environment has been truly enriching. I would also like to extend my heartfelt gratitude to Adrian Voicila for his supervision and support, which were crucial in successfully completing this work. Furthermore, I want to extend my deepest appreciation to Tarek Hamel, whose support was instrumental in the launch of this thesis. I am genuinely grateful for his dedication.

I am profoundly grateful to my friends and colleagues. Your camaraderie and positivity made this time truly extraordinary. I extend my sincere thanks to my family, both immediate and extended, for their unwavering support, unending encouragement, and constant presence in my life. Your support has been a wellspring of strength throughout this journey. I am endlessly grateful for each of you. Finally, to all who contributed, in ways big or small, your support has been instrumental in propelling me forward.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Autonomous driving (AD) is one of the most complex research and engineering challenges. It refers to the ability of a vehicle to operate without human intervention. It is driving innovation for computer vision and mobile robotics. The potential benefits of AD are immense, including increased safety and comfort, reduced traffic congestion, and improved mobility for people with disabilities. The challenge of AD is multifaceted, and it requires the development of systems that can enable a vehicle to **perceive** its environment (this refers to a vehicle's ability to accurately interpret its surroundings, including identifying the geometry of the scene, vehicles, pedestrians ... etc.), **make decisions**, and take appropriate **actions** as shown in Fig. 1.1. These tasks are not trivial, and they require the integration of various disciplines, such as computer vision, machine learning, mobile robotics, control theory, and others. Moreover, the development of AD systems requires addressing a wide range of issues, including legal and ethical considerations, societal impacts, and economic feasibility. Solving this challenge is a huge step towards developing general intelligence.

Recent years have seen tremendous progress on deep learning (DL), especially in computer vision, with impressive results on several tasks such as classification [61, 156, 40, 107], detection [60, 144, 142, 100, 90, 36], segmentation [5, 105, 23, 84] and depth prediction [147, 92, 167, 140, 54, 51, 189, 50, 75, 141]. One particular advantage of deep learning is the ability to extract features directly from data without the need for handcrafted expert systems. Deep learning methods have shown the potential to extrapolate into unseen situations that are not present in the training set for large scale datasets. This is practical, as one aspect of developing intelligent agents is the adaptability to new cases. Advances made in deep learning have motivated the research community to pursue and rely on deep learning methods as a dominant solution for addressing perception of the environment and providing these systems with intelligence.

The current endeavor to resolve the problem of general intelligence is to create systems inspired

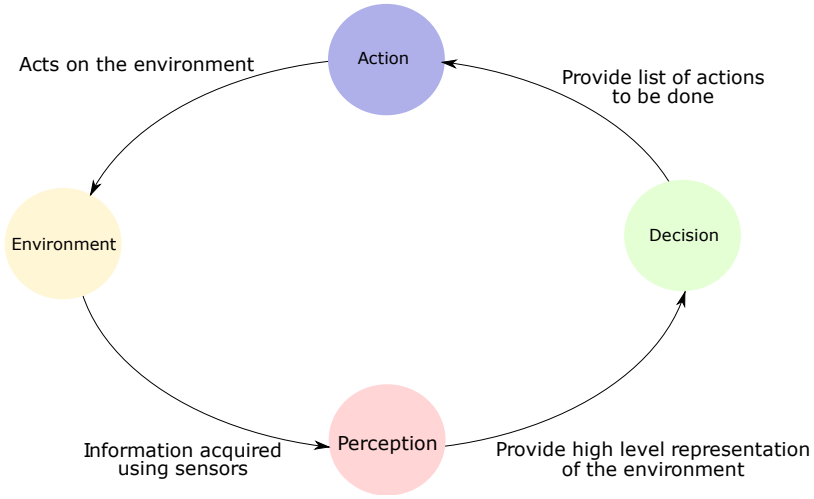


Figure 1.1: Illustration of steps of the cycle taken by an autonomous agent

from humans. Our intelligence includes the ability to learn, memorize and adapt to new situations. When we are starting to learn driving, we first learn the possible actions we could perform on the car and how these actions change the perceived environment. However, becoming an effective driver also requires developing a "specialized driving perception". An experienced driver has learned to identify all the pertinent agents in the scene, track and predict their behavior, asses the risk of each possible action and take the right action. The driver has **learned** a specialized driving perception that was based on his prior perception "model" and "fine-tuned" it to the new challenges of driving. During "inference" the driver applies all the aforementioned abilities intuitively. A key component of this driving perception is the ability to anticipate future states of the environment. Similarly, the development of an autonomous driving system requires the creation of an intelligent system that can anticipate future states of the environment. This will provide the vehicle with the ability to comprehend its surroundings and make well-informed decisions.

To enable an autonomous vehicle to perceive its surroundings, a range of sensors are installed, with the camera being a primary sensory modality. The visual information is central to understanding the environment. However, the RGB representation captured from cameras is not pertinent to AD applications as it does not explicitly provide the information about the geometry, the entities present in the scene and its motion. Depth and semantics extract pertinent information from the high-dimension information present in the RGB representation, thus provide an alternative representation of the state of the scene that is suitable for making decisions. Depth information can provide an estimation of the distance between objects in the scene and enable the creation of a 3D representation of the environment. Moreover, semantic information can provide a high-level abstract representation of the entities present in the scene, such as road boundaries, traffic signals, and other vehicles. Therefore, depth information plays a pivotal role in providing a more comprehensive

representation of the environment, enabling more effective decision-making for autonomous vehicles.

In the community, there has been an ongoing and in-depth discussion regarding the use of cameras versus LiDAR technology as the primary source of depth perception. While LiDAR is undoubtedly a powerful technology for depth perception, it has several advantages, including real-time and high accurate depth sensing at the sensor level. However, the cost of implementing it in automotive applications is a significant drawback. In contrast, cameras have emerged as a more cost-effective and practical solution for achieving accurate depth perception, making them a popular choice for many automotive manufacturers. In addition to their affordability and versatility, cameras also offer significant advantages in terms of power consumption and computing resources. Compared to LiDAR, cameras require less power to operate and fewer computing resources to process data. Moreover, the advances in deep learning over recent years have significantly improved the capabilities of camera-based-depth perception. One promising avenue of research in this area is the use of self-supervised learning approaches for depth prediction using cameras. These approaches circumvent the need for huge labeled datasets that are expensive and laborious to collect. By leveraging the vast amounts of unlabeled data available in the form of videos, it is possible to train deep learning models to predict depth without the need for explicit supervision or ground truth data.

To clarify ambiguous terminology found across different papers in the literature, the term "prediction" will be considered here to encompass both "depth inference" and "future forecasting". The term "inference" will be reserved for the prediction at time t (*i. e.* the mapping $\mathbf{D}_t = f(\mathbf{I}_t; \boldsymbol{\theta})$ where $\mathbf{D}_t$ is the depth map at time t, $\mathbf{I}_t$ is the image at time t and $\boldsymbol{\theta}$ are the model's parameters) and "forecasting" will be used for future predictions ($t+1 : t+k$) (*i. e.* the mapping $\mathbf{D}_{t+1:t+k} = f(\mathbf{I}_{t-n:t}; \boldsymbol{\theta})$ such as k is the future horizon and n is the past context).

1.2 Objective

This thesis aims to investigate the use of depth maps as a representation of a scene, with a focus on developing self-supervised deep learning models for monocular depth prediction. The ultimate goal is to solve the prediction of the future depth as a way to bring intelligence into the systems As discussed in Sec. 1.1.

Depth maps provide a rich representation of a scene, allowing the understanding of motion and geometry. By utilizing deep learning, It is possible to develop models that are highly accurate and can generalize to new scenarios, allowing for the development of more effective and efficient intelligent systems. However, this requires huge and annotated datasets, which is impractical. Therefore, self-supervision approaches will be leveraged to train these models, allowing to exploit the vast amounts of unlabeled data available, and providing for the development of models that are both accurate and efficient. In order to enable a wide range of applications, the explored models in this thesis are predominantly monocular, which is the most challenging setting, since recovering 3D from 2D

images is an ill-posed problem. However, this will enable a wide range of applications.

Predicting the future depth of a scene is a challenging task, but has significant implications for intelligent systems. By accurately predicting future depth, systems can better anticipate and react to changes in their environment, allowing for more efficient and effective operation.

1.3 Contribution

This thesis focuses on the challenge of depth prediction using self-supervised learning techniques. several key contributions were made. Firstly, the problem is approached from a broader perspective, exploring conditional generative adversarial networks (cGANs) as a potential technique to achieve better generalization was performed. In doing so, a fundamental contribution to the conditional GANs, the *a contrario* cGAN was proposed. The second significant contribution involves the development of a self-supervised method for single image-to-depth inference. This method proposes a solution to overcome the rigid-scene assumption of the classical SfM model by utilizing a novel transformer-based approach that outputs a pose for each dynamic object. The third contribution revolves around the proposal of a video-to-depth forecasting approach. This includes the development of a novel transformer model capable of forecasting the future depth of a scene, thereby extending the application of self-supervised methods to predict future depths. Finally, the various limitations of the aforementioned method were addressed and a video-to-video depth prediction model was proposed. This model leverages the spatio-temporal consistency of the input and output sequence to predict a more accurate depth sequence output. These methods have significant applications in autonomous driving (AD) and advanced driver assistance systems (ADAS). Our approach is self-supervised, which eliminates the need for manual labeling of depth maps during training, making it efficient and cost-effective.

1.4 Outline

The organization of this thesis is as follows: Chapter 2 provides an overview of the background information relevant to the thesis. This includes a review of machine learning basics, and a literature review on deep learning depth methods. Additionally, a brief literature review is presented that explores the use style transfer (will be used in Chapter 3)of semantic information (will be used in Chapter 3 and Chapter 4) and the Transformer module as building block for the proposed architectures.

Chapter 3 The problem is approached with a wider perspective. The generalization of deep learning models was explored. This was done through domain adaptation methods, by utilizing generative adversarial networks, specifically conditional GANs for style transfer. During this exploration, a fundamental limitation of cGANs is revealed, namely their lack of complete conditionality. To address

this issue, the chapter presents an innovative solution referred to as the "*a contrario* method". The main objective of the *a contrario* method is to enhance conditional GANs and empower them with full conditionality.

Chapter 4 explores single-image-to-depth inference and extends the classical self-supervised methods with dynamic objects. Before aiming to forecast the future, it is essential at first to understand the present. In this chapter, we propose a solution for the static-scene assumption of the classical SfM model, using a novel transformer-based method that outputs a pose for each dynamic object. This model is a single image-to-depth mapping. The depth model takes a single image as input and outputs the corresponding depth map.

Chapter 5 explores video-to-depth forecasting. This was a first attempt to forecast the future depth using self-supervision. The input to this model is a sequence of present and past images, and the model output a depth map that represents the future depth at step k. A novel transformer-based architecture is proposed to aggregate the temporal information, this enables the network to learn a rich spatio-temporal representation.

Chapter 6 explores a video-to-video depth model. This model takes a sequence of images of past and present images and outputs a sequence of the present and future depth maps. This method addresses the limitations of the previous methods and extends the forecasting into a sequence of future depth. A self-supervised model that simultaneously predicts a sequence of future frames from video input with a novel spatial-temporal attention (ST) network is presented.

Finally, in Chapter 7, summarizes the main contributions and presents perspectives for future work.

List of publication

The work presented in this thesis led to the following publications:

International publications :

- **Journal paper:** Boulahbal Houssem Eddine, Adrian Voicila, and Andrew I. Comport. "Instance-aware multi-object self-supervision for monocular depth prediction." IEEE Robotics and Automation Letters 7.4 (2022): 10962-10968.
- **Conference paper:** Boulahbal Houssem Eddine, Adrian Voicila, and Andrew I. Comport. "Are conditional GANs explicitly conditional?." British Machine Vision Conference. 2021.
- **Conference paper:** Boulahbal Houssem Eddine, Adrian Voicila, and Andrew I. Comport. "Instance-aware multi-object self-supervision for monocular depth prediction." 2022 35th International Conference on Intelligent Robots and Systems (IROS). IEEE, 2022.
- **Conference paper:** Boulahbal Houssem Eddine, Adrian Voicila, and Andrew I. Comport. "Forecasting of depth and ego-motion with transformers and self-supervision." 2022 26th International Conference on Pattern Recognition (ICPR). IEEE, 2022.
- **Conference paper:** Boulahbal Houssem Eddine, Adrian Voicila, and Andrew Comport. "STDepthFormer: Predicting Spatio-temporal Depth from Video with a Self-supervised Transformer Model." arXiv preprint arXiv:2303.01196 (2023). **To be submitted**

National publication :

- **Conference video poster:** Boulahbal, Houssem Eddine, Adrian Voicila, and Andrew I. Comport. "Un apprentissage de bout-en-bout d'adaptateur de domaine avec des réseaux antagonistes génératifs de cycles consistants." Journée des Jeunes Chercheurs en Robotique. 2020.

Chapter 2

Background

2.1 Machine learning basics

Learning is the process of acquiring new knowledge or skills through experience or study. Machine learning attempts to create algorithms that can gain knowledge from and make decisions based on data. Primarily, machine learning involves designing models that are capable of extracting knowledge or insights from data employing the learning procedure.

According to Mitchell [119] "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." In the context of deep learning, the computer program is a model $\hat{\mathbf{y}} = f(\mathbf{x}; \boldsymbol{\theta})$ defined by its parameters $\boldsymbol{\theta}$. $\mathbf{x}$ is the input variable and $\hat{\mathbf{y}}$ is the output of the model. The experience, or dataset, is a collection of examples $\mathbf{D} = (\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), ..., (\mathbf{x}_N, \mathbf{y}_N)$ where each example is a pair of input $\mathbf{x}$ and output $\mathbf{y}$. The task varies depending on the user's need, *i. e.* classification, object detection ... etc. The performance measure P assesses how the model performs on the given task.

For deep learning, the back-propagation algorithm is commonly used to learn these parameters through optimization with gradient descent. More often, the performance metric P is not differentiable and cannot be used with the back-propagation framework. In this case, a surrogate loss function is used instead. It acts as a proxy for the performance metric. By minimizing the loss function, the performance measure is improved. For example, the cross-entropy loss is the proxy loss for classification precision, and minimizing it leads to improved classification precision. Learning can be defined more formally as:

$$\mathbf{R}(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{data}(\mathbf{x},\mathbf{y})} L(f(\mathbf{x}; \boldsymbol{\theta}), \mathbf{y}) \tag{2.1}$$

Learning involves finding the set of parameters $\boldsymbol{\theta}$ that minimizes the expected value of the loss

function L across the data generating distribution $p_{data}(\mathbf{x}, \mathbf{y})$. The quantity $\mathbf{R}(\boldsymbol{\theta})$ is known as the risk. In practice, however, this quantity cannot be optimized as $p_{data}(\mathbf{x}, \mathbf{y})$ is not known. Instead, the empirical distribution represented by the training set $\mathbf{D}_{train}$ is used and Eq. 2.1 becomes:

$$\mathbf{r}(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \sum_{D_{train}} L(f(\mathbf{x}; \boldsymbol{\theta}), \mathbf{y}) \tag{2.2}$$

This quantity is known as the **empirical risk** it is also known as the **generalization error**.

2.1.1 Generalization

Generalization is achieved when the model is capable of learning representative and abstract features that capture complex relationships and patterns in the data. In order to assess the model for generalization, Eq. 2.2 is also calculated for the validation dataset $\mathbf{D}_{val}$:

$$\mathbf{r}(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \sum_{D_{val}} L(f(\mathbf{x}; \boldsymbol{\theta}), \mathbf{y}) \tag{2.3}$$

A model reaches good generalization when the training error Eq. 2.1 is very low and the gap between the validation Eq. 2.3 and training errors Eq. 2.1 is very low. Poor generalization can be divided into two categories:

- **Over-fitting :** If the gap between the training and validation generalization error is big, this indicates that the model has memorized the training set, and it is not able to generalize for unseen data. This happens when the model is too complex and has learned patterns and relationships that are specific to the training data
- **Under-fitting:** This is characterized when the training error and validation error are poor, the validation error may even be lower than the training error. This happens when the model is too simple, and it is not capable of learning a meaningful and useful feature.

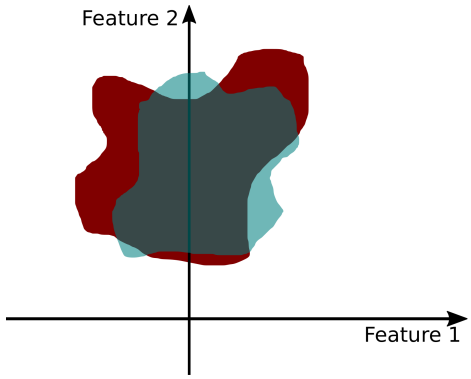
Generalization could be improved using several techniques including regularization, increasing the dataset size, and using ensemble methods.

2.1.2 Representation learning

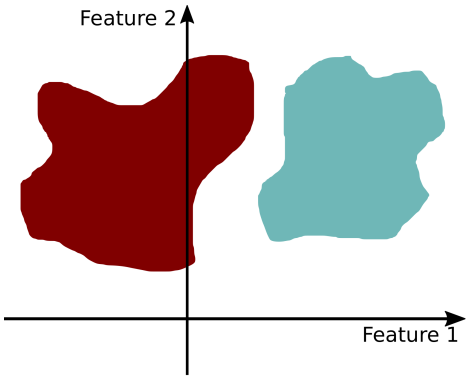
The complexity of the data can sometimes make the learning process challenging. For example, an RGB image with a resolution of 1024×512 and 8-bit representation per pixel contains a total of $(256 \times 256 \times 256)^{1024 \times 512}$ possible images. Modeling the distribution of this high-dimensional space is intractable.

Despite the fact that images are typically represented as high-dimensional data, it has been observed that natural images actually reside on a low-dimensional manifold. In other words, the set

of all natural images exists in a lower-dimensional subspace of the high-dimensional space. One of the key insights behind the manifold hypothesis is that natural images exhibit a high degree of structure and regularity, which can be captured by low-dimensional representations. This assumption, which is central to machine learning, enables the identification of a meaningful representation of the data. The goal of learning is to find a good representation of this manifold. Good generalization occurs when the model is able to learn a good representation that not only accurately represents the training examples *i. e.* interpolation, but can also accurately predict the behavior of examples that were not seen during training *i. e.* extrapolation. Fig. 2.1 illustrates how the data can be mapped onto two different manifolds. A good representation allows us to leverage it for the chosen tasks. For example, the representation Fig. 2.1b is better than Fig. 2.1a for classification, as it is easier to define the decision boundary that separates the two domains.



(a) This representation is not suitable for classification applications



(b) This representation is better suitable for classification applications as it shows clearly the decision boundary

Figure 2.1: An example of a possible mapping of a dataset that contains two classes using two features

Learning algorithms can be classified into different categories based on the type of supervision they require. Supposing $\hat{\mathbf{y}} = f(\mathbf{x}; \boldsymbol{\theta})$ we can make the following definitions :

- **Supervised learning:** In this setting the ground-truth $\mathbf{y}$ is known, and it is leveraged when calculating the loss function. The model is trained on a labeled ground-truth dataset, and it makes predictions based on this input/output mapping. Some examples of supervised learning tasks include classification: Predicting which category a new example belongs to (e.g., spam or not spam); and regression: Predicting a continuous value (e.g., the depth of an object in the scene).
- **Unsupervised learning** The model does not have access to the ground-truth labels. the model is trained to discover the underlying pattern of the data. Some examples of unsupervised learning tasks include image generation, clustering, and domain adaptation.

- **Self-supervised learning** is a special class of unsupervised learning where the model is trained against a proxy task, and when improving the proxy the downstream task is also improved. An example of self-supervision is self-supervised depth prediction. the proxy task is the image reconstruction and the depth is the downstream task. Training this model for image reconstruction provides a sufficient gradient signal to supervise the depth.
- **Semi-supervised** These models leverage both supervised and unsupervised techniques. When working with partially labeled datasets, it can be beneficial to incorporate both supervised and unsupervised techniques in order to make the most of the available data, especially when the labeling cost is high.

Based on the applications, two types of models could be distinguish : generative models and predictive models. For a dataset $p_{data}(\mathbf{x}, \mathbf{y})$, the generative models try to model the distribution of the data $p(\mathbf{x})$. Predictive models on the other hand model the distribution of the output y as $p(\mathbf{y}|\mathbf{x})$ usually y have small dimensions such as a class label.

2.1.3 Generative models

Generative models aim to model the distribution of $p(\mathbf{x})$ or a conditional distribution $p(\mathbf{x}|\mathbf{y})$ such as $\mathbf{y}$ is the label of $\mathbf{x}$. By learning the distribution of the dataset, it is possible to generate new samples $\mathbf{x}_{sythetic} \sim p(\mathbf{x})$. Application of these models includes image generation, text generation, text-to-images, image-to-image translation, domain adaptation. The most used generative models are: Variational auto-encoders (VAEs), Generative adversarial networks GANs, and Diffusion models.

Variational auto-encoders

An autonecoder is a neural network that is used to reconstruct its input. It consists of an encoder that models $p(\mathbf{h}|\mathbf{x})$ and a decoder that models $p(\mathbf{x}_{rec}|\mathbf{h})$. AE can be used as a generative model by leveraging the decoder. an auto-encoder defines a generative model of the form:

$$p_{model}(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{h})\, p(\mathbf{h}) dh \tag{2.4}$$

Where h is the latent variable. If the autoencoder is trained as a generative model, it is optimized to maximize the likelihood of $p_{model}(\mathbf{x})$ with respect to $p_{data}(\mathbf{x})$. However, the exact inference of Eq. 2.4 is intractable. It is possible to approximate this quantity using the evidence lower bound

defined as follows:

$$p_{model}(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{h})p(\mathbf{h})dh \tag{2.5}$$

$$log[p_{model}(\mathbf{x})] = log\big[\int p(\mathbf{x}|\mathbf{h})p(\mathbf{h})\frac{q(\mathbf{h}|\mathbf{x})}{q(\mathbf{h}|\mathbf{x})}dh\big] \tag{2.6}$$

$$log[p_{model}(\mathbf{x})] = log\big[E_{q(\mathbf{h}|\mathbf{x})}\big[\frac{p(\mathbf{x}|\mathbf{h})p(\mathbf{h})}{q(\mathbf{h}|\mathbf{x})}\big] \tag{2.7}$$

Using Jansen inequality : $E[log(z)] \geq logE[z]$:

$$log[p_{model}(\mathbf{x})] \geq E_{q(\mathbf{h}|\mathbf{x})}\Big[log\big[\frac{p(\mathbf{x}|\mathbf{h})p(\mathbf{h})}{q(\mathbf{h}|\mathbf{x})}\big]\Big] = E_{q(\mathbf{h}|\mathbf{x})}p(\mathbf{x}|\mathbf{h}) + E_{q(\mathbf{h}|\mathbf{x})}\frac{p(\mathbf{h})}{q(\mathbf{h}|\mathbf{x})} \tag{2.8}$$

Further simplifying with the KL-divergence :

$$log[p_{model}(\mathbf{x})] \geq E_{q(\mathbf{h}|\mathbf{x})}p(\mathbf{x}|\mathbf{h}) - D_{KL}(q(h|x) \parallel q(h)) \tag{2.9}$$

maximizing Eq. 2.9 is equivalent to maximizing the likelihood of the decoder's output and minimizing the distance of the distribution $q(\mathbf{h}|\mathbf{x})$ and $p(\mathbf{h})$. q is chosen to be a Gaussian distribution. This second term makes the approximate posterior distribution $q(\mathbf{h}|\mathbf{x})$ and the model prior p(h) approach each other.

The VAE model is both elegant, theoretically pleasing, and simple to implement. It also achieves good results and is among the leading approaches in generative modeling. However, when used in image generation, the output of the model tends to be blurry. One possibility is that the blurring is an intrinsic effect of maximum likelihood. The denoising VAE provides the base for other models, such as Diffusion models.

Generative adversarial networks (GANs)

Generative Adversarial Networks (GANs) [52] have introduced an alternative framework for training generative models that have led to a multitude of publications with high impact over a very large number of applications. The training of these models involves two networks that compete against each other. A generator that models the distribution $p(\mathbf{x})$. It tries to fool the discriminator by generating samples that are as close as possible to real samples. The discriminator models $p(real|\mathbf{x})$. The discriminator tries to classify the real and synthetic samples. This is formulated as a zero-sum game between two networks G and D competing to reach a Nash equilibrium. This game is commonly formulated through a min-max optimization problem as follows :

$$\min_{G\in\mathbb{G}} \max_{D\in\mathbb{D}} V(G, D) \tag{2.10}$$

The function V determines the payoff of the discriminator. The discriminator receives $-V(G, D)$ as its own payoff, and the generator receives $V(G, D)$ as its own payoff. In other words, this formulation could be interpreted as a learned loss function as the discriminator provide the supervision signal to update the generator. A common way to supervise the discriminator is based on the cross entropy :

$$\mathcal{L}_{adv} = \min_{G} \max_{D} \mathbb{E}_{(\mathbf{x})\sim p_{data}(\mathbf{x})} log[D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_{(}\mathbf{z})} log[1 - D(G(\mathbf{z}))] \tag{2.11}$$

At convergence, the generator's samples are indistinguishable from real data, and the discriminator outputs $\frac{1}{2}$ everywhere. Conditional GANs [118], introduced shortly after, have extended GANs to incorporate conditional information as input and have demonstrated resounding success for many computer vision tasks such as image synthesis [71, 130, 170, 24, 155, 159, 106], video synthesis [169, 18, 102], image correction[89, 184, 135], text-to-image[143, 185, 178, 96]. In all these works, the underlying GAN model as proposed in [53] and [118] have formed the basis for more advanced architectures and their properties have been analysed in detail and established in terms of convergence[86, 125], mode collapse[152], Nash equilibrium[161, 45], vanishing gradients[2]

Conditional GANs

A GAN is considered conditional [118] when the generator's output is conditioned by an extra input variable $\mathbf{y}$ such that $G(\mathbf{y}) \approx= p(\mathbf{x}|\mathbf{y})$ and discriminator's output is conditioned such that $D(\mathbf{x}, \mathbf{y}) \approx p(\mathbf{z}|\mathbf{x}, \mathbf{y})$ where $\mathbf{z}$ is the probability of the input being true or generated given $(\mathbf{x}, \mathbf{y})$. The condition variable can be any kind of information such as a segmentation mask, depth map, image, or data from other modalities. In the literature there are various methods that have been proposed for incorporating conditional information into the generator including the introduction of new modules: Conditional Batch Normalization (CBN) [164], Conditional Instance Normalization (CIN) [42], Class Modulated Convolution (CMConv) [188], Adaptive Instance Normalization (AdaIN) [68], Spatial Adaptive Normalization (SPADE) [130]. Recently, [159] introduced a classification-based feature learning module to learn more discriminating and class-specific features. Additional generator losses have also been proposed including feature matching [148], perceptual loss [74], and cycle-consistency loss [190]. All these methods propose approaches that improve the conditionality of the generator, however, they do not act on making the discriminator conditional.

Alternatively, several methods have been proposed which investigate how to incorporate conditional information into the discriminator of adversarial networks. [118] proposed an early fusion approach by concatenating the condition vector to the input of the discriminator. [120, 77, 126, 106] proposed a late fusion by encoding the conditional information and introducing it into the final

layers of the discriminator. [155] replaces the discriminator with a pixel-wise semantic segmentation network. Several papers improve results by adding various loss terms to the discriminator [94, 127, 39, 29], however, they don't explicitly focus on testing and constraining the conditionality of the discriminator. Similar to[120], [79] proposes an auxiliary classifier to the discriminator and use of Crammer-Singer multi-hinge loss to enforce conditionality. However, this method is task specific to only generation conditioned on class labels. The conditionality will be analysed further in Chapter 3

2.1.4 Predictive models

Predictive models aim to model the distribution of $p(\mathbf{y}|\mathbf{x})$ such that $\mathbf{y}$ is a low-dimensional label for the input $\mathbf{x}$. Prediction, estimation, inference and forecasting, these terms are used interchangeably in the literature and depend on the context of the task. However, a more formal definition to these terms is provided here:

- **Prediction**: as mentioned earlier, predictive models aim to model the distribution of $p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta})$. For a new measurement, $\mathbf{x}_{new}$, these models "predicts" a new $\mathbf{y}_{new}$. Prediction could be related to time such as predicting the y_{t+n} based on x_t that is modeling, $p(\mathbf{y_{t+n}}|\mathbf{x_t};\boldsymbol{\theta})$ or not related to time such as predicting the bounding box of an object that is modeling $p(\mathbf{b}|\mathbf{x};\boldsymbol{\theta})$ and $\mathbf{b}$ is the bounding box of an object present in the image $\mathbf{x}$.

- **Inference:** This term is often used in the literature to determine the latent variables that generate the observed data $p(\mathbf{h}|\mathbf{x})$. In the deep learning community, the term inference and prediction are interchangeable [123], Both aim to model a distribution of some variable $\mathbf{y}$ given some other variable $\mathbf{x}$ *i. e.* $p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta})$ but they focus on different aspects of this process. Inference involves determining the distribution of latent variables based on the observed data. One common method for doing this is Bayesian inference or using approximate inference. While prediction is used to determine the actual output $\mathbf{y}$ of the predictive model.

- **Forecasting:** it is a specialized formulation of prediction that is concerned with predicting the future value $\mathbf{y_{t+n}}$ based on the present and the past measurements $\mathbf{x_{t-k:t}}$. It models the distribution $p(\mathbf{y_{t+n}}|\mathbf{x_{t-k:t}})$

- **Estimation**: Kent *et al* [81] defines the process of estimation as "using the value of a statistic derived from a sample to estimate the value of a corresponding population parameter". In the context of deep learning, it involves finding the model parameters that minimize the error between the model's predictions and the true values of the output variables.

2.2 Depth prediction

The ability to perceive the 3D structure of the world is a fundamental aspect of human vision. It allows us to perceive the distance and relative position of objects in the world, as well as to judge the size, shape, and orientation of objects. One common use is in the field of robotics is autonomous driving (AD) and advanced driver-assistance system (ADAS), depth perception is used for path planing, automatic emergency braking (AEB), automatic cruise control (ACC) and many more.

The human brain is able to recognize the depth of the objects based on several clues that are present in the visual scene such as disparity, motion parallax, perspective, memorization, and shading. For example, the lateral separation of the eyes enables identifying an object from two angles, The brain uses this difference to compute the relative disparity. Motion parallax is the relative motion of objects at different distances as the viewer moves or the camera pans. The brain uses this motion to infer the depth of objects.

2.2.1 Camera versus LiDAR for autonomous driving

The use of cameras and LiDAR in autonomous driving is an ongoing topic of discussion among researchers and engineers. Although both technologies have their own advantages and disadvantages, the cameras are generally considered to be more affordable and versatile, while LiDAR is known for its superior accuracy and range.

LiDAR (Light Detection and Ranging) is an active 3D sensor that emits light to measure the distance to objects and provide a high-resolution 3D representation of the environment. LiDAR has become an increasingly popular technology in a wide range of applications such as autonomous vehicles due to its several advantages.

- One of the main advantages of LiDAR is its high accuracy. Lidar can acquire high-precision data with a resolution of a few centimeters, making it suitable for applications that require precise measurements. In addition, LiDAR has a long range and can detect objects several kilometers away. This makes it useful in applications such as self-driving cars, localizing and mapping. Another advantage of LiDAR is its 3D imaging capabilities. LiDAR can produce real-time 3D point clouds of the environment, allowing for the creation of detailed 3D maps and models. Furthermore, LiDAR as active sensor can operate in any poor lighting conditions, making it useful for both day and nighttime applications. However, despite its advantages, LiDAR has several disadvantages, the most notable being the cost, which can range from €4,000 to €80,000, making it less cost-effective than cameras. Additionally, LiDAR systems are weather sensitive and can be affected by conditions such as fog or rain, which can affect their accuracy and effectiveness. In addition, LiDAR systems consume a significant amount of power and are typically larger and heavier than cameras.

On the other hand, cameras are passive sensor that detect and measure the intensity of electromagnetic radiation, such as light. it consists of a lens to direct the incoming light and a light sensor to measure the intensity. Cameras have several advantages compared to LiDAR :

- The main advantages of cameras is their cost-effectiveness. They are widely available at a lower cost compared to LiDAR systems. Given the small margins in the automotive industry, the use of LiDAR is not practical and cameras are a more cost-effective sensor option. Cameras also have the added benefit of being able to perform multiple tasks, such as object detection, lane segmentation, and traffic sign recognition, making them a more attractive option for self-driving vehicles. Despite its advantages, cameras have some other limitations, including lighting conditions. Cameras can be affected by poor lighting conditions, which can make it harder to obtain accurate images. They also have a limited range, which means they can only detect objects within a certain distance. Using cameras for mapping can raise privacy concerns, particularly when they are used in public spaces.

Fusing LiDAR and camera modalities has proven to be an extremely effective approach for autonomous vehicles and other applications. The LiDAR sensor provides precise 3D spatial information, while the cameras provide high-resolution visual data. By using data from both modalities, it is possible to improve the accuracy and robustness of object detection, location, and tracking. In particular, LiDAR can provide accurate depth information and help detect objects in low light conditions or obscured by other objects. On the other hand, cameras can provide detailed visual information that can be used to detect objects and improve their identification. Furthermore, integrating both modalities reduces the disadvantages of each individual sensor, such as using LiDAR to detect objects in poor weather conditions and cameras to detect objects in optimal weather conditions. In summary, the fusion of LiDAR and camera data provides a more complete and accurate view of the environment, enabling the development of safer and more reliable autonomous systems.

2.2.2 Deep learning for depth prediction

In recent years, deep learning has emerged as a powerful tool in the field of depth prediction. This is due in large part to the success of convolutional neural networks (CNNs), which have surpassed other classical methods that rely on hand-crafted features. One of the key advantages of deep learning is its ability to learn features directly from data. This allows the model to automatically adapt to the task at hand and model complex situations, making it more powerful tool than traditional methods that relied on hand-crafted features. Furthermore, CNNs have a hierarchical structure, which allows them to learn features at different levels of abstraction. This hierarchical structure is especially useful for depth prediction, as it allows the model to learn both low-level features, such as edges and textures, as well as high-level features, such as object shape and context.

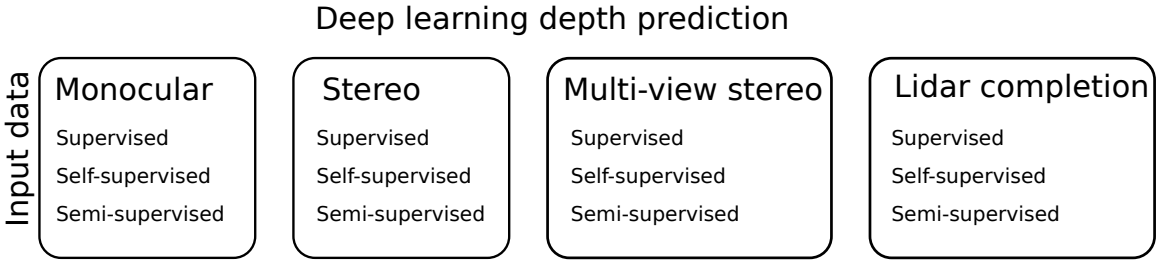


Figure 2.2: Taxonomy of deep learning methods for depth prediction

Depth prediction taxonomy

Depth prediction is a challenging task that has been widely studied in the field of computer vision. One way to organize the various methods and models for depth prediction is through a taxonomy based on the input data or the supervision used Fig. 2.2.

Classification based on input data

One way to classify the methods for depth prediction is based on the input data used. This includes:

- **Monocular depth prediction:** uses a single camera and predict the depth information from it.
- **Stereo depth prediction:** uses two cameras to capture the scene from different viewpoints and predict the depth information.
- **Multi-view depth prediction:** use multiple images captured from different viewpoints as input and estimate depth information by combining information from all the images. These methods have been shown to be effective in improving the performance of depth prediction on large-scale datasets. Multi-view depth prediction methods benefit from the strong epipolar geometry prior. This constraint provides additional information that can be used to improve the accuracy of depth prediction.
- **LiDAR's completion:** LiDAR completion is a method that uses a partial depth map acquired by a LiDAR sensor as input and infers the missing depth information. LiDAR's sensors are known for their high accuracy and resolution, which makes them suitable for depth prediction. LiDAR's completion aim to generate a dense depth map from the sparse LiDAR map using camera information.

Deep learning models perform best when provided with more information. Among the different methods for depth prediction, LiDAR completion has the lowest entropy (*i. e.* provide the lowest uncertainty), as it already provides a sparse and accurate depth map that only needs to be completed.

Multi-view systems come in second, as this method leverages more information and have a strong epipolar geometry constraint. On the other hand, monocular depth prediction has the highest entropy. This is because extracting 3D information from a single 2D image is an ill-posed problem. The scale of the objects in the scene is not known. Monocular depth prediction methods have to rely on other cues, such as texture, color, and motion, to estimate depth. These methods have the highest uncertainty and are prone to the scale ambiguity problem. However, the performance gap of these methods is closing and the prevalence of monocular cameras is making their applications more accessible and interesting. Another aspect to consider is the application: a monocular camera is found on all devices nowadays. Performing depth on these devices is more interesting as it enables a wide range of applications.

Classification based on learning

Another way to classify depth prediction methods is based on the type of supervision used during training. This includes:

- **Supervised depth prediction:** uses ground truth depth maps as the supervision during training. Supervised learning is the best-performing method for depth prediction in the benchmark, as it uses acquired depth maps obtained from depth sensors as supervision for the network. Assuming good accuracy of the ground-truth provided by depth sensors, these models have the best performance as the information provided for supervision is accurate and reliable. However, the use of ground-truth depth maps for supervision is not always feasible, as they can be expensive to acquire or unavailable. Furthermore, as the amount of the available labeled data is very limited, the generalization of these models are not guaranteed when deployed in complex environment.

- **Self-supervised depth prediction:** these methods have been proposed as an alternative to supervised methods. These methods use a differentiable warping to reconstruct a set of a source images (monocular video, stereo video ... etc.) into a target images. This is done using a depth network and pose a network when the pose is not known (*e. g.* the case of monocular video). These models are very practical as they can be easily trained, as they only require a video as input. The data is cheap to collect and widely available, as billions of videos are already available on platforms such as YouTube.

- **Semi-supervised depth prediction:** proposed as a solution that combines the advantages of both supervised and self-supervised methods. These models use a combination of ground-truth depth maps and other forms of supervision, such as depth estimates from self-supervised methods or depth priors. These methods have been shown to be effective in improving the performance of depth prediction on large-scale datasets, while also being more practical as they do not require ground-truth depth maps for all the data.

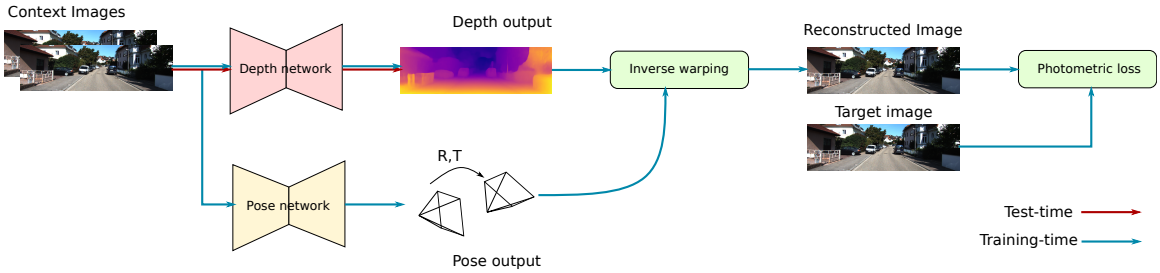


Figure 2.3: Training-time versus test-time for self-supervised depth prediction. The red arrow shows the test-time pipeline, while the blue arrows show the training-time pipeline. The warping function and the pose network are used only during training.

2.2.3 Self-supervised monocular depth

The focus of this thesis is on the self-supervised learning (SSL) of monocular image or video to depth of the corresponding scene. In this section, a brief introduction is presented on the pipeline of the SSL depth. Further development of the warping, the objective, and training will be done in Chapter 4.

Fig. 2.3 Shows the pipeline for self-supervised monocular depth prediction. It composes two networks, a depth network and a pose network. During training, the forward pass is propagated through the depth network and the pose network. The output of these networks will then be used to reverse warp a source image to reconstruct the target image. The photometric loss is calculated and the backward pass optimize both depth and pose network parameter simultaneously. At test-time, **only the depth network** is used and the pose network, unless used to reconstruct the image, it is discarded.

The SSL depth pipeline comprises several crucial elements, including the depth network, the pose network, and the photometric loss. The depth network leverages convolutional neural networks (CNNs) or other deep learning architectures to learn the intricate relationships between pixel intensities and depth values. The pose network, on the other hand, is responsible for estimating the camera's motion or pose between different frames in a video sequence. By accurately determining the camera's movement, the pose network aids in aligning the source and target images during the reconstruction process.

2.2.4 Limitations

Self-supervised learning (SSL) methods for monocular depth prediction have shown promise, but it is important to acknowledge their limitations. In this section, we discuss some of these limitations:

- **Reliance on the Assumption of Photometric Consistency**: SSL methods heavily rely on the assumption that pixel intensities remain consistent across different views of the same scene, which may not hold true in challenging real-world scenarios. Variations in lighting conditions,

occlusions, this assumption and lead to inaccurate depth predictions. Several methods were proposed to account for this problem.

- **Requirement for Large Amounts of Training Data**: SSL methods often demand a substantial amount of unlabeled data for effective training. While unlabeled data is typically abundant and easy to obtain, it can be challenging to ensure its quality and diversity. Inadequate or biased training data may result in suboptimal depth estimation performance and generalization to real-world scenarios.

- **Limitation of Monocular Input**: The reliance on monocular input limits the accuracy of depth estimation compared to methods that utilize stereo or multi-view setups. Monocular depth estimation inherently suffers from scale ambiguity, as a single 2D image cannot provide sufficient information to uniquely determine 3D values.

- **Rigid Scene Assumption**: The vast majority of self-supervised learning methods for monocular depth estimation assume a rigid scene, meaning they do not handle dynamic objects. Moving objects within the scene can disrupt the consistency assumption and introduce errors in depth estimation. This limitation restricts the applicability of SSL methods in scenarios with significant object motion or dynamic environments.

2.2.5 Related work

Depth prediction has been successful with self-supervised learning from videos. The seminal work of Zhou *et al* [189] introduced the core idea to jointly optimize the pose and depth network using image reconstruction and a photometric loss. Due to its simplicity and generality, this approach has attracted significant attention from researchers, leading to a series of related works, including [163, 177, 111, 29, 7, 30, 141, 167, 116, 173].

Recognizing the inherent challenges associated with the ill-posed nature of depth prediction, researchers have endeavored to tackle various aspects. Addressing the rigid scene assumption, Vijayanarasimhan et al. [163], Xu et al. [177], and Luo et al. [111] employed optical flow and motion clustering techniques to overcome this limitation. Chen et al. [29], on the other hand, focused on enhancing generalization by incorporating camera parameter learning into the framework. Meanwhile, Bian et al. [7], Chen et al. [30], Rares et al. [141], and Wang et al. [167] directed their efforts towards mitigating the scale ambiguity problem, proposing innovative approaches that enforce depth scale and structure consistency. Furthermore, to enhance the performance of depth prediction models during inference, McCraith et al. [116] and Watson et al. [173] introduced test-time refinement strategies. These techniques allow for dynamic variation of model parameters using a photometric loss, thereby refining the depth estimation results.

In summary, this thesis explores the advancements in the field of depth prediction through self-supervised learning from videos. It addresses the problem of rigid scene assumption with a novel

method [14] presented in Chapter 4. Furthermore, the framework is extended to enable future forecasting [13], as discussed in Chapter 5. Finally, a novel method is introduced in Chapter 6, which outputs a sequence of future depth predictions [10].

2.2.6 Depth evaluation

The evaluation of depth estimation methods often relies on the KITTI benchmark [49], which is widely used in the field. The KITTI dataset provides a set of sequence captured from a car-mounted camera, accompanied by accurate depth maps obtained using LiDAR sensors. The Eigen *et al* [44] train/validation split is widly used to train and evaluate the models. To evaluate the quality of depth estimation results, several metrics are commonly employed. These metrics are defined in Table 2.1

Metric	**Formula**	**Range**	**Description**
RMSE	$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(d_i - \hat{d}_i)^2}$	$[0, \infty)$	Root Mean Squared Error
Abs rel	$\frac{1}{N}\sum_{i=1}^{N}\frac{\lvert d_i - \hat{d}_i\rvert}{d_i}$	$[0, \infty)$	Absolute Relative Difference
Sq rel	$\frac{1}{N}\sum_{i=1}^{N}\frac{\sqrt{(d_i - \hat{d}_i)^2}}{d_i}$	$[0, \infty)$	Squared Relative Difference
Log RMSE	$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\log(d_i) - \log(\hat{d}_i))^2}$	$[0, \infty)$	Logarithmic Root Mean Squared Error
$\delta < 1.25$	$\frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(\max(\frac{d_i}{\hat{d}_i}, \frac{\hat{d}_i}{d_i}) < 1.25)$	$[0, 1]$	% of pixels with $\delta < 1.25$
$\delta < 1.25^2$	$\frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(\max(\frac{d_i}{\hat{d}_i}, \frac{\hat{d}_i}{d_i}) < 1.25^2)$	$[0, 1]$	% of pixels with $\delta < 1.25^2$
$\delta < 1.25^3$	$\frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(\max(\frac{d_i}{\hat{d}_i}, \frac{\hat{d}_i}{d_i}) < 1.25^3)$	$[0, 1]$	% of pixels with $\delta < 1.25^3$

Table 2.1: Depth Benchmark Evaluation Metrics

2.3 Style transfer using conditional GANs

Style transfer is a fascinating technique that allows the transformation of the visual style of an image while preserving its underlying content. Conditional generative adversarial networks (cGANs) have emerged as a powerful approach for achieving style transfer in an automated and data-driven manner. By training GANs on large datasets of paired images with different styles, these models can learn to capture the essence of each style and apply it to new images. The following section delves into two notable examples of conditional GANs for style transfer: Pix2Pix and CycleGAN. These two networks will be used extensively in Chapter 3.

2.3.1 Pix2Pix for paired datasets

This section provides a brief introduction to this architecture. Additional information can be found in [71]. Pix2Pix, introduced by Isola *et al* [71] is a pioneering architecture that showcases the potential of conditional GANs in image-to-image translation tasks. It provides a framework for translating

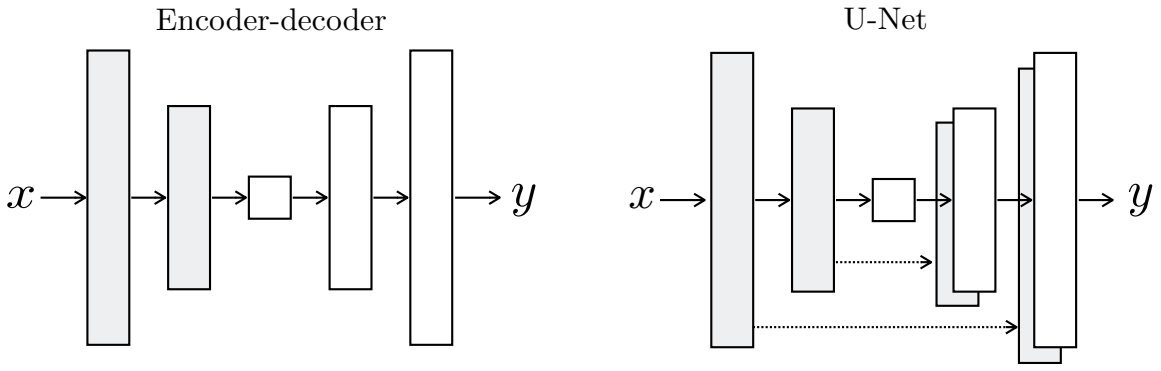


Figure 2.4: The Pix2Pix architecture with encoder-decoder structure and skip connections, inspired by the U-Net architecture. The encoder captures essential features from the source image, while the decoder generates the corresponding output image in the target domain. The inclusion of skip connections helps preserve details, leading to high-quality outputs. Figure from [71]

.

images from a source domain to a target domain while maintaining their content. Pix2Pix's strength lies in its ability to bridge the gap between domains by learning the mapping between paired images.

Generative Adversarial Networks (GANs) are models designed to generate realistic images by learning a mapping from a random noise vector ($\mathbf{z}$) to an output image ($\mathbf{y}$), denoted as $\mathbf{y} = G(\mathbf{z})$. However, conditional GANs take it a step further by learning a mapping from both an observed image ($\mathbf{x}$) and a random noise vector ($\mathbf{z}$) to the output image ($\mathbf{y}$), expressed as $\mathbf{y} = G(\mathbf{x}, \mathbf{z})$. The main objective is for the generator (G) to produce outputs that are indistinguishable from authentic images, thereby fooling a discriminator (D) trained to identify the generator's fakes.

As shown in Fig. 2.4, The Pix2Pix architecture employs an encoder-decoder structure with skip connections, inspired by the U-Net architecture. The encoder captures essential features from the source image, while the decoder generates the corresponding output image in the target domain. The skip connections aid in preserving details during the translation process, resulting in high-quality outputs.

Objective

The objective of a conditional GAN can be expressed as

$$\begin{aligned}\mathcal{L}_{cGAN}(G, D) =& \mathbb{E}_{x,y}[\log D(x, y)] + \\ & \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z))],\end{aligned} \tag{2.12}$$

where G is the generator that tries to minimize this objective against a discriminator D that tries to maximize it, i.e. $G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D)$.

Previous approaches have found it beneficial to mix the GAN objective with a more traditional

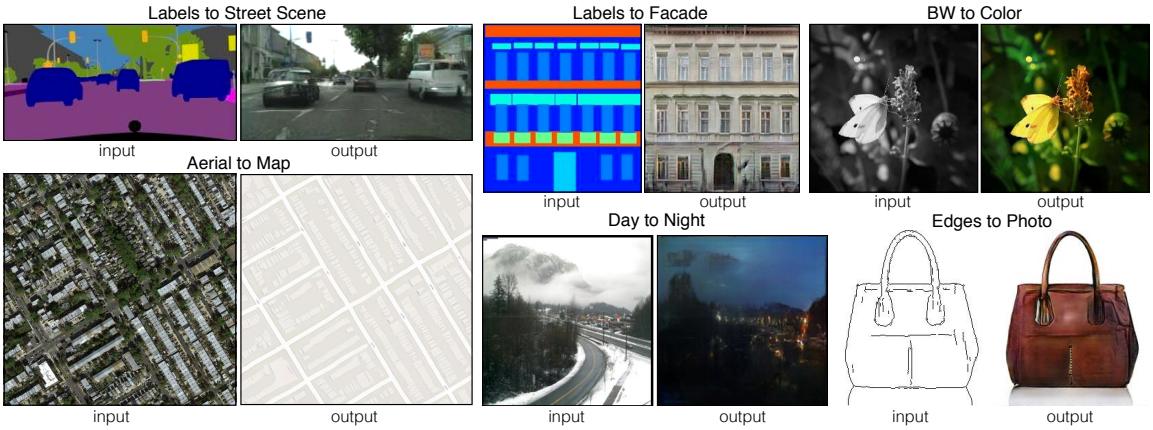


Figure 2.5: The figure displays a curated set of images demonstrating the effectiveness of Pix2Pix in transforming images from the source domain to the target domain while preserving essential details and structural integrity. Figure from [71]

loss, such as L2 distance [132]. The discriminator's job remains unchanged, but the generator is tasked to not only fool the discriminator, but also to be near the ground truth output in an L1 sense. This will be explored further in Chapter 3 when the conditionality of cGAN is analyzed

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}|y - G(x,z)|. \tag{2.13}$$

The final objective is

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G,D) + \lambda \mathcal{L}_{L1}(G). \tag{2.14}$$

Pix2Pix requires a paired dataset for training. In other words, it relies on having images from the source domain and their corresponding images in the target domain. These paired images serve as the training data, enabling Pix2Pix to learn the mapping between the two domains. Fig. 2.5 shows the qualitative results of the style transfer of Pix2Pix and showcases a selection of images that have undergone style transfer. It illustrates how the model successfully transforms images from the source domain into the target domain while preserving important details and structure.

This basic architecture will be used to study the conditionality and based on that analysis a novel method will be proposed to improve the conditionality of cGANs.

2.3.2 CycleGAN for unpaired datasets

This section provides a brief overview of CycleGAN [190], a framework for unpaired image-to-image translation. While traditional methods like Pix2Pix require paired images for training, CycleGAN addresses this limitation by enabling style transfer between domains without the need for explicit

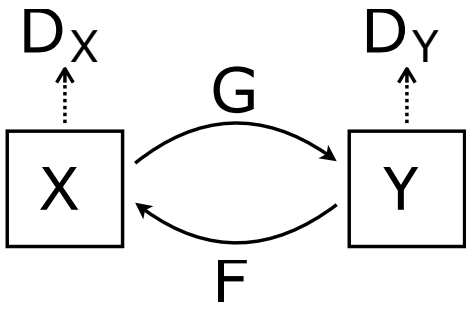


Figure 2.6: CycleGAN pipeline. The figure illustrates two generator networks, $G : X \to Y$ and $F : Y \to X$, along with two discriminator networks, D_x and D_y. The generators learn to translate images between domains, while the discriminators distinguish between translated and real images. Figure adopted from [190]

pairs of corresponding images.

As shown in Fig. 2.6, CycleGAN consists of two generator networks, namely the generator $G : X \to Y$ and the reverse generator $F : Y \to X$, along with two discriminator networks, D_x and D_y. The generators learn to translate images from one domain to another and back, while the discriminators aim to distinguish between the translated images and real images from each domain. To ensure content preservation during the translation process, CycleGAN incorporates a cycle consistency loss, which enforces that the reconstructed image should resemble the original image.

The objective of CycleGAN involves adversarial losses [53] for both mapping functions. For the mapping function $G : X \to Y$ and its discriminator D_Y, the objective is expressed as:

$$\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log D_Y(y)] \ + \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x)))],$$

where G generates images $G(x)$ that resemble images from domain Y, and D_Y distinguishes between translated samples $G(x)$ and real samples y. A similar adversarial loss is introduced for the mapping function $F : Y \to X$ and its discriminator D_X as well.

However, adversarial losses alone cannot guarantee that the learned function can accurately map an individual input to a desired output. To address this, CycleGAN introduces cycle consistency, which ensures that the learned mapping functions are cycle-consistent. This means that for each image x from domain X, the image translation cycle should be able to bring x back to the original image ($x \to G(x) \to F(G(x)) \approx x$), and vice versa for images from domain Y. This behavior is incentivized using a cycle consistency loss:

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}|F(G(x)) - x| + \mathbb{E}_{y \sim p_{\text{data}}(y)}|G(F(y)) - y|. \tag{2.15}$$

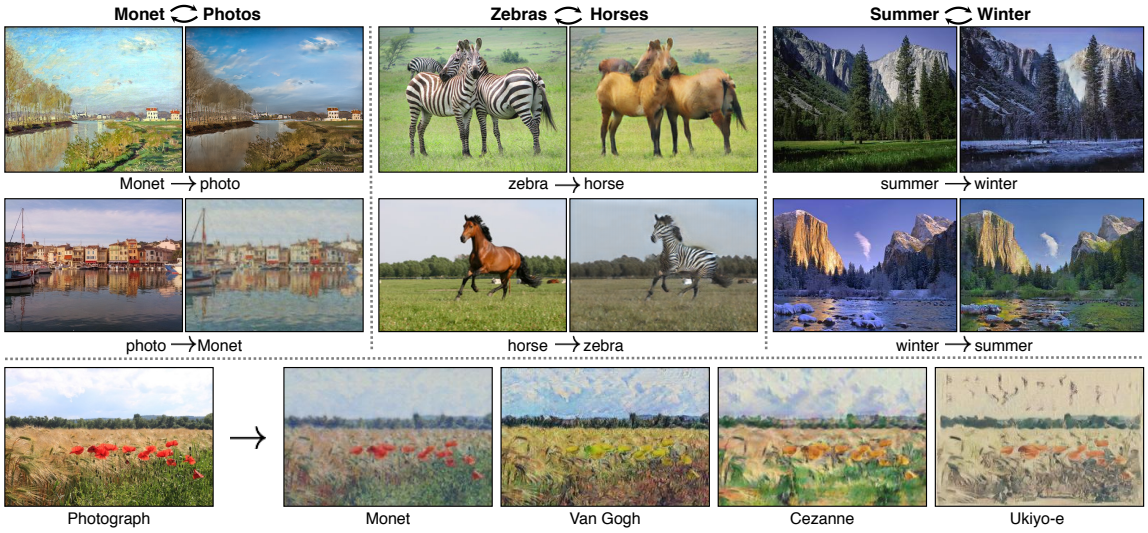


Figure 2.7: Qualitative results achieved by CycleGAN in various translation settings. The images demonstrate the effective conversion from one domain to another while preserving essential visual characteristics. CycleGAN exhibits versatility in handling diverse types of translations, maintaining the original input's structure and content while incorporating distinct attributes of the target domain.

The full objective of CycleGAN combines the adversarial losses and the cycle consistency loss:

$$\mathcal{L}_(G, F, D_X, D_Y) = \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) + +\lambda \mathcal{L}_{\text{cyc}}(G, F), \tag{2.16}$$

where λ controls the relative importance of the two objectives. The aim is to solve the optimization problem:

$$G, F = \arg\min_{G,F} \max_{D_x, D_Y} \mathcal{L}(G, F, D_X, D_Y). \tag{2.17}$$

Fig. 2.7 illustrates the qualitative results achieved by CycleGAN in various translation settings, demonstrating its ability to effectively convert images from one domain to another while maintaining crucial visual characteristics. These results showcase CycleGAN's versatility in handling different types of translations. The translated images successfully retain the overall structure and content of the original input, while incorporating the unique attributes of the target domain.

In this thesis, the exploration of Generative Adversarial Networks (GANs) began in the early stages of 2020. During that time, GANs gained popularity for their remarkable ability to perform style transfer and generate visually appealing images. However, since then, more recent methods have emerged that exhibit even more impressive results, surpassing the capabilities of GANs. One such

method is the employment of diffusion models [146, 139]. These models have demonstrated superior performance in various image generation tasks. Nonetheless, the focus of this thesis primarily revolves around depth prediction rather than image generation. Therefore, while these advanced techniques like diffusion models exist, they fall beyond the scope of this particular study.

2.4 Image segmentation

The utilization of segmentation representation in Chapter 3 and Chapter 4 of this thesis indicates its significance, thereby suggesting the need to expand upon the background information. Intelligent systems require the ability to reason about their environment in order to make accurate decisions. However, the raw representation of image intensity (i.e., a pixel value matrix) is not adapted to this goal. A more abstract representation is required, where each pixel encodes a more abstract information rather than an intensity value. This representation is known as segmentation. Image segmentation involves dividing an image into distinct meaningful domains, transforming it into a simplified, high-level representation. This type of representation has received a lot of attention in recent years due to its wide range of applications, including autonomous driving, medical imaging, etc. There are three main types of segmentation: semantic segmentation, instance segmentation and panoptic segmentation.

2.4.1 Semantic segmentation

Semantic segmentation is a form of image segmentation that assigns class labels to each pixel within an image, where these class labels correspond to objects or regions present within the image. For instance, in a semantic segmentation of a city street, pixels associated with the road, buildings, trees, sky, and other objects would be assigned different class labels. This approach to segmentation is valuable for tasks such as scene understanding, object detection, and scene labeling.

In recent years, there has been substantial progress in the field of semantic segmentation. Classical methods were based on hand-crafted features. These models are limited by the representation as it does not fully capture high-level and low-level relations, thus limiting their performance. With the recent advance of deep learning methods, researchers have extended such methods to semantic segmentation. (FCNs) [109], revolutionized the field of semantic segmentation. It is an encoder-decoder architecture where the encoder is based on the VGG-16 [151] architecture, and the decoder consists of convolution and transposed convolution layers. The subsequent SegNet [5] architecture introduced novel layers for upsampling in place of transposed convolutions, while ParseNet [105] modeled global context directly. The PSPNet [186] architecture focused on multi-scale features, proposing pyramid pooling to learn feature representations at different scales. DeepLabV3+ [21] proposed the Atrous Spatial Pyramid Pooling (ASPP) module to improve the receptive field of the backbone.

With the advent of transformers and visual transformer (ViT) [162], several methods were proposed. [17] proposed a hybrid architecture with a CNN based backbone and a transformer encoder-decoder to perform as semantic segmentation decoder. (ViT) [41] has proposed the first end-to-end backbone transformer based model for segmentation. Swin transformers [107] made it possible to process high resolution images efficiently. With the success of multi-modalites pre-training on video audio and text, semantic segmentation have benefitted from this advancement. [154] combined all the pre-training paradigm, including supervised pre-training, weakly-supervised and self-supervised resulting in an state-of-art results on ADE20K benchmark [187]. [31] pretrained a model on image, text and used an adapter to introduce the image-related inductive biases into the model. This representation is used to validate domain adaptation in chapter 3.

2.4.2 Instance segmentation

Instance segmentation aims to recognizing individual instances of objects within the image. This is particularly useful for tasks like object tracking and counting, or for identifying specific instances of objects for further analysis. For instance, in an instance of segmentation of a street scene, individual cars and bicycles would be distinguished and separated from one another.

Mask R-CNN [60] is one of the most popular models for. Fig. 2.8 Shows the architecture of the mask-RCNN model. Similar to [144], this model consists of a backbone that extracts the features from the image. Typically, a ResNet[21]. In the first stage, A RPN network uses these features to propose N possible object in the scene. For the second stage, This method proposed a *RoIAlign* module that pools the features with better alignment to the input, and incorporates an additional object segmentation branch, in parallel to the bounding box regression and classification branch of [144]. Further advancements have been made with proposal based networks [104, 98, 19], single stage network proposal free-networks [8, 47, 93], and transformers based networks [26, 26, 59]. This network is extended in Chapter 4 with the object pose.

2.4.3 Panoptic segmentation

Panoptic segmentation combines semantic segmentation with instance segmentation. This task is represented with a foreground/background classes. The fourground 'thing' class represents countable categories in the real world, such as people, cars. Each of these objects is assigned a unique identifier along the object mask. The background 'stuff' class represents categories that cannot be counted, such as road and wall. It was shown that combining these two tasks improves the performance for both tasks. The introduction of panoptic segmentation was first presented in the seminal work of Kirillov *et al* [84]. The authors of this work formulated the task, established evaluation parameters, and presented a basic baseline. Since then, the task has received considerable attention in the research community, resulting in numerous techniques and approaches [121, 84, 28, 23, 168, 27].

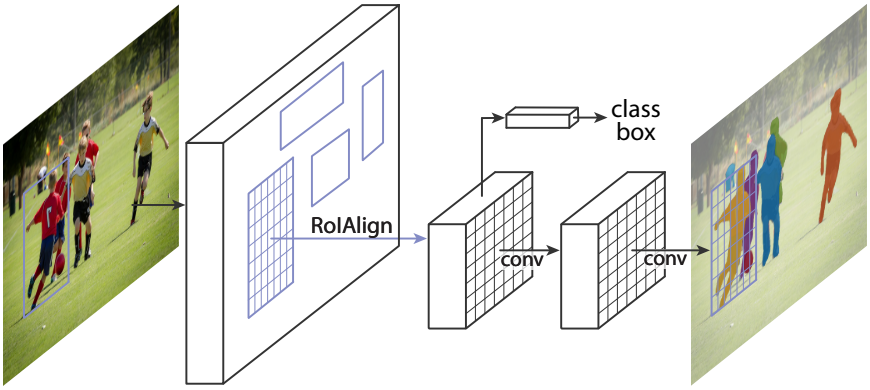


Figure 2.8: Overview of Mask-RCNN architecture. Similar to [144], after extracting the features of an object proposal from image, the RoIAlign module is used to pool the features of tha object. 3 Parallel heads are used after, the bounding box regression head, the class head, and the mask head. Figure adopted from [60]

EfficientPS [121] is a panoptic segmentation networj that improves upon EfficientNet [157] by proposing 2-way FPN that both encodes and aggregates semantically rich multiscale features in a top-down and bottom-up way. This model consists of a shared backbone, EfficientNet [157], the complexity of this model could be controlled. The two-way FPN that aggregates the features in a top-down and bottom-up way, resulting in a semantically rich multiscale features. The semantic segmentation head uses the output of the FPN to perform segmentation using DPC and modules [121]. The instance information is extracted using a Mask-RCNN [60] head. And finally, the panoptic fusion head fuses the semantic and instance segmentations and outputs the panotic segmentation. This model has several advantageous including, the ability to control the backbone's complexity, a state-of-art results on Cityscapes [35] benchmark. The code is easy to integrate, and it is open-sourced. This model was extended with depth, ego-pose and object pose information and will be presented in Chapter 4.

When evaluating the performance of segmentation algorithms, several metrics are commonly used. In the context of semantic segmentation, the goal is to assign a label to each pixel in an image. Instance segmentation extends this by not only labeling each pixel, but also separating individual instances of objects. Panoptic segmentation combines both semantic and instance segmentation by providing a unified labeling scheme for all pixels, including both things and stuff classes.

- **Mean Intersection over Union (mIoU)**: Also known as Jaccard Index, it measures the overlap between the predicted segmentation and the ground truth. It is computed as: $mIoU = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i+FP_i+FN_i}$, where N is the number of classes and TP, FP, and FN represent true positive, false positive, and false negative pixels, respectively, for each class.

- **Panoptic Quality (PQ)**: PQ measures the overall quality of panoptic segmentation, considering both semantic segmentation and instance segmentation. It combines the accuracy of

segmentation masks and the alignment between predicted segments and ground truth objects. PQ is computed as follows: $PQ = \frac{\sum_{(p,g)\in \mathrm{TP}} \mathrm{IoU(p,g)}}{\mathrm{TP}+\frac{1}{2}\mathrm{FP}+\frac{1}{2}\mathrm{FN}}$. The numerator sums up the Intersection over Union (IoU) ratios for all true positive (TP) instances. The denominator is combines precision and recall, dividing all the True Positives and half the False Positives and False Negatives.

2.5 The transformer modules

In the seminal work, Vaswani *et al* [162] have introduced the transformer. A neural network module that relies on attention to perform computation on sequences. Initially this module was intended for natural language processing (NLP), however, its success has shone in all data domains including, audio, image, video ... etc. This module has revolutionized the field, with great success in nearly all modalities and cross-modality tasks. In this section, we will first introduce the original transformers [162] and the image variation derived from it [107, 41]. In this thesis, these modules are used extensively to construct the architecture of the proposed models. Therefore, the background of these building blocks is developed in this section.

2.5.1 Attention is all you need: the transformer

Classical sequence modeling approaches used LSTMs and RNNs which processed data sequentially, where the current state S_t was based on the previous state S_{t-1} and an implicit memory that represents the past. This method had limitations as the current state did not have explicit access to all previous states and only relied on the implicit memory. Transformers were introduced to overcome this problem by using an attention mechanism. This allowed each state S_i to pay attention to all other states based on their importance, determined by a weighting in the attention map. For example, in the sentence "A man walked into the park with his dog. He stopped to tie his shoe." the state "He" and "his" would pay more attention to "A man" as it is more relevant to these state.

Fig. 2.9 shows the architecture of the transformer. First, the text input is embedded into a continuous higher dimension. As the transformer is permutation invariant, explicit information of the embedding position should be added. A sinusoidal positional encoding is added to represent the position of each word. The first block of the transformer architecture is the *encoder*. It processes the embedded and positional encoded input through a series of multi-head attention mechanisms, where each attention mechanism allows the model to attend to different parts of the input sequence and to weigh the importance of each part. The multi head attention is defined as:

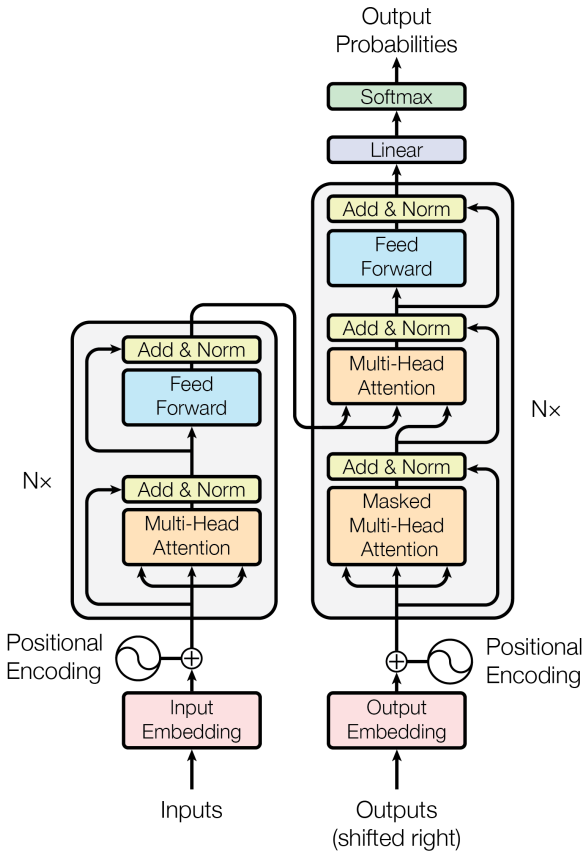


Figure 2.9: Overview of the transformer architecture. Figure adopted from [162]

First we constitute the Key, Query, Value as follows :

$$\text{Key: } \mathbf{K} = \mathbf{X}\mathbf{W}_\mathbf{k} \tag{2.18}$$

$$\text{Query: } \mathbf{Q} = \mathbf{X}\mathbf{W}_\mathbf{q} \tag{2.19}$$

$$\text{Value: } \mathbf{V} = \mathbf{X}\mathbf{W}_\mathbf{v} \tag{2.20}$$

The multi head attention is defined as :

$$\textbf{Attention} = \text{Softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_{dim}}})\mathbf{V} \tag{2.21}$$

The Softmax($\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_{dim}}}$) produces an attention map normalized with the softmax that indicates the relative importance of the values with respect to each other. Thus, enabling the model to attend to the most relevant parts of the input. This hidden state is passed through a feedforward neural network to increase the complexity of the learned function. The skip connection allows better gradient flow. Finally, the output is normalized.

The second block of the transformer architecture is the *decoder*, which generates a target sequence. Similar to the encoder, the decoder also uses multi-head attention mechanisms and feed-forward neural networks. The difference is that the decoder uses the output of the encoder for the key and the value enabling cross attention. The final output of the decoder is then compared to the target sequence and the model is trained to minimize the difference between the two.

This architecture has several advantages as the attention is fully parallelizable and each state have access to all other states. However, the complexity of the attention grows quadratically with the sequence length n.

2.5.2 The Visual transformer (ViT)

Applying directly, self-attention to images requires that each pixel attends to every other. This does not scale for realistic images, as the cost of the attention is quadratic in the number of pixels. To this end, [40] marked the first transformer only architecture. The idea of this paper is instead of using each pixel as input to the standard transformer, the image is divided into small patches of 16×16 and the image will then be represented as a sequence of patches.

The authors applied a standard Transformer enocder [162] directly to images, with the fewest possible modifications. To do so, they split an image into patches of 16×16. Image patches are treated the same way as tokens (words) in an NLP application. They provide this sequence of linear embeddings of these patches as an input to a Transformer.

This architecture has known great success in the computer vision community. This backbone has been applied to nearly all tasks, replacing its CNN counterpart. The flexibility and the global receptive field of transformer enable the context-aware rich feature extraction. However, as the

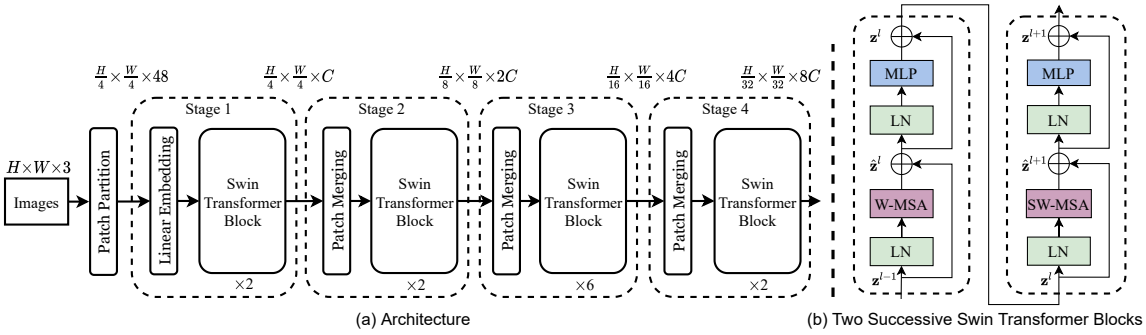


Figure 2.10: Overview of Swin tranformer architecture. Adpoted from [107]

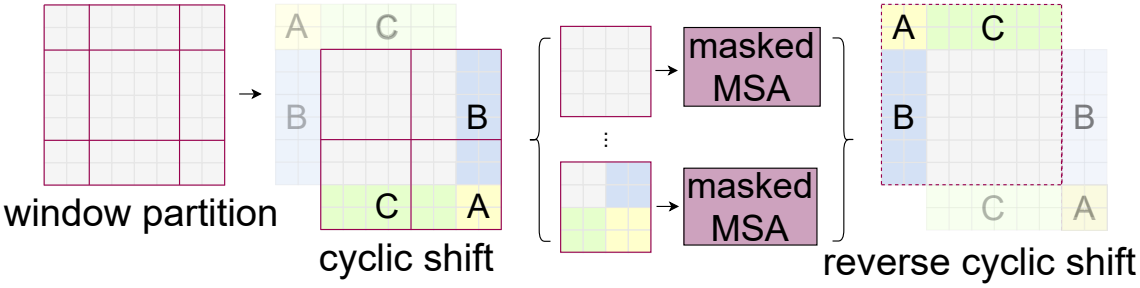


Figure 2.11: Overview of Swin tranformer block. Adpoted from [107]

transformer is permutation invariant and lacks the indicative bias of convolution networks (*i. e.* the spatial information is hard-coded in the convolution filters. a 3×3 convolution is applied locally on a square of 3), it is not able to obtain great results when trained on smaller datasets from scratch.

2.5.3 SwinTranformer (SwinT)

In standard ViTs, the number of tokens and token feature dimension are kept fixed throughout different blocks of the network. This is limiting, since the model is unable to capture fine spatial details at different scales. The Swin Transformer [107] proposes a multi-stage hierarchical feature extraction that computes attention within a local window, by partitioning the window into multiple sub-patches. To capture interactions between different windows (image locations), window partitioning is gradually shifted, along the depth of the network, to capture overlapping regions

Fig. 2.11 shows the architecture. Swin Transformer incorporates a shifted window approach for computing self-attention, as demonstrated in the accompanying illustration. In Layer l (on the left), a standard window partitioning strategy is used and self-attention is computed within each window. In the subsequent Layer l+1 (on the right), the window partitioning is shifted, leading to the creation of new windows. The self-attention computation in these new windows crosses the boundaries of the previous windows in Layer l, thereby establishing connections between the windows.

Chapter 3

Image-to-X with conditional GANs

In the initial stages of the thesis, the problem of using images for autonomous driving was approached from a broader perspective, exploring not only the depth modality as output but also other crucial modalities, such as semantic segmentation and image generation. These modalities play a vital role in enabling driving systems to comprehend the geometry of the surrounding environment and identify different participating agents.

During the first year of the thesis, the primary objective was to model complex environmental situations that autonomous driving systems may encounter. Factors like weather conditions and night-time scenarios present significant obstacles for these systems. It is crucial to develop the capability to handle such variations and ensure consistent performance in diverse circumstances.

This chapter therefore explores domain adaptation techniques using generative adversarial networks (GANs) to bridge the gap between the source and target domains. It focuses using on conditional GANs for style transfer. Specifically, translating annotated overcast-daytime images into night-time or other weather condition offers a cost-effective means of constructing annotated datasets. Additionally, this study reveals a limitation in traditional conditional GANs, which lack full conditional capabilities, which is also known as "hallucination". To address this limitation, a novel approach to train cGANs called "*a contrario* cGAN" is proposed. This method allows for more consistent output generation, making conditional GANs fully conditional. Furthermore, a noteworthy feature developed in this research is the generative model night-to-day, designed specifically for domain adaptation. This model is selected as a potential feature to be added to the Renault ADAS stack.

This chapter is based on two publications :

- Boulahbal Houssem Eddine, Adrian Voicila, and Andrew I. Comport. "Are conditional GANs explicitly conditional?." British Machine Vision Conference. 2021.
- Boulahbal, Houssem Eddine, Adrian Voicila, and Andrew I. Comport. "Un apprentissage

de bout-en-bout d'adaptateur de domaine avec des réseaux antagonistes génératifs de cycles consistants." Journée des Jeunes Chercheurs en Robotique. 2020

3.1 Introduction

In the rapidly evolving field of artificial intelligence, one of the key challenges is the ability to generalize, especially in the domain of Autonomous Driving (AD). Autonomous vehicles are operating in various and dynamic environments where they are faced with a variety of situations, such as weather variations. To ensure safe and reliable function of autonomous vehicles, it is crucial to develop AI models capable of adapting and generalizing to these complex environmental situations.

One major obstacle in training deep learning models for autonomous driving is the availability of annotated datasets. Annotated datasets are essential for supervised learning, where the model is trained on labeled examples to recognize and interpret different objects and events in the environment. However, the manual annotation of a large-scale dataset for autonomous driving is an exceedingly laborious and time-consuming task, which makes it practically impossible to create a fully annotated dataset covering all possible environmental variations.

As shown in Fig. 3.1, the model trained solely on daytime images (DeepLabV3+ [21] performs semantic segmentation) has significant limitations when it comes to accurately predicting outcomes in nighttime scenarios. The dramatic difference between the visual characteristics of daytime and nighttime environments poses a considerable challenge for AI models, as lighting conditions, object appearance and overall scene dynamics undergo significant changes. To circumvent this annotation bottleneck, researchers have explored various techniques, and one promising approach is the use of domain adaptation techniques [165, 180, 149, 82, 133]. Domain adaptation aims to transfer knowledge from a source domain, where labeled data is available, to a target domain, where only unlabeled or sparsely labeled data exists. By leveraging the knowledge from the source domain, domain adaptation techniques enable the model to generalize well to the target domain without requiring extensive annotation efforts.

In this chapter, we propose the utilization of domain adaptation techniques using conditional adversarial generative networks (cGANs), specifically the CycleGAN [190] (Cycle-Consistent Generative Adversarial Network), to address the challenge of translating the source annotated domain into the target domain. The CycleGAN framework is a powerful tool that allows for the generation of synthetic data in the target domain by learning the mapping between the two domains in an unsupervised manner. Furthermore, while developing this method, the conditionality of the cGANs is further explored and novel method is proposed to address the problem of conditionality of conditional GANs.

Figure 3.1: Comparison of DeepLabV3+ [21] model's performance on daytime and nighttime scenarios. The model trained solely on daytime images shows significant limitations in accurately predicting outcomes in nighttime scenarios due to the dramatic differences in lighting conditions, object appearance, and overall scene dynamics.

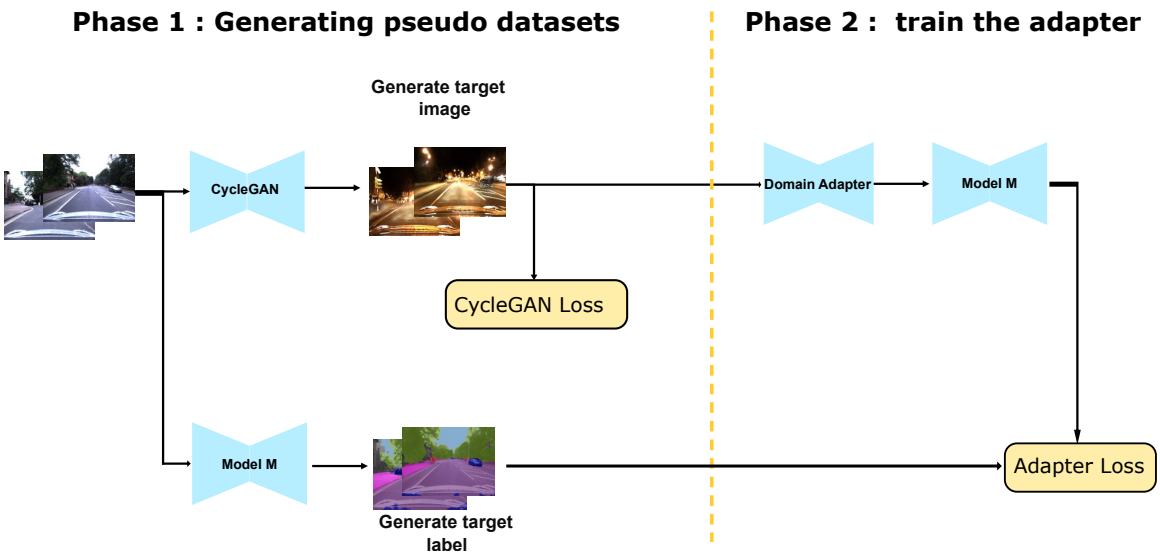


Figure 3.2: This figure shows a two-phase pipeline used to address the domain shift problem. In the first phase, a pseudo dataset is generated using a CycleGAN model that is trained for day-to-night image translation. This translation enables the creation of a synthetic annotated dataset of night images using the model M on the day-time images. The second phase involves training an adapter to handle the day-night variation, maximizing the performance of the model M.

3.2 Improving generalization using cGANs

The first objective was to reproduce the results of the "Don't Worry About the Weather" paper [133]. The pipeline shown in Fig. 3.2. To achieve this, a two-phase pipeline was performed. In this context, a CycleGAN model for image translation is firstly trained. CycleGAN is a popular framework used to learn mappings between two different domains. In our case, we want to perform day-to-night translation. Using this translation, it is possible to construct an annotated dataset of synthetic night images. The annotation of these synthetic night images is done using the model M on the day-time images. The second phase involves training an adapter to account for the shift between the day-night variation, and it is trained to maximize the performance of the model M.

The Robotcar dataset [112] is used to train and evaluate the system. The dataset comprises a diverse collection of sensor data captured by an autonomous vehicle as it navigates Oxford city. The dataset covers a wide range of environmental conditions, such as varying weather conditions (e.g., sunny, cloudy, rainy), different times of day (daytime, nighttime), and diverse urban scenarios (e.g., streets, intersections, landmarks). Overall, the RobotCar dataset provides a comprehensive and representative collection of data for training and evaluating the system. Its diverse environmental conditions make it an ideal choice for domain adaptation experiments. Fig. 3.3 shows the results of translating the day-to-night translation. As observed, the model is able to handle well the variations in lighting conditions and object appearance, successfully transforming daytime images into realistic

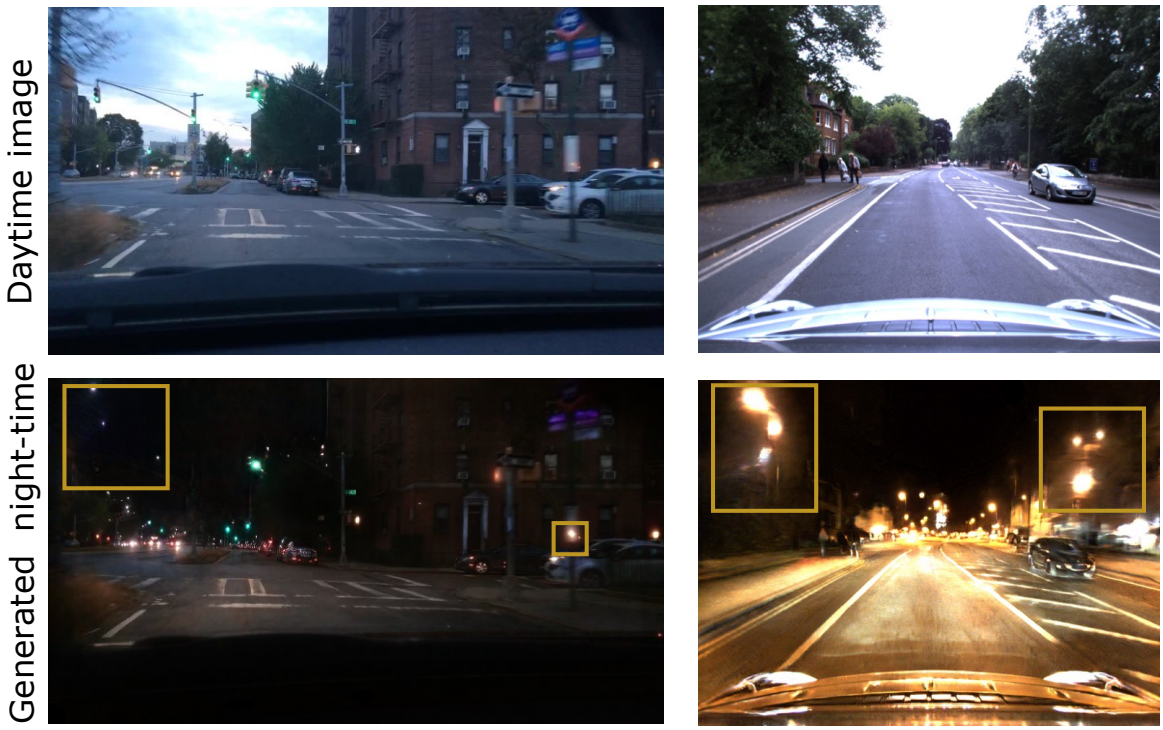


Figure 3.3: Day-to-night translation results. The model successfully handles variations in lighting conditions and object appearance, transforming daytime images into realistic nighttime counterparts. However, occasional hallucinations of poles and objects not present in the scene can be observed, as depicted in the examples shown in the figure.

nighttime counterparts.

To enhance the performance of the system, a domain adapter, A, is trained. This adapter is designed to improve a pretrained and frozen model M, which is the model that will be used in production. The adapter plays a crucial role in accounting for the changes that occur during the translation process. The adapter is supervised using the following loss function:

$$\mathcal{L}_{adpt} = \sum_{i=0}^{m} |M(A(\mathbf{I}^i_{night})) - \mathbf{S}^i_{mask}| \tag{3.1}$$

Where m is the number of images in the dataset, M is the semantic segmentation model (DeepLabV3+ [21]), A is the adapter, $\mathbf{I}^i_{night}$ is the synthetic image obtained from translating $\mathbf{I}^i_{day}$ with the GAN, and $\mathbf{S}^i_{mask}$ is the semantic mask obtained using $\mathbf{S}^i_{mask} = M(\mathbf{I}^i_{day})$. The adapter is trained to make the prediction of the semantic segmentation consistent for day and night.

To effectively handle the domain shift caused by various weather conditions, an adapter is trained to minimize the domain shift between the source domain and the target domain. For instance, during

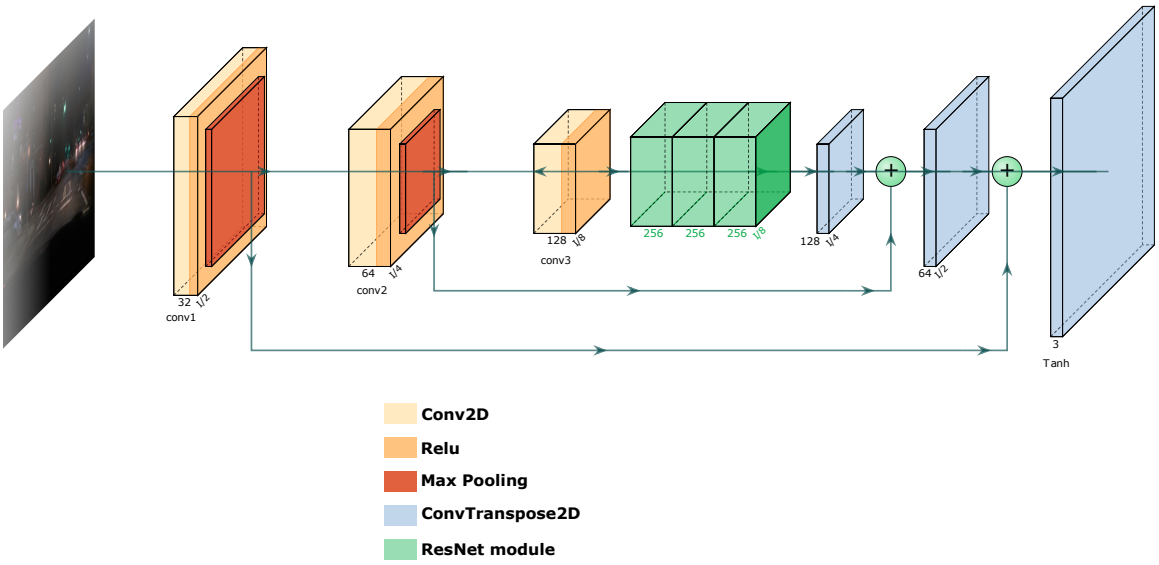


Figure 3.4: Adapter architecture. Leveraging encoder-decoder design. Relu is used as The activation function. Max pooling is used to downsample the feature maps. and the ConvTranspose2D are used to upsample the feature maps. The ResNet blocks are used in the lower resolution feature maps to remap the representation of night images into a representation that is optimal for segmentation.

inference, if the source image belongs to the overcast-daytime category, we simply use the model M without any adaptations. However, if the source image comes from a different condition, we apply the corresponding adapter to ensure accurate and context-specific output. This approach could also be applied for other weather variation such as snow, rain ... etc.

The advantage of this approach is its ability to effectively manage various weather conditions and enhance the system's overall performance. Instead of retraining the entire model, we can train specialized adapters for each condition. Additionally, this method is versatile and can be applied to different tasks. For instance, the model M can be replaced with other modalities like depth or object detection, requiring only modifications to the objective function. Nevertheless, it is important to consider the computational cost associated with this method, particularly in the context of embedded systems such as autonomous driving.

3.2.1 Architecture

Fig. 3.4 shows the architecture of the adapter. The architecture of the adapters is based on a simple yet effective encoder-decoder design. With the aim of preserving the overall structure of the scene even when conditions change, skip-connections and a ResNet bottleneck are employed. This approach enables seamless feature transfer from the input side to the output side of the network. By incorporating skip-connections, we ensure that important features are preserved and propagated throughout the network. This combination of techniques not only facilitates direct feature transfer,

but also enhances the adaptability and robustness of the model.

3.2.2 End-to-end training

After reproducing the results, one improvement that could be proposed to improve the system is to train the pipeline in end-to-end manner. That is training the CycleGAN and the adapter at the same time.

Training the model in a two-phase approach is suboptimal due to the presence of artifacts, particularly hallucinations, introduced when the CycleGAN is trained separately. These artifacts undermine the consistency of the pair (day, synthetic-night), consequently impacting the quality of the adaptation results. In [12], we proposed a novel solution to address this challenge. We advocate for training the CycleGAN model alongside the adapter in an end-to-end manner. By incorporating the adapter within the training process, the CycleGAN model can benefit from additional supervision based on semantic information. This integration enables the entire system, composed of the CycleGAN and the adapter, to collectively enhance the performance of domain adaptation.

The end-to-end training approach holds several advantages. Firstly, it allows for tighter coordination between the CycleGAN and the adapter, fostering a more seamless integration of the two components. This integration ensures that the semantic information captured by the adapter is effectively utilized during the transformation process carried out by the CycleGAN. Secondly, the additional supervision provided by the adapter's semantic information can help guide the training of the CycleGAN model. This guidance plays a crucial role in reducing the occurrence of artifacts, such as hallucinations, which often arise when the CycleGAN is trained independently. By leveraging the semantic cues, the end-to-end training framework promotes more coherent and consistent translations between the day and synthetic-night domains.

The objective function becomes :

$$\mathcal{L} = \mathcal{L}_{adpt} + \mathcal{L}_{CycleGAN} \tag{3.2}$$

Where $\mathcal{L}_{adpt}$ is defined in Eq. 3.1 and $\mathcal{L}_{CycleGAN}$ is defined in Eq. 2.17.

3.2.3 Results

As it is defined in [133], to evaluate the performance of the proposed pipeline, an experiment was conducted using the RobotCar Dataset [112]. The experiment involved applying style-transfer techniques with cycle-consistency GAN generators to generate testing sequences specifically for night conditions. The groundtruth is obtained using the model M on day-images. the selected model M is DeepLabV3+ [21] The mIoU is used as metric.

	DeepLabV3+ [21] without adaptation	DeepLabV3+ [133] with adapter	Adapter with end-to-end training
mIoU on RobotcarDataset [112]	0.1850	0.5198	**0.5721**

Table 3.1: The table presents the results of adaptation and performance enhancement in semantic segmentation, specifically for nighttime images. The proposed adaptation technique demonstrates significant improvement in the model's performance on nighttime images. Additionally, the end-to-end training further enhances the results achieved.

Figure 3.5: Comparison of the qualitative results of the segmentation model with/without the adapter.

Table 3.1 represents the results of the adaptation, as observed. The domain adaptation that was proposed in [133] improves significantly the performance of the semantic segmentation on the night condition. The end-to-end approach improves further, and this results in a notable enhancement in overall accuracy. This suggests that the proposed domain adaptation technique effectively bridges the gap between the source and target domains, enabling the model to better handle the challenges posed by nighttime imagery. Moreover, Fig. 3.5 demonstrates the qualitative performance of the end-to-end trained model, showcasing the model's ability to accurately segment objects in low-light conditions. The segmentation outputs exhibit improved object consistency, sharper boundaries, and reduced noise, highlighting the effectiveness of the proposed adaptation method in enhancing semantic segmentation results.

3.3 Conditionality of Conditional GANs

In Figure 3.3, it can be observed that the translation from day to night is not fully consistent, as the model generates poles that are not present in the actual scene. The conditional GAN model, which was trained to generate night images, successfully captures the overall distribution of night scenes. However, it is not fully capable of representing night images conditioned on day images without hallucinating objects, resulting in inconsistencies in certain regions of the generated image. This issue is not unique to conditional GANs and is observed in other generative models across different domains, including language, even in lower-dimensional spaces [128, 129].

The conditionality of cGANs is at the crux of their theoretical contribution, and its impact

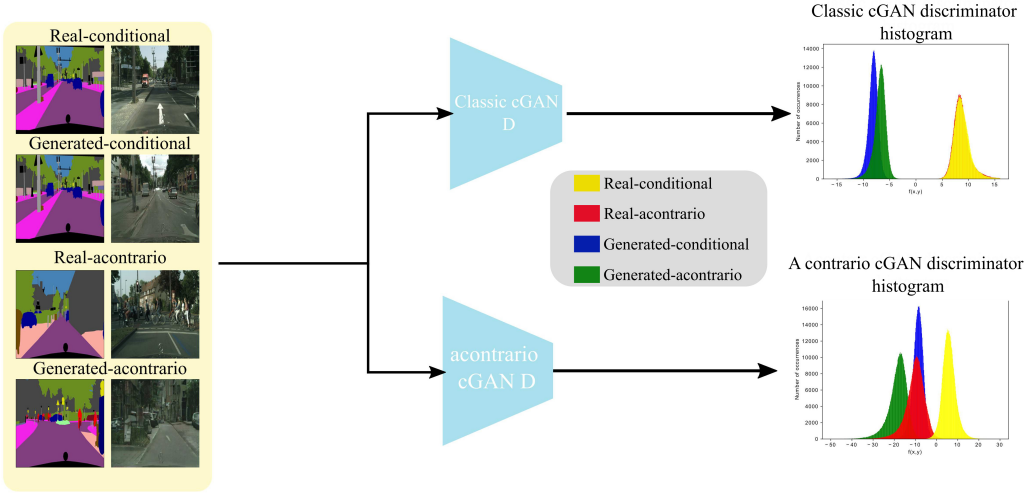


Figure 3.6: The classic cGAN and the proposed *a contrario* cGAN discriminators are tested with 500 validation images of the Cityscapes dataset on both conditional and unconditional label-to-image input set. Unconditional inputs (Real *a contrario* and Generated *a contrario*) set are obtained by randomly shuffling the original conditional sets of data. The classic cGAN discriminator fails to classify unconditional input set as false, as seen by the histogram distributions on the right (real *a contrario* in red is classified as true). The proposed method trains the discriminator with a general *a contrario* loss to classify an unconditional input set as fake (note that no extra training samples are required). The proposed *a contrario* cGAN correctly classifies all four modalities (blue, green, red, yellow) correctly.

therefore merits in-depth analysis. From the existing literature it is not clear, however, if this widely used architecture fully models conditionality. Empirically, the impressive results obtained with cGANs show that the generator automatically seeks to incorporate conditional variables into its generated output. Fundamentally, the generator is, however, free to generate whichever output as long as it satisfies the discriminator. Therefore, the conditionality of cGAN also depends on the conditionality of the discriminator. This begs the question as to whether or not the baseline architecture of cGANs explicitly models conditionality and if not, how can the core adversarial architecture be redefined to explicitly model conditionality? This is therefore the object of the following sections.

Problems with cGANs conditionality have been observed independently for different tasks in the literature. Label-to-image tasks observe that using only adversarial supervision yields bad quality results [155, 130, 170, 106]. "Single Label"-to-image tasks [15] observe class leakage. It is also well known that cGANs are prone to mode collapse [145]. In the following sections, it is suggested that all these problems are related to the lack of a conditional discriminator.

Consider a simple test of conditionality on a learned discriminator for the task of label-to-image

translation, shown in Figure 3.6. The conditional label input is purposely swapped with a non-corresponding input drawn randomly from the input set (*e. g.* labels). From this test, it is revealed that the discriminator does not succeed to detect the entire set of *a contrario* examples (defined in Section 3.3.3) as false input pairs. This suggests that the generator is not constrained by the discriminator to produce conditional output, but rather to produce any output from the target domain (street images in this case). Furthermore, in practice, the large majority of methods that exploit cGANs for label-to-image translation, if not all, add additional loss terms to the generator to improve conditionality. These loss terms are, however, not adversarial in their construction. For example, high resolution image synthesis approaches such as [170] suffer from poor image quality when trained with only adversarial supervision [155]. Considering the well known pix-to-pix architecture [71], a L1 loss applied to the generator was introduced to improve performance. This additional term seeks to enforce conditionality on the generator, but does not act explicitly on the discriminator. Subsequently, one could question if the conditionality obtained by such methods is obtained via this loss term, which is not part of the adversarial network architecture. Moreover, adding an extra loss term to the generator has now become the defacto method for improving cGANs results. For example, perceptual loss [74] and feature-matching [148] have been proposed and reused by many others[170, 24, 130, 126]. As demonstrated in the experiments, different tasks such as image-to-depth or image-to-label also exhibit these drawbacks.

In this chapter it will be argued that simply providing condition variables as input is insufficient for modelling conditionality and that it is necessary to explicitly enforce dependence between variables in the discriminator. It will be demonstrated that the vanilla cGAN approach is not explicitly conditional via probabilistic testing of the discriminator's capacity to model conditionality. With this insight, a new method for explicitly modelling conditionality in the discriminator and subsequently the generator will be proposed. This new method not only offers a solution for conditionality, but also provides the basis for a general data augmentation method by learning from the contrary (*a contrario* data augmentation).

3.3.1 Classic cGAN

Classical cGAN training is based on conditionally paired sets of data $\mathcal{C}(\mathbf{x}, \mathbf{y})$ where $\mathbf{x} \sim p(\mathbf{x})$ is the condition variable and $\mathbf{y} \sim p(\mathbf{y}|\mathbf{x})$ is the real training variable. The generator of a cGAN outputs a transformed set of data $\mathcal{C}_G(\mathbf{x}, \mathbf{y}_G)$ composed of the generator output variable $\mathbf{y}_G \sim p_G(\mathbf{y})$ and the condition variable. These sets of data will be called "real-conditional" and "generated-conditional" respectively. The discriminator is defined as:

$$D(\mathbf{x}, \mathbf{y}) := \mathcal{A}(f(\mathbf{x}, \mathbf{y})) \tag{3.3}$$

Where $f(.)$ is a neural network function of $\mathbf{x}$ and $\mathbf{y}$, and $\mathcal{A}$ is the activation function whose choice depends on the objective function. The cGAN objective function is defined as:

$$\mathcal{L}_{adv} = \min_G \max_D \big(\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}),\mathbf{y}\sim p(\mathbf{y}|\mathbf{x})}\big[log(D(\mathbf{x},\mathbf{y})]\big] + \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}\big[log[1 - D(\mathbf{x}, G(\mathbf{x}))]\big]\big) \tag{3.4}$$

The min-max activation function is defined as a Sigmoid $\mathcal{A}(\mathbf{x}) = \Big(\frac{1}{1+e^{-\mathbf{x}}}\Big)$.

3.3.2 Evaluating conditionality

The objective of this section is to propose methods to test the conditionality of cGAN networks. State-of-the-art approaches have focused on evaluating cGAN architectures with metrics applied to the generator output. Since the generator and discriminator are coupled, these metrics essentially evaluate the full GAN architecture.

A proposal is made to test the conditionality by visualizing the probability distribution at the output of the discriminator. Due to the fact that adversarial training involves a zero-sum game between a generator and a discriminator, both the generator and discriminator should seek to reach an equilibrium (Eq 2.10) at the end of training. One issue for GANs is that when the discriminator dominates, there is a vanishing gradient problem [2]. It is therefore more difficult (but not impossible) to isolate the discriminator during training to evaluate its capacity to detect unconditional examples as false. For this reason, an optimal discriminator can be used to give insight for evaluation purposes, as in [2, 45]. An optimal discriminator is essentially a binary classifier which classifies between true and fake (see Eq (3.4)).

In order to test the optimal discriminator, consider that the generator has been fixed after a certain number of iterations and the discriminator has been allowed to converge to an optimal solution based on the following objective function:

$$\max_{D\in\mathbb{D}} V(G_{fixed}, D) \tag{3.5}$$

The evaluation subsequently involves analyzing the distributions produced by the optimal discriminator (Eq. 3.5) given test distributions containing unconditional or *a contrario* sets of data-pairings. The capacity of the discriminator to correctly classify unconditional data as false is then analyzed statistically. Section 3.3.3 provides a formal definition of these unconditional data pairings. Probability distributions are visualized and evaluated by histogram analysis on the discriminator features in the last convolution layer.

3.3.3 A contrario conditionality loss

The proposed *a contrario* cGAN approach is based on training with unconditionally paired sets of data, obtained by randomly shuffling or re-paring the original conditional sets of data. The

a contrario set is defined as $\mathcal{C}_U(\tilde{\mathbf{x}}, \mathbf{y})$, where $\tilde{\mathbf{x}} \sim p(\mathbf{x})$ is the *a contrario* conditional variable ($\tilde{\mathbf{x}} \neq \mathbf{x}$) and $\mathbf{y}$ is the real training variable as in Section 3.3.1. In this case $\tilde{\mathbf{x}}$ and $\mathbf{y}$ are independent. The generator of the *a contrario* cGAN outputs a transformed set of data $\mathcal{C}_{UG}(\tilde{\mathbf{x}}, \mathbf{y})$ composed of the generator output variable $\mathbf{y}_G \sim p_G(\mathbf{y})$ and the random variable $\tilde{\mathbf{x}}$. For the purpose of this paper these two sets of data will be called "real-*a contrario* " and "generated-*a-contrario* " respectively. The motivation to create these new sets is to train the discriminator to correctly classify unconditional data as false. Figure 3.6 shows the four possible pairings. In practice, random sampling of *a contrario* pairs is carried out without replacement and attention is paid to not include any conditional variables into a same batch while processing.

In order to enforce conditionality between $\mathbf{y}$ and $\mathbf{x}$ an *a contrario* term is proposed as:

$$\mathcal{L}_{ac} = \max_D \left(\mathbb{E}_{\tilde{\mathbf{x}} \sim p(\tilde{\mathbf{x}}), \mathbf{y} \sim p(\mathbf{y})} \left[log(1 - D(\tilde{\mathbf{x}}, \mathbf{y})) \right] + \mathbb{E}_{\tilde{\mathbf{x}} \sim p(\tilde{\mathbf{x}}), \mathbf{x} \sim p(\mathbf{x})} \left[log(1 - D(\tilde{\mathbf{x}}, G(\mathbf{x}))) \right] \right) \tag{3.6}$$

The first term enforces the real-*a-contrario* pairs to be classified as fakes. The second terms enforce the generated-*a-contrario* as fake. The final loss is:

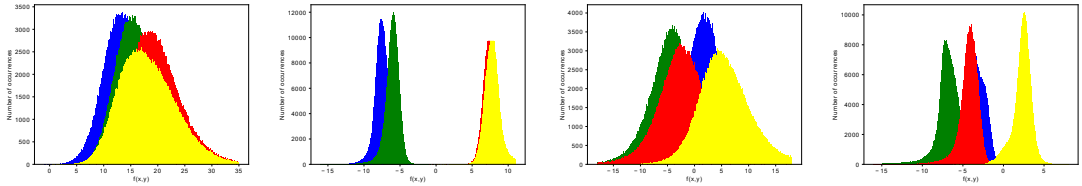
$$\mathcal{L}_{adv}^{'} = \mathcal{L}_{adv} + \mathcal{L}_{ac} \tag{3.7}$$

3.4 Experimental section

Several experiments will be presented that evaluate the conditionality of cGANs including: Monocular depth estimation on [124]; Real image generation from semantic masks on Cityscapes dataset [35]; "Single label"-to-image on CIFAR-10 [87]; Semantic segmentation using pix2pix on Cityscapes dataset. For label-to-image generation, pix2pix[71], pix2pixHD[170], SPADE[130] and CC-FPSE[106] were used to test the conditionality and to highlight the contribution of the *a contrario* cGAN with respect to more recent approaches. Depth estimation and image-to-label are structured prediction problems offer strong metrics for evaluating cGANs and various public datasets are available for training. While the scope of conditional evaluation has been limited to tasks that could provide a metric to evaluate both the conditionality and the quality of the generation, the proposed approach is general and not specific to these particular tasks. During training, the network's architecture, the additional losses, the hyper-parameters and data augmentation schemes are kept as in the original papers [71, 130, 170, 106]. The new additional *a contrario* term is the only difference between the compared methods.

3.4.1 Evaluating conditionality

Preliminary conditionality evaluation follows the method presented in Section 3.3.2 using *a contrario* sets to evaluate an optimal discriminator. In a first part, experiments were carried out on the vanilla



(a) (b) (c) (d)

Figure 3.7: Label-to-image histogram results when validating 500 Cityscape images on a discriminator trained until epoch 200. Blue is Generated-conditional, Green is generated *a contrario* , Red is real-*a contrario* , Yellow is Real-conditional. (a) The trained baseline discriminator, (b) Optimal baseline discriminator, (c) *a contrario* cGAN discriminator, (d) Optimal *a contrario* cGAN discriminator. (a) and (c) are still learning, this indicate that there is no vanishing gradient or mode collapse [2]. (b) doesn't detect conditionality since *a contrario* real is classified as true (red) (d) succeeds to classify all modes correctly.

pix2pix cGAN with a discriminator PatchGAN architecture with 70×70 receptive field. The model was trained on the Cityscapes dataset [35] for label-to-image translation with 2975 training images resized to 256×256. The pix2pix cGAN model is trained with the same hyperparameters as specified in the original paper [71]. The evaluation histogram is calculated on the values of the last convolution layer of the discriminator ($f(x, y)$ of Eq (3.3)) based on the 500 validation images. Each sample from the last convolutional layer is composed of a 30×30 overlapping patches with one channel.The proposed approach is trained in exactly the same manner, with the only difference being the new objective function.

Various tests were carried out to investigate the output distributions of each set of data for both the baseline architecture and the proposed method. The underlying accuracy of the implementation was first validated to ensure the accuracy reported in the original paper. A histogram analysis was then performed for different levels of training including: training for 20, 100, 200 epochs and evaluating after each. In another experiment the discriminator was allowed to continue to converge for one epoch after fixing 20, 100 and 200 epochs of cGAN training. In particular, training is performed with the objective given in Eq (3.5) and as proposed in [2, 45]. These results are plotted for each data pairing: real-conditional, generated-conditional, real-*a-contrario* and generated-*a-contrario* in Figure 3.7.

Figure 3.7 (a) and (c) show that, since the discriminator did not converge, the generator is still learning with no vanishing gradient or mode collapse. In Figure 3.7 (b) the discriminator has been allowed to reach an optimal value by fixing the generator. The real *a contrario* pairing is wrongly classified 99.9% of the time, indicating that the discriminator has not learned conditionality. (d) Shows clearly four distinct distributions and shows the ability of the proposed approach to learn conditionality and correctly classify real *a contrario* pairing 91.9% of the time. Similar conditionality tests were performed for various discriminator alternative architectures, including using a

Method	RMSE log	silog	$\log_{10}$	abs rel
baseline	0.3520± 0.0016	28.54± 0.1932	0.1247 ± 0.0005	0.3318± 0.0026
a contrario	**0.3036**± 0.0055	**23.51**± 0.1932	**0.1079** ± 0.0021	**0.2868** ± 0.0093

Table 3.2: Monocular Depth prediction experiments were repeated on the baseline and *a contrario* cGANs 6 times with different seeds. The mean and standard deviation are reported for each metric. The results shows that the *a contrario* cGAN outperforms the baseline on the depth metric [43].

separate/shared network for $\mathbf{x}$ and $\mathbf{y}$ and early/late/at-each-layer fusion. In all cases, conditionality was not learned.

These results strongly suggest that classic cGAN is unable to learn conditionality and that the spectacular results obtained by cGAN architectures are largely due to higher a level style constraints that are not specific to the input condition variable, since swapping condition variables produces no effect. The proposed histogram test allows to demonstrate the ability of the discriminator to classify the various underlying classes of data and shows their statistical distribution.

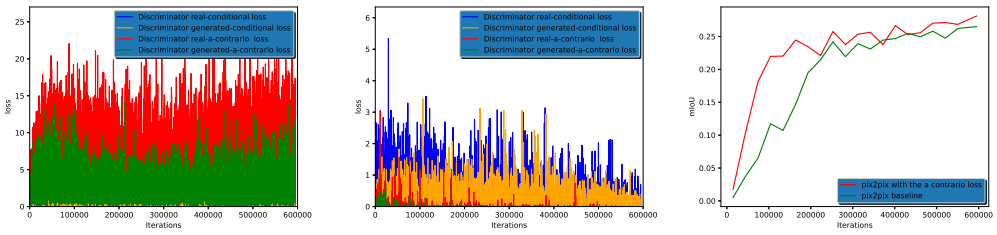
3.4.2 Image-to-depth

In this setting, the pix2pix model is trained on the NYU Depth V2 dataset [124] to predict depth from monocular 2D-RGB images only. The official train/validation split of 795 pairs is used for training and 694 pairs are used for validation. The dataset images are resized to have a resolution of 256×256. The experiment is repeated 6 times, and the mean and standard deviation are reported.

Table 3.2 shows the comparison of the two models across different metrics. Clearly, the *a contrario* cGAN reaches a better performance with log RMSE 0.3036 versus 0.3520 for the baseline (the mean is reported here). The qualitative results are shown in Figure 3.8.

For the classical cGAN, the discriminator is optimized only to distinguish real and generated samples, its decision boundary is independent of the conditional variable. The baseline cGAN architecture will not penalize the generation of outputs belonging to the target domain, but that do not correspond to the input (*i. e.* not conditional pair). Not only does this leave the generator with a larger search space (the generator is less efficient), but it can allow mode collapse, whereby the generator always produces the same output. The *a contrario* loss explicitly avoids this by penalizing unconditional generation. As observed in the qualitative results, both methods generate smooth and depth that resemble the distribution of an indoor depth. However, for the classical cGAN baseline, the depth output is not consistent to the conditioning input. The model hallucinate a room, it has the ability to freely invent an output that has the distribution of the room depth map. The acontario method enforces the conditionality explicitly, and it is able to generate an accurate and a consistent input-output.

Figure 3.8: Qualitative results for depth prediction. The *a contrario* cGAN shows better performance and more consistent prediction with respect to the input. The first row shows a case of mode collapse for the baseline as it ignores completely the input.



(a) (b) (c)

Figure 3.9: Comparison of the proposed approach on the Cityscape label-to-image training set. (a) The loss function for each set of data-pairing for the baseline cGAN method (*a contrario* are for evaluation only). (b) The loss function for each set of data-pairing for the *a contrario* cGAN method. (c) The evolution of the mIOU for both methods, performed on the validation dataset. It can be seen in (b) compared to (a) that the *a contrario* loss converges to 0 rapidly for the proposed approach. In (c) the proposed approach is much more efficient and converges much faster and with higher accuracy.

3.4.3 Label-To-Image translation

Generating realistic images from semantic labels is a well suited task to evaluate the effect of the *a contrario* at a high level, since many images can be potentially generated for each semantic class label. Figure 3.9(c) shows a comparison of the mIoU for the baseline pix2pix model and proposed pix2pix model with the additional *a contrario* loss. It can be observed that the *a contrario* cGAN converges faster than the baseline. The mIoU of the model with *a contrario* at iteration 163k is 24.46 whereas the baseline is 14.65. The mIoU oscillates around that value for the *a contrario* model, indicating that the model has converged. After 595k iterations, the mIoU for both models are very close 28.28 and 26.41. It is worth noting that evaluating using real images yields 29.6. The convergence is reported for the generator where the computational cost is exactly the same. the *a contrario* loss is specific only to the discriminator and adds a small computational cost. By restricting the search space of the generator to only conditional pairs, the generator's convergence is faster.

Table 3.3 shows a comparison of different architectures with and without *a contrario* augmentation. For a fair comparison, all the networks are trained from scratch and the same hyperparameter are used. The *a contrario* loss is the only difference between the two networks. The batch size for SPADE is 32 and 16 for CC-FPSE. Through explicitly enforcing the conditionality with *a contrario* examples, the discriminator learns to penalize unconditional generation achieving better results.

Moreover, Figure 3.9(a) and Figure 3.9(b) show the comparison of the losses of the discriminator for both models on this dataset. The baseline is trained with only conditional pairs. The *a contrario* data pairs are plotted to assess the ability of the discriminator to learn the conditionality automatically. The *a contrario* losses remain high for the baseline and converge to 0 for the proposed

Method	Resolution	FID	mIoU	Pixel accuracy (PA)
pix2PixHD	256×512	66.7	56.9	92.8
a contrario pix2pixHD	256×512	**60.1**	**60.1**	**93.2**
SPADE	256×512	65.5	60.2	93.1
a contrario SPADE	256×512	**59.9**	**61.5**	**93.7**
CC-FPSE	256×512	**52.4**	61.8	92.8
a contrario CC-FPSE	256×512	53.5	**63.9**	**93.5**

Table 3.3: A comparison of different architectures trained from scratch with and without *a contrario* augmentation. The networks with *a contrario* achieves better results with a mean improvement of $\Delta mIoU = +2.3$, $\Delta PA = +0.56$, and $\Delta FID = -3.8$.

a contrario cGAN. Figure 3.7 presented the histogram results for this experiment showed that the proposed approach better models conditionality

3.4.4 Single-label-to-image

The generality of the proposed *a contrario* cGAN can also be demonstrated by showing that it also improves architectures other than image-to-image. An example of a different task is conditioning the generated image on a single input class-label as in [15, 120, 127, 78, 77]. This different architecture is of interest because many new methods for improving cGANs are often tested on this task. Unfortunately, these methods are mainly evaluated on the FID [63] and IS [148] scores. As stated earlier, these metrics measure the quality/diversity and they favor models that memorise the training set [58]. They have not been designed to evaluate conditionality and therefore not sufficient for the purpose of this chapter. Despite that, these criteria are still important for evaluating the quality of GANs, however, an additional criterion is required for testing conditionality.

Here a simple conditionality test is proposed specifically for "single label"-to-image generation tasks based on a pretrained Resnet-56 [61] classifier trained on CIFAR-10 [87]. BigGAN [15] was selected as the baseline. Since BigGAN uses the Hinge-loss [99], the *a contrario* loss is adapted as follows:

$$\begin{aligned}\mathcal{L}_D = &-\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}),\mathbf{y}\sim p(\mathbf{y}|\mathbf{x})}\big[min(0,-1+D(\mathbf{x},\mathbf{y}))\big] - \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}\big[min(0,-1-D(\mathbf{x},G(\mathbf{x})))\big] \\ &-\mathbb{E}_{\tilde{\mathbf{x}}\sim p(\tilde{\mathbf{x}}),\mathbf{y}\sim p(\mathbf{y})}\big[min(0,-1-D(\tilde{\mathbf{x}},\mathbf{y}))\big] - \mathbb{E}_{\tilde{\mathbf{x}}\sim p(\tilde{\mathbf{x}}),\mathbf{x}\sim p(\mathbf{x})}\big[min(0,-1-D(\tilde{\mathbf{x}},G(\mathbf{x})))\big] \\ \mathcal{L}_G = &-\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}D(\mathbf{x},G(\mathbf{x}))\end{aligned} \tag{3.8}$$

Both models are trained from scratch on CIFAR-10 [87] dataset using the hyper-parameter specified in [15]. The conditionality is tested by generating 10k images for each label(100k images in total) and calculating the accuracy. The results[1]are shown in Table 3.4.

The conditionality improved significantly over the baseline with $\Delta Acc = +5.59$ and the quality

[1]The Pytorch IS and FID implementations were used for comparison

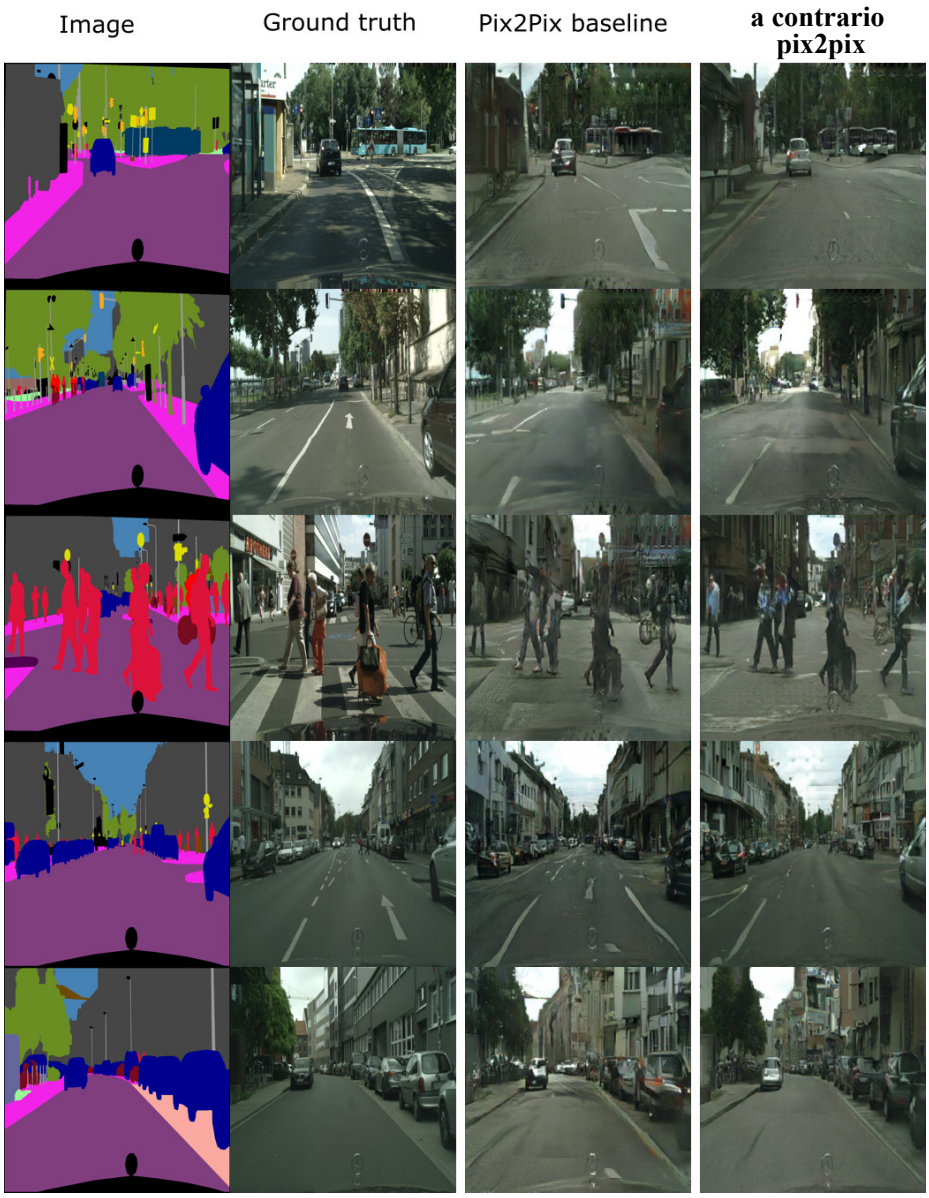


Figure 3.10: Qualitative results of Cityscapes label-to-image synthesis. In line with the quantitative results reported in Section 3.4.3, the qualitative results show better results for the *a contrario* in comparison to the baseline.

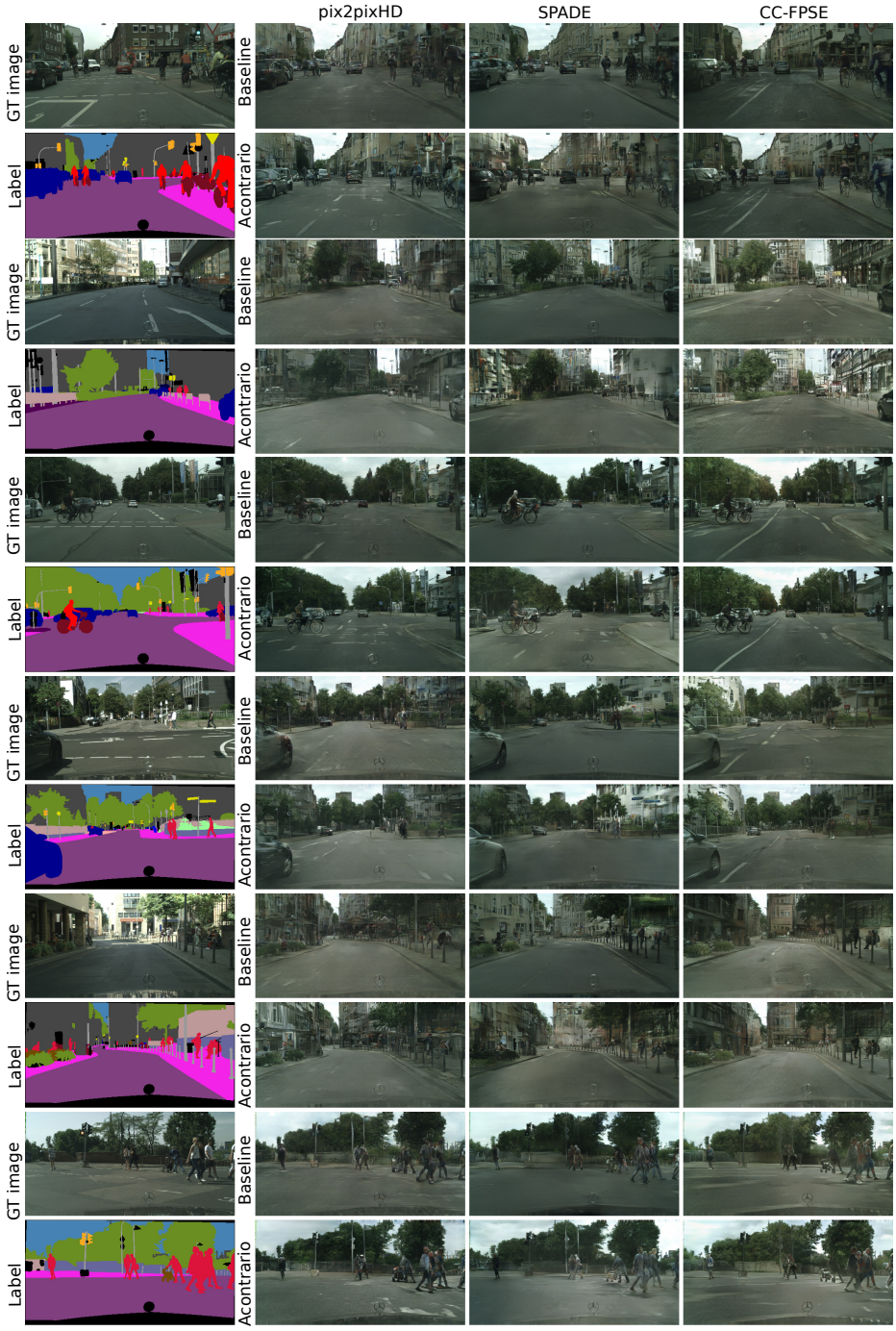


Figure 3.11: Qualitative comparison between different state-of-the-art methods for label-to-image trained and tested on Cityscapes[35] dataset. As observed, CC-FPSE baseline is the best baseline among classic cGAN. The *a contrario* improves all the baseline and the best model among the 6 models is *a contrario* CC-FPSE

Method	IS score	FID score	Acc
BigGAN[15]	8.26 ± 0.095	6.84	86.54
a contrario BigGAN	**8.40** ± 0.067	**6.28**	**92.04**

Table 3.4: A comparison of BigGAN [15] with and without the *a contrario* GAN. The network with *a contrario* achieves significantly better results with an improvement of $\Delta Acc = +5.59$, $\Delta IS = +0.14$, and $\Delta FID = -0.56$.

Method	Pixel accuracy (PA)	Mean Acc	FreqW Acc	mIoU
Baseline	66.12	23.31	53.64	15.97
a contrario	**72.93**	**26.87**	**60.40**	**19.23**

Table 3.5: Comparison on the Cityscapes dataset validation set. The proposed method consistently obtains more accurate results and finishes with a largely different score at the end of training with mIoU of 19.23 versus for the baseline 15.97.

also improved with $\Delta FID = -0.56$, $\Delta IS = +0.14$. Similar to the observation made before *a contrario* enforces the conditionality without compromising the quality. A failure mode of the lack of conditionality of the discriminator is class leakage : images from one class contain properties of another. While is it not easy to define a proper metric for such failure mode, it is shown that using the *a contrario* loss the classification was improved and therefore the generation is better constrained and does not mix class properties. This result shows that *a contrario* GAN also improves on a different SOTA task and confirms again that conditionality is an overlooked factor in current SOTA metrics.

3.4.5 Image-to-label segmentation

Image-to-label is a simpler task compared to depth prediction and label-to-image prediction as the goal of the generator is to transfer from a high-dimensional space to a lower-dimensional space. Furthermore, the evaluation is simpler since the image mask does not have multiple solutions and it is not necessary to use an external pre-trained segmentation network for comparison as in the case of label-to-image translation. It is worth mentioning that pix2pix is trained to output 19 classes as a segmentation network and is not trained as an image-to-image network as it is often done in cGAN architectures. FCN [109] trained on [71] obtains 21.0 mIoU. The performances are shown in Table 3.5. The training was unstable. However, the *a contrario* cGAN shows superior mIoU performance with 19.23 versus 15.97 for the baseline model. Figure 3.12 shows the qualitative results of the both models. It can be observed that the model baseline has invented labels that are not specified by the input. Training with *a contrario* helps the discriminator to model conditionality. Thus, the generator search space is restricted to only conditional space. The generator is penalized for conditionality even if the generation is realistic.

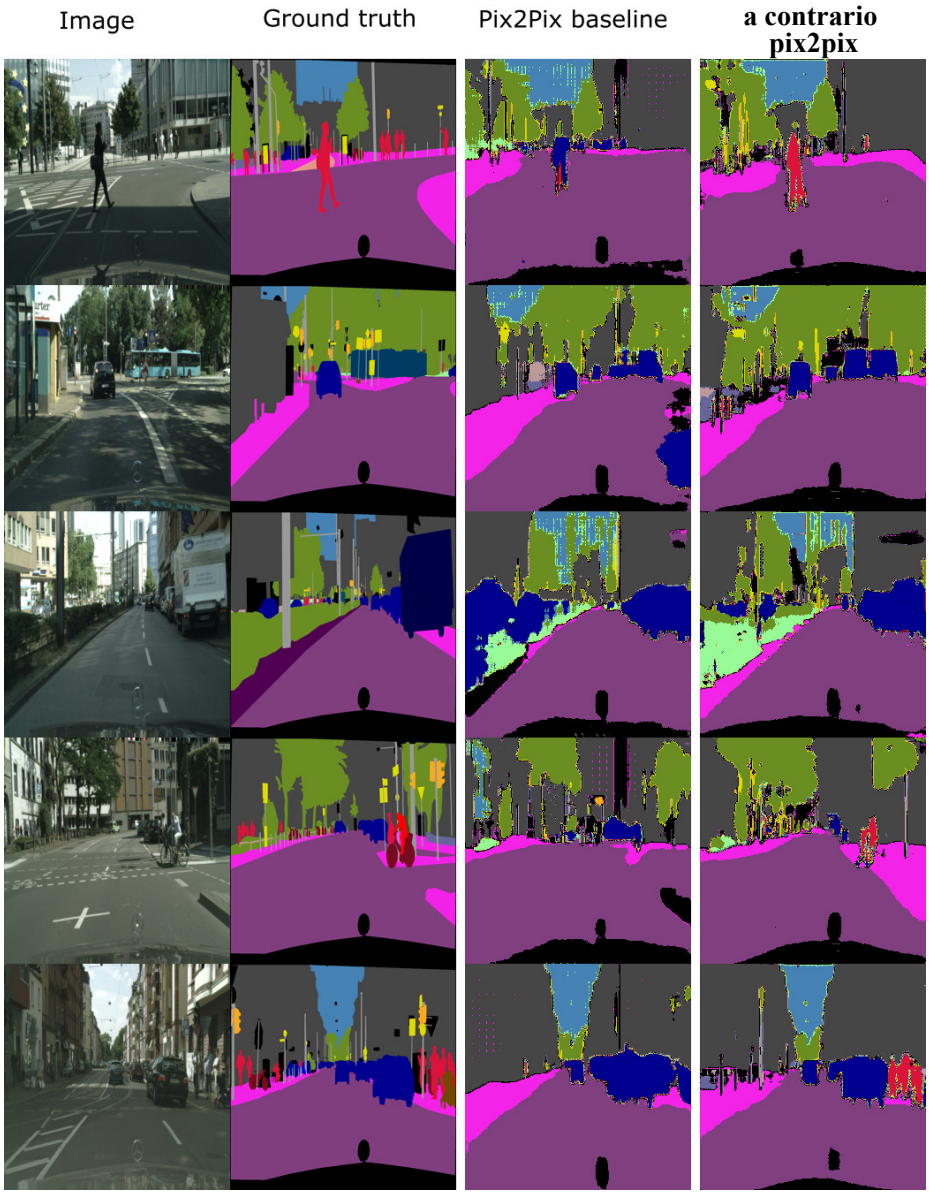


Figure 3.12: Qualitative results of Cityscape image-to-label task. It can be seen that the baseline model hallucinates objects. For instance, in the second row, the baseline hallucinates cars while the *a contrario* cGAN segments the scene better. In the first row, the baseline wrongly classifies the pedestrian as a car. While training the model, the discriminator does not penalize the generator for these miss-classifications

3.5 Discussion

This chapter has presented a new method called ***a contrario* cGAN**, which explicitly models conditionality for both parts of the adversarial architecture through a novel *a contrario* loss. This loss involves training the discriminator to learn unconditional (adverse) examples. The *a contrario* learning approach restricts the search space of the generator to conditional outputs using adverse examples. Extensive experimentation has demonstrated significant improvements across various tasks, datasets, and architectures.

One limitation of these models is their reliance on paired datasets. The requirement of paired data may limit the applicability of the models in scenarios where such data is not readily available. This is the case in domain adaptation, where often the input is unpaired dataset of source and target domain. However, this method could potentially be adapted for domain adaptation, it is important to note that it is not within the scope of this thesis, which focuses on performing depth prediction.

It is worth mentioning that even with recent advances in transformers and diffusion models, the problem of hallucination remains crucial, particularly in the development of generative large language models where accuracy is required. The proposed method in this thesis could potentially address this issue and provide insights for improving generative language models.

Moreover, it is important to acknowledge that inconsistency is not always a bug; it can be a desirable feature, especially in domains such as digital art where creativity is required. The ability of generative models to introduce controlled inconsistency can enhance their creative output.

Chapter 4

Image-to-depth inference

In the previous chapter, the objective of the thesis was approached with a broad perspective, exploring depth and other modalities and investigating generalization through domain adaptation. Building upon that, the following chapters delve into a detailed analysis of self-supervised depth prediction. The aim of this chapter is to develop a model capable of inferring depth from a single image using a self-supervised monocular approach, which poses a substantial challenge due to the inherent ambiguity in converting 2D images to 3D representations. The goal is to address a critical limitation in existing methods, which assumes a static scene. Many current approaches assume that the scene being captured remains unchanged over time with no significant object movements or variations. While this simplification facilitates depth training, it fails to account for the dynamics and temporal changes observed in real-world scenes.

To overcome the aforementioned limitation, an innovative approach is proposed in this chapter. The proposed approach relaxes the assumption of a rigid scene by inferring the pose of dynamic objects and compensating for their dynamics during model training. As a result, the performance of depth inference is enhanced. By incorporating the dynamic nature of scenes, this method represents a significant advancement in monocular self-supervised depth inference, thereby opening up possibilities for more advanced forecasting techniques.

The chapter starts by emphasizing the benefits and advantages of self-supervised learning. Utilizing unlabeled data through this approach proves advantageous due to its accessibility and cost-effectiveness compared to labeled data. The practicality of self-supervised learning makes it an attractive choice in different situations. The chapter offers a detailed introduction to self-supervised learning for depth prediction, covering problem formulation, self-supervision techniques. Lastly, the proposed method is presented, along with a discussion on the results, limitations, and future prospects.

This chapter is based on the following publication:

- **Journal paper:** Boulahbal Houssem Eddine, Adrian Voicila, and Andrew I. Comport. "Instance-aware multi-object self-supervision for monocular depth prediction." IEEE Robotics and Automation Letters 7.4 (2022): 10962-10968.
- **Conference paper:** Boulahbal Houssem Eddine, Adrian Voicila, and Andrew I. Comport. "Instance-aware multi-object self-supervision for monocular depth prediction." 2022 35th International Conference on Intelligent Robots and Systems (IROS). IEEE, 2022.

4.1 Supervised versus self-supervised approaches

In recent years, the field of deep learning has known a tremendous an exponential growth that has revolutionized the field of artificial intelligence. Since the unprecedented success of deep learning methods on the ImageNet [37], a plethora of expert models that can learn from massive amounts of labeled data were developed. Since then, there has been a rapid evolution of these models that have demonstrated impressive performance on a wide range of tasks. However, the performance and generalization of these models are heavily dependent on the quality, quantity, and diversity of the training data.

As illustrated in Fig. 4.1, the size of labeled dataset represent only a tiny portion of the unlabeled dataset, which in turn represent a tiny portion of the real world. Training only on the labeled dataset, yields models that are overly specialized, producing models that are susceptible to poor generalization. Furthermore, even if the model can successfully extrapolate beyond the labeled dataset, it will still only represent a small fraction of the possible scenarios that exist in the real world. Consequently, a model trained only on the labeled datasets is likely to suffer from biases, domain shifts, and poor generalization when confronted with extreme scenarios that were not part of the training dataset.

Despite the remarkable success of supervised learning in deep learning, there are limitations to the extent that AI can progress solely based on this approach. One of the most significant challenges facing supervised learning is the difficulty of obtaining and labeling large amounts of data, especially for real-world problems that are complex and diverse, such as autonomous driving applications. This obstacle necessitates the development of alternative approaches that can learn from directly from unlabeled data. Labeling everything is just impossible.

One inspiration to learn without labels is the human intelligence. humans have the ability to learn directly through observation: Human beings possess the ability to formulate hypotheses based on experiences, conduct experiments to test these hypotheses, observe the results, and ultimately derive a conclusion. Similarly, it is also possible to make machines learn solely from the data, where the learning obtains the supervisory signals from experience, *i. e.* data, only. This is known as self-supervision. One good example that illustrates the potential of these methods is the GPT [128] family. These models have been trained to predict the next token (word or image patch) in a

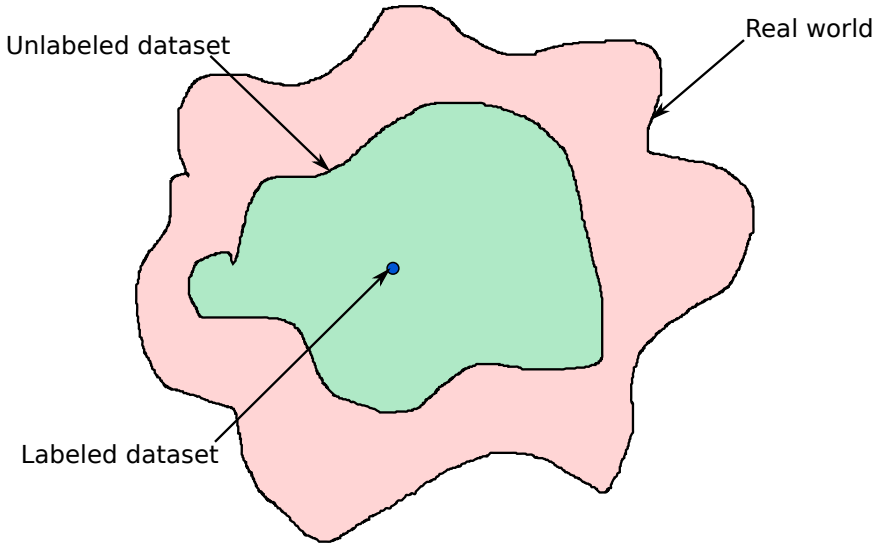


Figure 4.1: The size of labeled dataset represent only a tiny portion of the unlabeled dataset, which in turn represent portion of the real world

sequence, leveraging a huge dataset of text crawled from the internet, arXiv papers, and books. These self-supervised models have shown impressive results. They can even outperform supervised models on several tasks with zero-shot (without being trained explicitly to perform these tasks). Furthermore, some results suggest that the quality of self-supervised representations increase logarithmically in proportion to the volume of unlabeled pretraining data used [55]. This means that the performance may advance over time as advances in computational capacity and data acquisition enable ever-larger datasets to be utilized without the necessity of manually labeling new data.

As a summary, the reliance on labeled data in deep learning poses challenges for model performance and generalization. Limited labeled datasets result in specialized models with poor adaptability to real-world scenarios. Obtaining and labeling large amounts of data is difficult, hindering supervised learning progress. Self-supervised learning, inspired by human intelligence, offers a promising alternative. The following section delve into applying these methods on the depth modality.

4.2 Depth prediction with self-supervised methods

Self-supervised depth prediction refers to methods that only use images for input and supervision, without the need for ground-truth labels. These methods are becoming increasingly popular due to their practicality, as they do not require manually labeled training data.

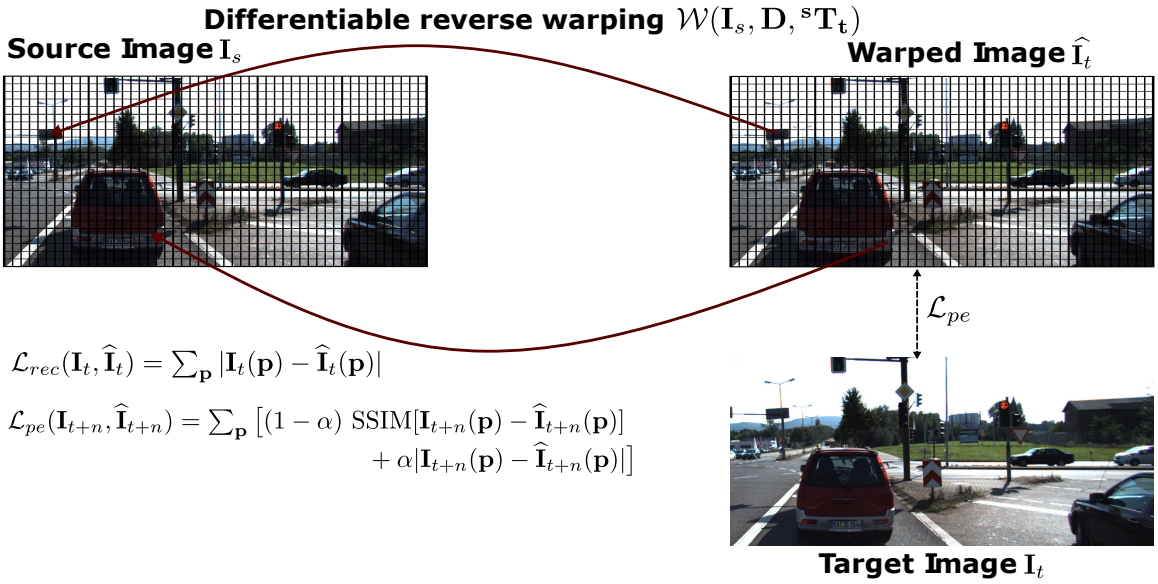


Figure 4.2: An illustration of the reverse (also called inverse) warping. Using the depth and the pose, the source image is warped into the target image. The self-supervised optimization is done using the photometric loss.

4.2.1 Problem formulation

The aim of monocular depth prediction is to learn an accurate depth map through the mapping $\mathbf{D}_t = f(\mathbf{I}_{t-k:t}; \boldsymbol{\theta})$ where $\mathbf{I}_{t-k:t} \in \mathbb{R}^{k \times W \times H \times 3}$ is k context images. $\mathbf{D}_t$ is the target depth. $\boldsymbol{\theta}$ are the network parameters. In self-supervised learning, this model is trained via novel view synthesis by reverse warping a set of source frames $\mathbf{I}_s$ into the target frame $\mathbf{I}_t$ using the learned depth $\mathbf{D}_t$ and the target to source pose ${}^s\mathbf{T}_t$. The reverse warping $\mathcal{W}$ is used to reconstruct the image. It involves mapping pixels from one image to another image, where the pixel coordinates in the new image are computed based on the differentiable warping function applied to the original image the mapping is defined as:

$$\widehat{\mathbf{p}}_s \sim \pi(\mathbf{K}\,{}^s\mathbf{T}_t H(\mathbf{D}_t \mathbf{K}^{-1} \mathbf{p}_t)) \tag{4.1}$$

Where H is the homogenous transformation operator and the π is the projection operator. For simplicity, these two operators are omitted:

$$\widehat{\mathbf{p}}_s \sim \mathbf{K}\,{}^s\mathbf{T}_t \mathbf{D}_t \mathbf{K}^{-1} \mathbf{p}_t \tag{4.2}$$

The image is reconstructed using the interpolation. Fig. 4.2 shows the warping process. The points of the target image are back-projected using the camera parameters and the learned depth. The ${}^s\mathbf{T}_t$ is applied to transform the point cloud. Finally, the point cloud is projected using the camera parameters. The target image is obtained using the interpolation. Therefore, by knowing the depth

and pose, the mapping from the image $\mathbf{I}_s$ is used to reconstruct $\hat{\mathbf{I}}_t$ through a bi-linear interpolation.

4.2.2 Loss functions

Let $\mathcal{L}$ be the objective function. The self-supervised setting casts the depth learning problem to an image reconstruction problem through the reverse warping. Thus, learning the parameters $\boldsymbol{\theta}$ involves learning $\widehat{\boldsymbol{\theta}} \in \boldsymbol{\Theta}$ that minimizes the following objective functions:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\operatorname{argmin}} \frac{1}{n} \sum_{n} \mathcal{L}(f(\mathbf{I}_t, \widehat{\mathbf{I}}_t; \boldsymbol{\theta})) \tag{4.3}$$

where n is the number of training examples. There are several surrogate loss functions proposed to supervise the depth though image reconstruction, some of these losses are :

- **Photometric loss:**

  Following [189, 50, 141], the photometric loss seeks to reconstruct the target image by warping the source images using the static/dynamic pose and depth. The L_1 loss is defined as follows:

  $$\mathcal{L}_{rec}(\mathbf{I}_t, \widehat{\mathbf{I}}_t) = \sum_{\mathbf{p}} |\mathbf{I}_t(\mathbf{p}) - \widehat{\mathbf{I}}_t(\mathbf{p})| \tag{4.4}$$

  where $\widehat{\mathbf{I}}_t(\mathbf{p})$ is the reverse warped target image obtained by Eq. 4.2. The L1 loss is a widely used loss function in computer vision tasks, and it is particularly useful for self-supervised depth prediction because it is robust to outliers. However, the L1 loss alone is not sufficient, as it does not take into account the structural similarity between the predicted image map and the ground-truth image.

- **SSIM (Structural Similarity Index)** is used to improve the photometric loss for self-supervised depth prediction models. SSIM is a widely used metric for image quality assessment, and it measures the structural similarity between two images by comparing the luminance, contrast, and structure of the images. SSIM is particularly useful for self-supervised depth prediction because it is more sensitive to changes in the structure of the images than the L1 loss.

  Therefore, the photometric loss is defined as:

  $$\begin{aligned} \mathcal{L}_{\text{pe}}(\mathbf{I}_t, \widehat{\mathbf{I}}_t) = \sum_{\mathbf{p}} \big[& (1-\alpha)\ \text{SSIM}[\mathbf{I}_t(\mathbf{p}) - \widehat{\mathbf{I}}_t(\mathbf{p})] \\ & + \alpha |\mathbf{I}_t(\mathbf{p}) - \widehat{\mathbf{I}}_t(\mathbf{p})| \big] \end{aligned} \tag{4.5}$$

- **Depth smoothness:** An edge-aware gradient smoothness constraint is used to regularize the photometric loss. The depth map is constrained to be locally smooth through the use of

an image-edge weighted L_1 penalty, as discontinuities often occur at image gradients. This regularization is defined as [62]:

$$\mathcal{L}_s(D_t) = \sum_p \big[|\partial_x D_t(\mathbf{p})| e^{-|\partial_x \mathbf{I}_t(\mathbf{p})|} + |\partial_y D_t(\mathbf{p})| e^{-|\partial_y \mathbf{I}_t(\mathbf{p})|}\big] \tag{4.6}$$

In practice, to optimize the self-supervised depth prediction network, often, a L1 loss is used in combination with the SSIM and depth smoothness losses. This yields better results than using a single loss function alone.

4.3 Dynamic object for self-supervised depth

Self-supervised monocular depth training methods presented in Sec. 4.2 are based on the assumption of a rigid scene, meaning that the scene is static and the camera is moving. However, this assumption is often violated in real-world scenarios due to the presence of moving objects in the scene. This poses a challenge for depth inference models, as the motion of objects can significantly impact the accuracy of depth inference. One potential solution to this issue is to mask the dynamic objects' points in the scene during training. This can be achieved through various methods such as: learned masking techniques [189], semantic guidance [85] or auto-masking [50, 173]. However, these methods only provide a workaround to the problem of non-rigid scenes, and they fail to utilize the information from moving objects that could be useful for further constraining depth inference. To address the challenge of moving objects in the scene, various studies have proposed methods that explicitly incorporate information about moving objects into the depth inference models. For example, some studies have proposed methods that learn a per-object semantic segmentation mask and a motion field that account for the motion of the objects in the scene [163, 91, 177]. Other studies have relied on optical flow to model the motion of objects in the scene [140, 182]. While these methods are optimized for local rigidity, they do not take into account the different dynamics of different object classes. As a result, they may not provide as accurate depth inference as the methods that explicitly model the $6-$DOF motion of objects.

A proposition is made here to alleviate this assumption. Non-rigid scenes are learned by factorizing the motion into the dominant ego-pose and **a piece-wise rigid pose for each dynamic object** explicitly. Therefore, for static objects, only the ego-pose is used for the warping, while the dynamic objects are subject to two transformations using the motion of the camera and the motion of each moving object. The proposed method explicitly models the motion of each object, allowing accurate warping of the scene elements.

In order to model the object motion in the scene, the proposed method makes use of the multi-head attention of the transformer network that matches moving objects across time and models

their interaction and dynamics. This enables accurate and robust pose estimation for each object instance. The proposed method achieves SOTA results on the KITTI benchmark. In summary, the contributions of the method proposed in this chapter are:

- A transformer-based network architecture that utilizes multi-head attention to match moving objects across time and accurately estimate their motion, enabling more robust and precise pose estimation for each object instance.
- An accurate and robust per-object pose is obtained by matching and modeling the interaction of the objects across time.
- High quality depth inference, achieving competitive performance with respect to state-of-the-art results on the KITTI benchmark [49].
- The demonstration that the KITTI benchmark has a bias favoring static scenes, and a method to test the quality of moving object depth inference.

4.3.1 Related work

Supervising the depth with a photometric loss is problematic when moving objects are present in the scene. This challenge has gained attention in the literature, a common solution is to disentangle the dominant ego-motion and the object motion. [29, 140, 182, 69] leverage an optical flow network to detect moving objects by comparing the optical flow with depth-based mapping. [92] learns a monocular depth in order to estimate the motion field as two stage learning. [163] learns a per-object semantic segmentation mask and a motion field is obtained by factorization of the motion of each mask and the ego-motion. [147] addresses the object motion without additional labels by proposing a scene decomposition into a fixed number of components where a pose is inferred for each component. [177] relaxes the problem using local rigidity within a predefined window, and the motion of each window is predicted to account for moving objects. [111] leverages the geometric consistency of depth, ego-pose and optical flow and categorises each pixel as either rigid motion, non-rigid/object motion or occluded/non-visible regions. A recent work that is closest to the proposed method is Insta-DM [91]. In that method, the source and target images are masked with semantic masks and an object PoseNet is used to learn the pose from the masked RGB images. Alternatively, the method proposed in this chapter factorizes the motion into ego-motion and object-motion and exploits a transformer attention network to perform instance segmentation and learn a per-object motion.

4.3.2 Problem formulation

In the method proposed in this chapter, rather than enforcing the rigid scene assumption, a proposition is made to alleviate this assumption. For each pixel, a **global rigid-scene pose** and a

piece-wise rigid pose for each dynamic object is learned. This is more precise and consistent with the non-rigid real-world situations. An instance segmentation network [121] is extended to incorporate the pose information so that the network learns an additional 6-DOF pose for each instance. Therefore, each instance i is represented by the class c^i, bounding box $\mathbf{B}^i$, mask $\boldsymbol{\mathcal{M}}^i$ and the additional pose $\mathbf{T}_o^i \in \mathbb{SE}[3]$ as illustrated in Fig. 4.3. The per-instance warping is defined as:

$$\widehat{\mathbf{p}}_s \sim \mathbf{K} \sum_{i=0}^{m} \left[\mathcal{M}^i_{\mathbf{p}_t} \mathbf{T}_o^i + (1 - \boldsymbol{\mathcal{M}}^i_{\mathbf{p}_t}) \boldsymbol{I}_4 \right] {}^s\mathbf{T}_t \mathbf{D}_t \mathbf{K}^{-1} \mathbf{p}_t \tag{4.7}$$

For simplicity, the homogenous and projection operator are omitted. m is the number of dynamic object instances and $\boldsymbol{I}_4$ a 4×4 identity matrix. For simplicity, the homogeneous pose and projection transformations are omitted in Eq. 4.7. The mask $\mathcal{M}^i$ is used to transform only the dynamic object i with its pose $\mathbf{T}_o^i$. Rigid scene points are transformed only with the pose ${}^s\mathbf{T}_t$. Using the Eq. 4.7, the image $\widehat{\mathbf{I}}_t$ is obtained by inverse warping.

4.4 Method

4.4.1 Architecture

In order to explicitly model the motion of the moving objects, an instance pose head is introduced into an instance segmentation network. EfficientPS [121] has demonstrated SOTA results for panoptic and instance segmentation and is therefore adopted in this method for depth inference. It consists of the EfficientNet backbone [157], BiFPN [158], MaskRCNN instance segmentation head [60] and the DPC [20] semantic head. The EfficientNet backbone has demonstrated its success as a task agnostic feature extractor for nearly all vision tasks. It is easily scalable allowing more complexity/FLOPS trade-off. The BiFPN allows low-level and high-level feature aggregation, thus, enabling a rich representation that accounts for the fine-details and more global abstraction at each feature map. During training, the FPN features $(P_4, P_8, P_{16}, P_{32})$ are extracted for the source and target frames. The two pose heads use both source and target features, while the instance, semantic, and depth heads use only the target features. The model architecture is shown in Fig. 4.3. The additional heads are detailed in the following.

Instance pose head

The key idea of the proposed method is to factorize the motion by explicitly estimating the $6-$DOF pose of each object in addition to the dominant ego-pose. In order to accurately estimate this motion, the objects should be matched and tracked temporally and its interaction should be modeled. Inspired by the prior work on object tracking [117, 179], a novel instance pose head that extends the instance segmentation is proposed using transformer module [162]. This head makes use of the

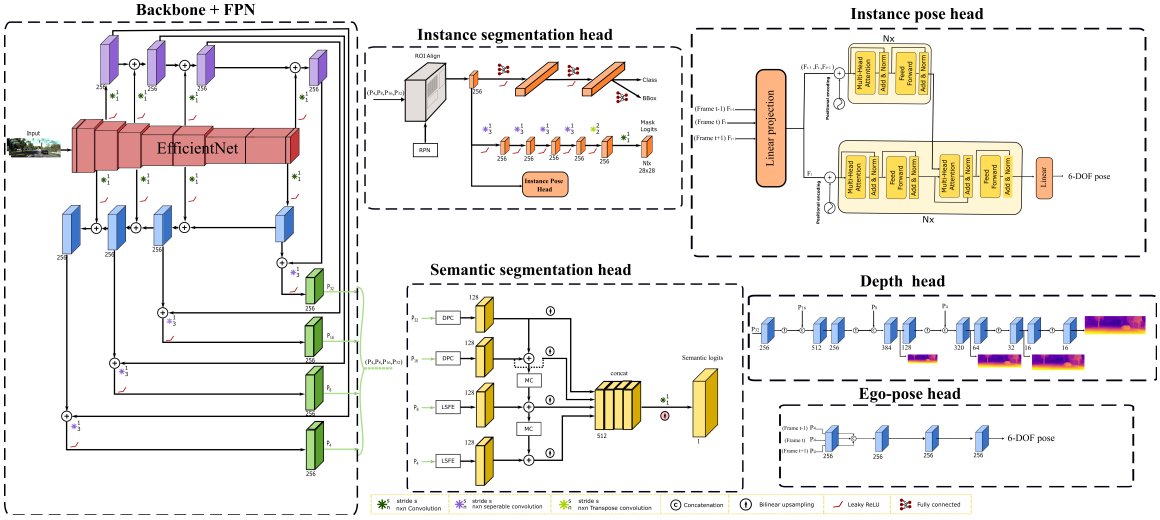


Figure 4.3: The proposed model architecture consisting of the EfficientNet backbone [157], BiFPN [158], the DPC [20] semantic head, the MaskRCNN instance segmentation head [60], the novel instance pose head, an ego-pose head and a depth head. During training, the FPN features $(P_4, P_8, P_{16}, P_{32})$ are extracted for the source $\mathbf{I}_t$ and target frames $\mathbf{I}_{t-1}$, $\mathbf{I}_{t+1}$. These features are pooled using the proposals of the RPN and the ROI Align modules. The class, bounding box and instance mask heads use only the features of frame $\mathbf{I}_t$. The Instance pose head uses both source and target frames as input. This head output a 6 axis-angle parameters for each instance. Similarly, the ego-pose head uses the both source and target frames P_4 FPN' features as input. This head output a 6 axis-angle parameters for the ego-pose. The depth head input the FPN features of the source frame $\mathbf{I}_t$ and output a multiscale depth.

multi-head attention to learn the association and interaction of the object across time.

The RPN network yields N proposals. The features of each proposal are pooled using a ROI Align module. These features are extracted for the three frames. Therefore, the input of the instance pose head is $b \times (s+1) \times N \times 256 \times 14 \times 14$. Where b is the batch size and s is the number of sources images. The first operation is to project these features into the transformer embedding. The linear projection layer flattens the 3 last dimensions and a linear layer is used to learn an embedding of each proposal. This mapping is defined as $\textit{Linear projection} : \mathbb{R}^{B\times(s+1)\times N\times 256\times 14\times 14} \rightarrow \mathbb{R}^{B\times(s+1)N\times 512}$.

The input of the encoder-decoder transformer is a $(s+1)N$ sequence with 512 features. The transformer-encoder multi-head attention enables the matching of target frame proposals with respect to the source proposals across time, while the feed-forward learns the matched-motion features. For the transformer-decoder, only the target proposals are used for input. The multi-head attention aggregates the matched-motion features of the encoder to the target proposals and further learns the interactions of the objects by learning an attention between the proposals. Finally, a linear layer is used to infer the $6-$DOF pose per object, yielding $B \times N \times s \times 6$ using a 6 axis-angle convention parameters. The non-maximum-suppression used for the object detection head is employed to filter the $N = 1000$ object proposals, keeping only the relevant objects. The object pose is inferred only for the filtered objects. Non-maximum suppression (NMS) is a post-processing technique commonly used in object detection algorithms. It helps eliminate redundant and overlapping bounding box predictions to generate a more concise and accurate set of detections. NMS works by comparing the confidence scores of neighboring bounding boxes and suppressing those that have a significant overlap and lower confidence, keeping only the most confident and non-overlapping detections. This process ensures that only the most relevant and highest-scoring object instances are retained while removing redundant or duplicate predictions.

Ego-pose branch

The ego-pose branch estimates the dominant pose of the camera. Since the low-level features that allow matching are usually extracted in the first layers, the P_4 features of the FPN for source and target features are used. The pose decoder is composed of 4 consecutive convolution with kernels of $k = 3$ and the output channels of these 4 convolutions are $256, 256, 256, 6\times$num_frames_to_predict_for. Since in this experiment the pose is predicted for $t-1$ and $t+1$, num_frames_to_predict_for $=$ 2. Therefore, this network outputs 6 parameters for the pose transformation using the axis-angle convention.

Depth branch

The depth branch consists of convolution layers with skip connections from the FPN module as in [50]. Similar to prior work [189, 50, 173], a multiscale depth is estimated in order to resolve the issue of gradient locality. The inference of depth at each scale consists of a convolution with a kernel

Method	Supervision	Resolution	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
SfMlearner [189]	M	640×192	0.183	1.595	6.709	0.270	0.734	0.902	0.959
GeoNet [182]	M+F	416×128	0.155	1.296	5.857	0.233	0.793	0.931	0.973
CC [140]	M+S+F	832×256	0.140	1.070	5.326	0.217	0.826	0.941	0.975
Self-Mono-SF[69]	M+F	832×256	0.125	0.978	4.877	0.208	0.851	0.950	0.978
Chen *et al* [22]	M+S	512×256	0.118	0.905	5.096	0.211	0.839	0.945	0.977
Monodepth2 [50]	M	640×192	0.115	0.903	4.863	0.193	0.877	0.959	0.981
Lee *et al* [92]	M+F	832×256	0.113	0.835	4.693	0.191	0.879	0.961	0.981
SGDepth [85]	M+S	1280×384	0.113	0.835	4.693	0.191	0.879	0.961	0.981
SAFENet [34]	M+S	640×192	0.112	0.788	4.582	0.187	0.878	0.963	0.983
Insta-DM [91]	M+S	640×192	0.112	0.777	4.772	0.191	0.872	0.959	0.982
PackNetSfm [141]	M	640×192	0.111	0.785	4.601	0.189	0.878	0.960	0.982
MonoDepthSeg [147]	M	640×192	0.110	0.792	4.700	0.189	0.881	0.960	0.982
Johnston *et al* [76]	M	640×192	<u>0.106</u>	0.861	4.699	0.185	<u>0.889</u>	0.962	0.982
Manydepth [173]	M+TS	640×192	**0.098**	<u>0.770</u>	**4.459**	**0.176**	**0.900**	**0.965**	<u>0.983</u>
Ours	M+S	640×192	0.110	**0.719**	<u>4.486</u>	<u>0.184</u>	0.878	<u>0.964</u>	**0.984**

Table 4.1: Quantitative performance comparison of on the KITTI benchmark with Eigen split [49]. For Abs Rel, Sq Rel, RMSE and RMSE log, lower is better, and for $\delta < 1.25$, $\delta < 1.25^2$ and $\delta < 1.25^3$ higher is better. The Supervision column illustrates the training modalities: (M) raw images (S)Semantic, (F) optical flow, (TS) Teacher-student. At test-time, all monocular methods (M) scale the estimated depths with median ground-truth LiDAR.The best scores are bold and the second are underlined

of 1×1 and a Sigmoid activation. The output of this activation, σ, is re-scaled to obtain the depth $D = \frac{1}{a\sigma+b}$, where a and b are chosen to constrain D between 0.5 and 100 units, similar to [50].

To maintain a self-supervised learning setting, a frozen pretrained EffiecientPS that was trained on the Cityscapes benchmark [35] is used. This pretrained model achieves $PQ = 50.2$ and $SQ = 76.8$ (see Sec. 2.4 for metric definition) on Cityscapes test benchmark. As the representation that was trained for panoptic segmentation may ignore details that are crucial for depth inference. A duplicate of the Backbone and FPN is used for the depth and pose heads. This allows learning features optimized for depth inference without degrading the performances of the panoptic segmentation heads.

The objective function, denoted by $\mathcal{L}$, was previously defined in Sec. 4.2.2. It involves minimizing a combination of two losses: the photometric loss ($\mathcal{L}pe$) and the depth smoothness loss ($\mathcal{L}s$). The final objective function is given by $\mathcal{L} = \mathcal{L}pe + \alpha_d \mathcal{L}s$, where α_d is a hyperparameter controlling the trade-off between the two losses.

4.5 Experiments

4.5.1 Setup

- KITTI benchmark [49]: Following the prior work [189, 181, 173, 50, 167], the Eigen *et al* [44] split is used with Zhou *et al* [189] pre-processing to remove static frames. For evaluation, the common metrics used in the KITTI benchmark will be used (see in CHSec. 2.2.6 for more details).

- Implementation details: PyTorch [131] is used for all the models. The networks are trained for 40 epochs and 20 for the ablation, with a batch size of 2. The Adam optimizer [83] is used with a learning rate of $lr = 10^{-4}$ and $(\beta_1, \beta_2) = (0.9, 0.999)$. The exponential moving average of the model parameters is used with the $decay = 0.995$. As the training proceeds, the learning rate is reduced at epoch 15 to 10^{-5}. The SSIM weight is set to $\alpha = 0.15$ and the smoothing regularization weight to $\alpha_d = 0.001$. The depth head outputs 4 depth maps. At each scale, the depth is up-scaled to the target image size. The hyperparameters of EfficientPS are defined in [121] with $N = 1000$ before the Non-maximum-suppression. Two source images $\mathbf{I}_{t-1}$ and $\mathbf{I}_{t+1}$ are used. The input images are resized to 192×640. Two data augmentations were performed: horizontal flips with probability $p = 0.5$ and color jitter with $p = 1$.

4.5.2 Results

During the evaluation, the depth is capped to 80m. To resolve the scale ambiguity, the inferred depth map is multiplied by the median scaling. The results are reported in Table 4.1. The proposed method achieves competitive performances compared to the state-of-the-art (SOTA) and outperforms [173] with respect to the *Sq Rel* with an improvement of 6.62%. As expected, the proposed method is superior to the prior works that factorize the motion using the optical flow [140, 182] as their estimated motion is only local, it does not take into account the class of the object. Besides, it outperforms other similar methods [91] that factorize using the pose for each object. Fig. 4.4 illustrates the qualitative result comparison. As observed, the proposed method enables high quality depth inference. Compared to the SOTA methods, The method proposed during this thesis is able to represent well the dynamic objects. As the network did not mask the dynamic objects during training, the dynamic objects are better learned compared to the methods that masks the dynamic objects [173, 141].

Dynamic and static evaluation

In contrast to training, where the points are categorized into moving and static-object points, testing is performed on all points that have Lidar ground truth. This does not take into account the relevance of the points and the static/dynamic category. Moving objects are crucial for autonomous driving applications. However, with this testing setup, it is not possible to convey how the model performs on moving objects, especially for methods that masks moving objects during training. This begs the question of whether or not a model trained with a rigid scene assumption learns to represent the depth of dynamic objects even when it is trained with only static objects?

In order to address this question, the performances of the different methods are evaluated separately with respect to static and dynamic motions. A mask of dynamic objects is used to segment moving objects, and the assessment can be carried out on each category separately. To avoid biasing the evaluation with the EffiecientPS mask, the evaluation mask is obtained using an independent

MaskRCNN [60] trained with detectron2 [176]. The first observation that could be made is that the static objects represent 86.43% of test points. This suggests that using the mean across all points will bias the evaluation towards the static objects. A better solution is to consider the per static/dynamic category mean. Table 4.2 illustrates the evaluation of the method versus the current SOTA method video-to-depth inference [173]. The proposed method outperforms the SOTA [173] for the dynamic objects with a large difference $\Delta Sq\ Rel = -0.698m$ while the gap for the static objects is only $\Delta Sq\ Rel = +0.011$. The results show that degradation induced by considering the rigid scene assumption is significant. This exposes the limitation of the prior evaluation methods. The KITTI benchmark is biased towards static scenes. In order to unbias the evaluation, the mean per-category is used to balance the influence. The proposed method outperforms the video-to-depth inference method [173] with $\Delta Sq\ Rel = -0.344m$. The analysis of Table 4.2 and Fig. 4.4 suggests that models with rigid scene assumption are still able to infer a depth for moving objects (probably due to the depth smoothness regularization and stationary cars), however, its quality is significantly degraded when compared to the static objects.

Moreover, the results reported in Table 4.2 show that the proposed method outperforms Insta-DM [91] with respect to both the static and dynamic objects. Insta-DM [91] proposes an Obj-PoseNet $\mathcal{O}_\psi : \mathbb{R}^{2\times H\times W\times 3} \rightarrow \mathbb{R}^6$ that takes per-object matched binary instance masks $(\mathbf{M}_1, \mathbf{M}_2)$ and outputs the object pose. It should be noted, however, that the Insta-DM has an unfair advantage since object matching (via binary masks) is provided as input in a supervised learning approach while the proposed method is self-supervised with matching being implicitly learned in the network. Even so, the proposed method still yields better results on average with respect to dynamic objects.

Ablation study

Table 4.3 illustrates an ablation study performed to validate the contribution of the proposed method. The results strongly suggest that the performance of the proposed network is mainly obtained by the introduction of the motion factorization through the proposed instance pose head.

- **A1 versus A4:** Introducing a more complex architecture did not contribute to the improvement of the performances.
- **A4 versus A5:** Sharing the backbone for the depth network did not contribute to the improvement of the performances. However, it did reduce the complexity of the network.
- **A5 versus A6:** Introducing the piece-wise rigid pose warping induces an improvement of $\Delta Sq\ rel = 14.1\%$
- **A2 versus A3 versus A4:** The pose head is sensitive to the choice of the features level. P_4 is the optimal level for this application.

Evaluation	Model	Abs Rel	Sq Rel	RMSE	RMSE log
All points mean	ManyDepth [173]	**0.098**	0.770	**4.459**	**0.176**
	Insta-DM [91]	0.112	0.777	4.772	0.191
	Ours	0.110	**0.719**	4.486	0.184
Only dynamic	ManyDepth [173]	0.192	2.609	7.461	0.288
	Insta-DM [91]	**0.167**	**1.898**	6.975	0.283
	Ours	**0.167**	1.911	**6.724**	**0.271**
Only static	ManyDepth[173]	**0.085**	**0.613**	**4.128**	**0.150**
	Insta-DM [91]	0.106	0.701	4.569	0.171
	Ours	0.101	0.624	4.269	0.163
Per category mean	ManyDepth[173]	0.139	1.611	5.794	0.219
	Insta-DM [91]	0.137	1.299	5.772	0.227
	Ours	**0.134**	**1,267**	**5,496**	**0,217**

Table 4.2: Quantitative performance comparison for dynamic and static objects. The proposed method outperforms the SOTA [173] that uses masking for the dynamic objects with a significant gap $\Delta Sq\ Rel = -0.698m$. In addition, it outperforms Insta-DM [91] which explicitly models dynamic objects.

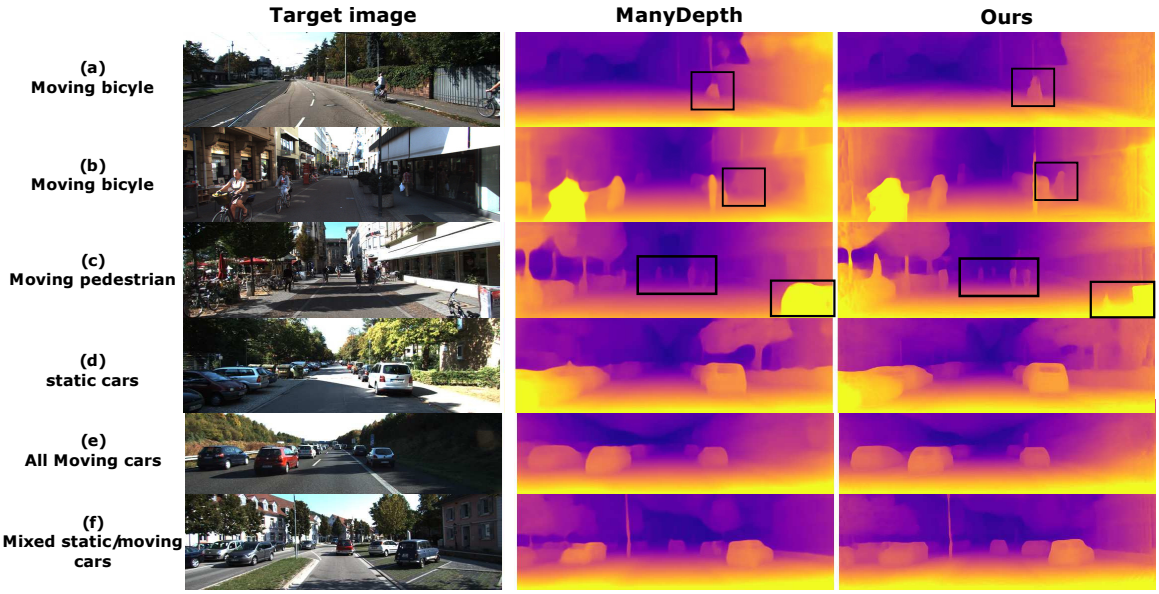


Figure 4.4: Qualitative results of the proposed method with SOTA methods [173]. (a-b-c) show complex situations, as pedestrians and bicycles tend to always move in the KITTI dataset. The qualitative results show that the proposed method outperforms the baselines. (d) The proposed method is on par with the baselines for static objects. (e) and (f) show cars as moving objects. Although the baseline [173] is trained with auto-masking, the dataset is rich with static cars that are not masked during training, this provides clues to learn the depth for moving cars. These results are validated further by the quantitative results reported in Table 4.2

Ablation	Backbone	Ego-pose input feature	Shared backbone	Piece-wise rigid pose	Abs Rel	Sq Rel	RMSE	RMSE log
A1	Resnet18 [61]	Layer5	-	-	0.121	0.914	4.890	0.196
A2	EfficientNet-b5	P_{16}	-	-	0.132	0.906	4.981	0.205
A3	EfficientNet-b5	P_8	-	-	0.127	0.983	5.010	0.201
A4	EfficientNet-b5	P_4	-	-	0.121	0.894	4.886	0.197
A5	EfficientNet-b5	P_4	✓	-	0.120	0.925	4.868	0.194
A6	EfficientNet-b5	P_4	✓	✓	0.113	0.795	4.689	0.190
A7	EfficientNet-b6	P_4	✓	✓	0.110	0.719	4.486	0.184

Table 4.3: An ablation study of the proposed method. The evaluation was done on KITTI benchmark using Eigen split [44]. As observed, the effect of the backbone is minimal A1 vs A5, the choice of the input feature for ego-pose head is sensible A2 vs A3 vs A4, the performance of the proposed method is obtained mainly by the introduction of the piece-wise rigid pose A5 vs A6. Increasing the complexity of the model allows better performances and better training stability A6 vs A7

These results suggest that not only the models learn an accurate depth, but also accurate instance pose. This result demonstrates that the transformer network is able to match and learn the interaction of the objects across time. The model in $A5$ is on the same setting of the other SOTA methods [173, 141]. Despite using this low performance baseline, the introduction of the dynamic warping enabled the proposed method to achieve the SOTA results.

An interesting observation during training is that $A6$ under-fits the data (i.e., the validation loss is less than the learning loss). The test performances are not stable, the best model among the 20 epochs is reported for this backbone. In order to resolve this under-fitting, the complexity of the model is increased $A7$. This allows for a better stability of the training loss and test performance. The best results are obtained using this complexity. The additional instance pose results in an additional run-time overhead during training. The training time for 1 epoch for A5 and A7 is $233mn$ and $58mn$ trained on RTX3090 respectively. However, the additional run-time is only for the training. At test-time, the depth network requires only a single pass of the image $\mathbf{I}_t$ with roughly 34FPS for $A7$ model and 38FPS for $A6$ model using a single RTX3090.

4.6 Disucssion

In this chapter, a novel instance poses head is introduced for self-supervising monocular depth inference. This head enables the factorization of the scene's motion. Thus, alleviating the rigid scene assumption. It is shown that it achieves the SOTA results on the KITTI benchmark [49]. The ablation study further validates that the multi-head attention of the transformer network infer an accurate object pose. Moreover, the impact of the dynamic motion on this benchmark is exposed. Namely, the bias towards static objects, where 86.43% of the test pixels correspond to static objects. A mean per static/dynamic category metric is proposed to unbias the assessment.

One fundamental limitation of these single-image-to-depth methods is that these models rely on the prior knowledge such as object shape, textures, camera position with respect to the floor in order to recover the depth. Recovering the geometry with triangulation or matching is not possible, as the

network uses single images only. However, even this capability of recovering 3D from 2D with good accuracy is already an impressive result. Another limitation of the current method is the depends on the performance of the instance segmentation network. While the panoptic segmentation network works well on KITTI, the performance of this model is not guaranteed when scaling the training for other datasets. This might limit the possibility to apply this method on huge datasets where the self-supervision is more pertinent.

Chapter 5

Video-to-depth forecasting

Now that self-supervised monocular depth inference has been presented, this next chapter will look at future depth forecasting. As discussed earlier, in this chapter, the term "forecasting" will be used to describe the methods that output the **future** depth of a sequence of images. Given a sequence of raw images, the aim is to forecast the 3D information using a self supervised photometric loss. The architecture is designed using both convolution and transformer modules. This leverages the benefits of both modules: the Inductive bias of CNN, and the multi-head attention of transformers, thus enabling a rich spatio-temporal representation that enables accurate depth forecasting. The approach performs significantly well on the KITTI dataset benchmark, with several performance criteria being even comparable to prior non-forecasting self-supervised monocular depth inference methods.

In the Section Sec. 5.1, we motivate our method. We discuss related work in Sec. 5.2. We present our approach in Sec. 5.3 and experimental results in Sec. 5.4. We conclude in Sec. 5.5.

This chapter is based on the following publication:

- **Conference paper:** Boulahbal Houssem Eddine, Adrian Voicila, and Andrew I. Comport. "Forecasting of depth and ego-motion with transformers and self-supervision." 2022 26th International Conference on Pattern Recognition (ICPR). IEEE, 2022.

5.1 Introduction

Forecasting the future is crucial for intelligent decision-making. It is a remarkable ability of human beings to effortlessly forecast what will happen next, based on the current context and prior knowledge of the scene. Forecasting sequences in real-world settings, particularly from raw sensor measurements, is a complex problem due to the exponential time-space space dimensionality, the probabilistic nature of the future and the complex dynamics of the scene. Whilst much effort from

the research community has been devoted to video forecasting [115, 46, 175, 138] and semantic forecasting [160, 6, 56, 150], depth and ego-motion forecasting have not received the same interest despite their importance. The geometry of the scene is essential for applications such as planning the trajectory of an agent. Anticipating is therefore important for autonomous driving autopilots or human/robot interaction, as it is critical for the agent to quickly respond to changes in the external environment.

The first work that explored depth forecasting was carried out by Mahjourian *et al* [113], the aim of that paper was to use the forecasted depth to render the next RGB image frame. They supervised the depth loss using ground-truth LiDAR scans and the warping was done using ground-truth poses. [134] used additional modalities for input, namely, a multi-modal RGB, depth, semantic and optical flow and forecasted the same future modalities. The supervision was carried out using the aforementioned ground-truth labels. [67] developed a probabilistic approach for forecasting using only input images and generated a diverse and plausible multi-modal future including depth, semantics and optical flow. However, it was supervised through ground-truth labels and the final loss was a weighted sum of future segmentation, depth and optical flow losses similar to [134]. While these methods enable forecasting the depth, they suffer from two shortcomings: [67, 134, 113] require the ground-truth labels for supervision during training and testing and [134] uses a multi-modal input for inference that requires either ground-truth labels or a separate network.

The work presented in this chapter addresses the problem of depth and ego-motion forecasting using only monocular images sequence with self-supervision. Monocular depth and ego-motion inference has been successful for self-supervised training [167, 1, 73, 25, 51, 50, 30, 140, 141, 182]. The basic idea is to jointly learn depth and ego-motion supervised by a photometric reconstruction loss. In this chapter, it is demonstrated that it is possible to extend this self-supervised training to sequence forecasting. An accurate forecasting requires a knowledge of the ego-motion, semantics, and the motion of dynamic objects. Powered by the advances of transformers [162, 38, 16, 41, 108], and using only sensor input, the network learns a rich spatio-temporal representation that encodes the semantics, the ego-motion and the dynamic objects. Therefore, avoiding the need for extra labels for training and testing. The results on the KITTI benchmark [49] show that the proposed method is able to forecast the depth accurately and outperform even non-forecasting methods [44, 101, 189, 181].

5.2 Related work

5.2.1 Sequence forecasting

Anticipation of the future state of a sequence is a fundamental part of the intelligent decision-making process. The forecasted sequence could be an RGB video sequence [46, 175, 4, 115, 88, 138], depth image sequence [174, 113], semantic segmentation sequence [160, 6, 110, 56, 33, 67, 56, 150] or even

a multi-modal sequence [67, 134]. Early deep learning models for RGB video future forecasting leveraged several techniques including: Recurrent models [175], variational autoencoder VAE [4], generative adversarial networks [115], autoregressive model [138] and normalizing flows[88]. These techniques have inspired subsequent sequence forecasting methods. Despite the importance of using geometry for developing better decision-making, depth forecasting is still in early development. [113] used supervised forecasted depth along with supervised future pose to warp the current image and generate the future image. Instead of using images as input, [174] used LiDAR scans and forecast a sparse depth up to 3.0s in the future on the KITTI benchmark [49]. [134] used a multi-modal input/output and forecast the depth among other modalities. [67] handled the diverse future generation by utilizing a variational model to forecast a multi-modal output. The use of multi-modalities requires additional labels or pretrained networks. This makes the training more complicated. Instead, the work presented in this chapter leverages only raw images and forecasts in a self-supervised manner.

5.2.2 Vision transformers

The introduction of the Transformers in 2017 [162] revolutionized natural language processing, resulting in remarkable results [38, 16, 136]. The year 2020 [41, 17] marked one of the earliest pure vision transformer networks. As opposed to recurrent networks that process sequence elements recursively and can only attend to short-term context, transformers can attend to complete sequences, thereby learning long and short relationships. The multi-head attention could be considered as a fully connected graph of the sequence's features. It demonstrated its success by outperforming convolution based networks on several benchmarks including classification [41, 183, 97], detection [97, 17, 95] and segmentation [108, 32]. This has led to a paradigm shift [103], transformers are slowly winning "The Hardware Lottery" [66]. However, training vision transformers is complicated as these modules are not memory efficient for images and need large dataset pretraining. [17] has demonstrated that it is possible to combine convolution and transformers to learn a good representation without requiring large pertaining. The proposed method proposes to leverage a hybrid CNN and transformer network as in [17] that is designed to forecast the geometry of the scene. The proposed network is simple and yet efficient. It outperforms even prior monocular depth inference methods [44, 101, 189, 181] that have access to the ground truth.

5.3 The method

5.3.1 The problem formulation

Let $\mathbf{I}_t \in \mathbb{R}^{w \times h \times c}$ be the t-th frame in a video sequence $\mathbf{I} = \{\mathbf{I}_{t-k:t+n}\}$. The frames $\mathbf{I}_c = \{\mathbf{I}_{\mathbf{t-k:t}}\}$ are the context of $\mathbf{I}_t$ and $\mathbf{I}_f = \{\mathbf{I}_{t+1:t+n}\}$ is the future of $\mathbf{I}_t$. The goal of the future depth and

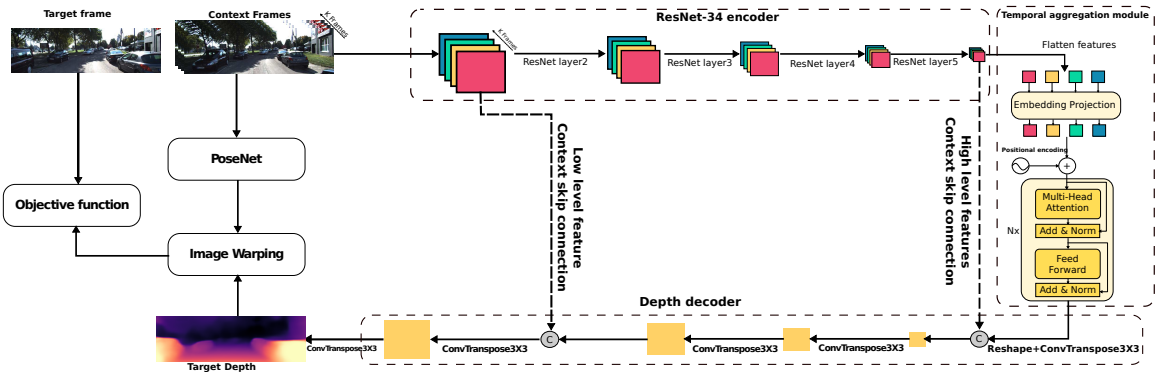


Figure 5.1: Illustration of the proposed architecture. Two sub-networks are used for training: The PoseNetwork as in network [80, 50] is used to forecast the ego-motion. The depth network combines both CNN and transformers. The Resnet34 [61] encoder extracts the spatial features for each context frame. The embedding projection module projects these features into $R^{k \times d_{model}}$ where $k = 4$ is the context frames. $N = 3$ transformer encoders are used to fuse the spatial temporal to obtain a rich spatio-temporal features. The output of the transformer module encodes the motion of the scene. The decoder uses simple transposed convolution. In order to recover the context, skip connections are pooled from the encoder. Only the last frame features are pooled for the context. The decoder outputs a disparity map that will be used along with the pose network to warp the source images onto the target.

ego-motion forecasting is to predict the future depth image of the scene $\mathbf{D}_{t+n}$ and the ego-motion ${}^{t+n}\mathbf{T}_t$ corresponding to $\mathbf{I}_{t+n}$ given only the context frames $\mathbf{I}_c$:

$$(\widehat{\mathbf{D}}_{t+n}, {}^{t+n}\widehat{\mathbf{T}}_t) = f(\mathbf{I}_c; \boldsymbol{\theta}) \tag{5.1}$$

where f is a neural network with parameters $\boldsymbol{\theta}$.

In self-supervised learning depth inference, the problem is formulated as novel view synthesis by warping the source frame $\mathbf{I}_s$ into the target frames $\mathbf{I}_{tar}$ using the depth and the ${}^{s}\mathbf{T}_{tar} \in \mathbb{SE}[3]$ pose target to source pose. The warping is defined as defined in Eq. 4.2:

$$\widehat{\mathbf{p}}_s \sim \pi(\mathbf{K}\,{}^{s}\mathbf{T}_t H(\mathbf{D}_t \mathbf{K}^{-1} \mathbf{p}_t)) \tag{5.2}$$

Reconstructing the frame $\mathbf{I}_{t+n}$ using the depth and the pose from only the context by a warping could be formulated as a maximum likelihood problem:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\operatorname{argmin}}\, L(\mathbf{I}_{t+n} | \mathbf{I}_c; \boldsymbol{\theta}) \equiv \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\operatorname{argmin}} \sum_m P_{model}(\mathbf{I}^m_{t+n} | \mathbf{I}^m_c) \tag{5.3}$$

where m is the number of samples. If P_{model} is assumed to follow a Laplacian distribution $P_{model}(\mathbf{I}_{t+n}|\mathbf{I}_c) \sim Lap(\mathbf{I}_{t+n}; \boldsymbol{\mu} = \widehat{\mathbf{I}}_{t+n}; \boldsymbol{\beta} = \sigma^2 \mathbf{I})$. $\widehat{\mathbf{I}}_{t+n}$ is the warped image. Then, maximizing the Eq. 5.3 is equivalent to minimizing an L_1 error of $\widehat{\mathbf{I}}_{t+n}$ and the known image frame $\mathbf{I}_{t+n}$. Similarly, if the distribution is assumed to follow a Gaussian distribution, the maximization is equivalent to minimizing an L_2 error.

5.3.2 The architecture

The architecture of the network is depicted in Fig. 5.1. The forecasting network is composed of two subnetworks: a pose net to forecast the future ${}^{s}\mathbf{T}_{t}$, that transform the target to the source frame, and a depth network that forecast $\mathbf{D}_{t+n}$ (see Eq. 5.1). Similar to [80], the pose-net is composed of a classification network [61] as a feature extractor followed by a simple pose decoder as in [50]. The pose network forecasts 6 parameters using the axis-angle representation. The depth network leverages a hybrid CNN and Transformer network as in [17] that is designed to forecast the geometry of the scene. This network benefits from both modules. The convolution module is used to extract the spatial features of the frames as it is memory efficient, easy to train and does not require large pretraining. The transformer module is used for better temporal feature aggregation. The multi-head attention could be considered as a fully-connected graph of the features of each frame. Therefore, the information is correlated across all the frames rather than incrementally, one step at a time, as in LSTM [65]. The architecture consists of three modules: an encoder, temporal aggregation module and a decoder.

Encoder:

ResNet [61] is one of the most used foundation models [9]. It has demonstrated its success as a task agnostic feature extractor for nearly all vision tasks. In this work, ResNet34 is used as feature extractor. It is pretrained on ImageNet [37] for better convergence. Each context frame is fed-forward and a pyramid of features is extracted. These features encode the spatial relationship between each scene separately. Thus, at the output of this module, a pyramid of spatial features for each frame is constructed. These features will be correlated temporally using the Temporal aggregation module TAM.

Temporal aggregation module

Since its introduction, transformers have demonstrated their performance, outperforming their LSTM/RNN counterparts in various sequence learning benchmarks [166, 137, 16, 38]. forecasting accurate depth requires knowledge of the static objects, accurate ego-motion and knowledge of the motion of the dynamic objects. The last layer of the encoder is assumed to encode higher abstraction features (*e. g.* recognizing objects). Therefore, correlating temporally these features allows the extraction of the motion features of the scene. The TAM consists of two submodules:

- **Embedding projection:** The dimensions after flattening the feature output of the last layer of the encoder is not memory efficient for the transformers. The embedding projection maps these features as:[1] $\mathbb{R}^{K\times C\times H\times W} \longrightarrow \mathbb{R}^{K\times d_{enc}}$.

[1] The batch is omitted

- **Transformer encoder:** After projecting the features using the embedding layer, a Transformer encoder with N layer, m multi-head attention and d_{enc} is used. It correlates the spatial features of the sequence, producing fused spatio-temporal features.

Depth decoder

After the spatio-temporal fusion, the decoder takes these spatio-temporal features along with the context features as input and decodes them to produce a disparity map. As depicted in Fig. 5.1, the context of the scene is obtained by pooling the features of last frame in the encoder. Two levels are pooled and concatenated. ${}^{dec}f_{t+n} = [{}^{enc}f_t\ ,\ TAM({}^{enc}f_{t-4:t})]$. The high level features (skip connection before the TAM) enable learning the motion, while the low level features (skip connection at the start of ResNet) recover the finer details lost by the down-sampling. Therefore, the decoder maps (context + motion $\longrightarrow$ depth).

Each level of the decoder consists of a simple sequential layer of: transposed convolution with a kernel of 3×3 with similar channels to the encoder, batch normalization and Relu activation in that order. The forecasting head consists of a convolution with a kernel of 1×1 and a Sigmoid activation. The output of this activation, σ, is re-scaled to obtain the depth $D = \frac{1}{a\sigma+b}$, where a and b are chosen to constrain D between 0.1 and 100 units, similar to [50]. For training, each level has a forecasting head, but only the last head is used for inference.

5.3.3 Objective functions

As formulated in Sec. 5.3.1, learning the parameters $\widehat{\boldsymbol{\theta}}$ involves maximizing the maximum likelihood of P_{model}. As presented in Sec. 4.2.2, the loss functions that will be used to optimize the parameters of the network are:

- **Photometric loss:** Following [189, 50, 141] The photometric loss seeks to reconstruct the target image by warping the source images using the forecast pose and depth. An L_1 loss is defined as follows:

$$\mathcal{L}_{rec}(\mathbf{I}_{t+n}, \widehat{\mathbf{I}}_{t+n}) = \sum_{\mathbf{p}} |\mathbf{I}_{t+n}(\mathbf{p}) - \widehat{\mathbf{I}}_{t+n}(\mathbf{p})| \tag{5.4}$$

  where $\widehat{\mathbf{I}}_{t+n}(\mathbf{p})$ is the reverse warped target image obtained by Eq. 4.2. This simple L_1 is regularized using SSIM [171] that has a similar objective to reconstruct the image. The final photometric loss is defined as:

$$\begin{aligned}\mathcal{L}_{\text{pe}}(\mathbf{I}_{t+n}, \widehat{\mathbf{I}}_{t+n}) = \sum_{\mathbf{p}} \big[&(1-\alpha)\ \text{SSIM}[\mathbf{I}_{t+n}(\mathbf{p}) - \widehat{\mathbf{I}}_{t+n}(\mathbf{p})] \\ &+ \alpha |\mathbf{I}_{t+n}(\mathbf{p}) - \widehat{\mathbf{I}}_{t+n}(\mathbf{p})|\big]\end{aligned} \tag{5.5}$$

- **Depth smoothness:** An edge-aware gradient smoothness constraint is used to regularize the photometric loss. The disparity map is constrained to be locally smooth through the use

an image-edge weighted L_1 penalty, as discontinuities often occur at image gradients. This regularization is defined as [62]:

$$\mathcal{L}_s(D_{t+n}) = \sum_p \big[|\partial_x D_{t+n}(\mathbf{p})| e^{-|\partial_x \mathbf{I}_{t+n}(\mathbf{p})|} + |\partial_y D_{t+n}(\mathbf{p})| e^{-|\partial_y \mathbf{I}_{t+n}(\mathbf{p})|} \big] \tag{5.6}$$

Training with these loss functions is subject to major challenges: gradient locality, occlusion and out of view-objects. Gradient locality is a result of bilinear interpolation[72, 189]. The supervision is derived from the fours neighbors of $I(\mathbf{p}_s)$ which could degrade training if that region is low-textured. Following [50, 51, 48], an explicit multiscale approach is used to allow the gradient to be derived from larger spatial regions. A forecasting head is used at each level to obtain each level's disparity map during training. Eq. 4.2 assumes global ego-motion to calculate the disparity. Supervising directly using this objective is inaccurate when this assumption is violated (*e. g.* the camera is static or a dynamic object moves with the same velocity as the camera). According to [50] this problem can manifest itself as 'holes' of infinite depth. This could be mitigated by masking the pixels that do not change the appearance from one frame to the next. A commonly used solution [189, 50] is to learn a mask μ that weighs the contribution of each pixel, while [189] uses an additional branch to learn this mask. This approach uses the auto-masking defined in [50] to learn a binary mask μ as follows:

$$\mu(\mathbf{I}_{t+n}, \widehat{\mathbf{I}}_{t+n}, \mathbf{I}_t) = \mathcal{L}_{pe}(\mathbf{I}_{t+n}, \widehat{\mathbf{I}}_{t+n}) < \mathcal{L}_{pe}(\mathbf{I}_{t+n}, \mathbf{I}_t) \tag{5.7}$$

μ is set to only include the loss when the photometric loss of the warped image $\widehat{\mathbf{I}}_{t+n}$ is lower than the original unwarped image $\mathbf{I}_t$. The final objective function is defined as:

$$\mathcal{L} = \sum_l [\, \mu \, \mathcal{L}_p + \alpha_d L_s \,] \tag{5.8}$$

where l is the scale level of the forecast depth.

5.4 Experiments

5.4.1 Setting

KITTI benchmark [49]:

Following the prior work [44, 101, 189, 181, 114, 50, 167], the Eigen *et al* [44] split is used with Zhou *et al* [189]. Frames without sufficient context (starting images in video) are excluded from the training and testing. This split has become the defacto benchmark for training and evaluating depth that is used by nearly all depth methods.

Baselines:

As discussed above, previous work on depth forecasting has been supervised using LiDAR scans, and has used a multimodal network that provides depth. Their evaluation is neither performed on the Eigen split, nor does it use the defacto self-supervised metrics. In order to fairly evaluate the proposed method, a self-supervised monocular formulation will be used to compare performance with the KITTI Eigen split benchmark. Comparisons will be made with three approaches: prior work on self-supervised depth inference [44, 101, 189, 181, 50, 167]; copy of the last observed LiDAR frame as done in [134]; and ForecastMonodepth2, a modified version of [50] that is adapted for forecasting pose/depth.

Evaluation metrics:

For evaluation, the metrics of previous works [44] are used for the depth (see Sec. 2.2.6). To resolve the scale ambiguity, the forecast depth map is scaled by median scaling where $s = \frac{median(D_{gt})}{median(D_{pred})}$. During the evaluation, the depth is capped to 80m. For the pose evaluation, the Absolute Trajectory Error (ATE) defined in [153] is used to evaluate on the KITTI odometery benchmark [49] for sequences 09 and 10.

Implementation details:

PyTorch [131] is used for all models. The networks are trained for 20 epochs, with a batch size of 8. The Adam optimizer [83] is used with a learning rate of $lr = 10^{-4}$ and $(\beta_1, \beta_2) = (0.9, 0.999)$. As training proceeds, the learning rate is decayed at epoch 15 to 10^{-5}. The SSIM weight is set to $\alpha = 0.15$ and the smoothing regularization weight to $\alpha_d = 0.001$. $l = 4$ scales are used for each output of the decoder. At each scale, the depth is up-scaled to the target image size. $d_{model} = 2048$, $m = 16$ and $N = 3$ for the TAM projection. The input images are resized to 192×640. Two data augmentations were performed: horizontal flips with probability $p = 0.5$ and color jitter with $p = 1$. $k = 4$ frames are used for context sequence and $n = 5$ is used for short term forecasting and $n = 10$ for midterm forecasting as in [134] which corresponds to forecasting 0.5s and 1.0s into the future. The ForecastedMonodepth2 is the same as [50] with a modified input. The context images are concatenated and used as input for both depth and pose networks.

5.4.2 Depth forecasting results

Table 5.1 shows the results of the proposed method on the KITTI benchmark [49]. As specified in Sec. 5.4.1, the method is compared to three approaches: prior work on depth inference; copying last frame; and adapting monodepth2 [50] for future forecasting. The proposed method outperforms the forecasting baselines for both short and midterm forecasting, especially for short range forecating. The results are even comparable to non-forecasting methods [44, 101, 189, 181] that have access

Method	Forecasting	Resolution	Supervision	Abs Rel	Sq Rel	RMSE log	RMSE	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Eigen *et al* [44]	-	576 x 271	D	0.203	1.548	0.282	6.307	0.702	0.898	0.967
Liu *et al* [101]	-	640 x 192	D	0.201	1.584	0.273	6.471	0.680	0.898	0.967
SfMLearner [189]	-	416 x 128	SS	0.198	1.836	0.275	6.565	0.718	0.901	0.960
Yang *et al* [181]	-	416 x 128	SS	0.182	1.481	0.267	6.501	0.725	0.906	0.963
Vid2Depth [114]	-	416 x 128	SS	0.159	1.231	0.243	5.912	0.784	0.923	0.970
Monodepth2 [50]	-	640 x 192	SS	0.115	0.882	0.190	4.701	0.879	0.961	0.982
Wang *et al* [167]	-	640 x 192	SS	0.109	0.779	0.186	4.641	0.883	0.962	0.982
LiDAR Train set mean	-	1240 x 374	-	0.361	4.826	0.377	8.102	0.638	0.804	0.894
ForecastMonodepth2	0.5sec	640 x 192	SS	0.201	**1.588**	0.275	**6.166**	0.702	0.897	0.960
Ours	0.5sec	640 x 192	SS	**0.178**	1.645	**0.257**	6.196	**0.761**	**0.914**	**0.964**
Copy last LiDAR scan	1sec	1240 x 374	-	0.698	10.502	15.901	7.626	0.294	0.323	0.335
ForecastMonodepth2	1sec	640 x 192	SS	0.231	**1.696**	0.303	6.685	0.617	0.869	0.954
Ours	1sec	640 x 192	SS	**0.208**	1.894	**0.291**	**6.617**	**0.701**	**0.882**	**0.949**

Table 5.1: Quantitative performance comparison of on the KITTI benchmark with Eigen split [49] for distances up to 80m. In the *Supervision* column, D refers to depth supervision using LiDAR groundtruth and (SS) self-supervision. At test-time, all monocular methods (M) scale the depths with median ground-truth LiDAR.

Method	forecasting	Seq.09	Seq.10
Mean Odom	-	0.032 ± 0.026	0.028 ± 0.023
ORB-SLAM [122]	-	0.014 ± 0.008	0.012 ± 0.011
SfMLearner [189]	-	0.021 ± 0.017	0.020 ± 0.015
Monodepth2 [50]	-	0.017 ± 0.008	0.015 ± 0.010
Wang *et al* [167]	-	0.014 ± 0.008	0.014 ± 0.010
Ours	0.5s	0.020 ± 0.011	0.018 ± 0.011

Table 5.2: ATE error of the proposed method and the prior non-forecasting methods on KITTI [49]. The proposed method is comparable to these methods even if it only accesses past frames.

to $\mathbf{I}_{t+n}$. The gap between state-of-the-art depth inference and the proposed forecasting method is reasonable due to the uncertainty of the future, the unobservability of certain events such as a new object entering the scene and the complexity of natural videos that requires modeling correlations across space-time with much higher input dimensions.

Fig. 5.2 shows an example of depth forecasting on the Eigen test split. Several observations can be made:

- The network handles correctly the out-of-view object.
- The network learned the correct ego-motion: The position of the static objects is accurate.

These results suggest that the network is able to learn a rich spatio-temporal representation that enables learning the motion, geometry, and the semantics of the scene. Thus, the proposed method extends the self-supervision depth inference to perform future forecasting with comparable results. A further analysis is done to evaluate and validate the choices of the network in Sec. 5.4.4.

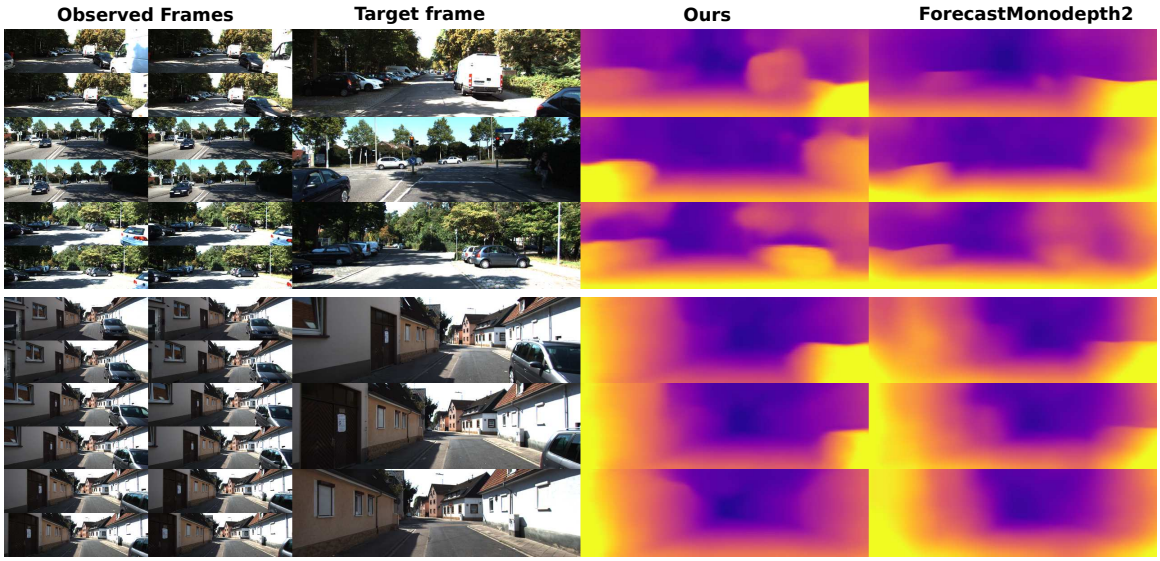


Figure 5.2: Qualitative results of the comparison of the proposed method with the ForecastMonodepth2 baseline. This comparison shows that the proposed method performs better than the baseline, especially for nearby dynamic objects. This observation is further validated in Table IV. In addition, the baseline method is showing a lack of detection of moving objects, which leads to a degradation of the forecasted depth. The proposed method is able to detect moving objects, thus accurately forecasting the depth of the scene.

Range	Method	Forecasting	Abs Rel	Sq Rel	RMSE log	RMSE	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
[00m 10m]	Monodepth2 [50]	-	0.066	0.264	0.106	1.076	0.959	0.987	0.994
	ForecastMonodepth2	0.5s	0.138	**0.586**	0.178	1.697	0.847	0.957	0.985
	Ours	0.5s	**0.112**	0.595	**0.155**	**1.573**	**0.893**	**0.964**	**0.986**
[10m 30m]	Monodepth2 [50]	-	0.119	0.858	0.192	3.706	0.876	0.956	0.978
	ForecastMonodepth2	0.5s	0.192	1.673	0.258	5.169	0.725	0.906	0.963
	Ours	0.5s	**0.167**	**1.453**	**0.241**	**4.803**	**0.782**	**0.921**	**0.965**
[30m 80m]	Monodepth2 [50]	-	0.188	3.094	11.115	0.288	0.709	0.897	0.950
	ForecastMonodepth2	0.5s	**0.213**	**3.526**	**11.940**	**0.292**	**0.631**	**0.874**	**0.953**
	Ours	0.5s	0.224	4.052	12.638	0.312	0.622	0.862	0.941

Table 5.3: Quantitative performance comparison on the KITTI benchmark with Eigen split [49] for multiple distances range. For Abs Rel, Sq Rel, RMSE and RMSE log lower is better, and for $\delta < 1.25$, $\delta < 1.25^2$ and $\delta < 1.25^3$ higher is better. Three ranges are considered: short range [0 10m] which represents 37.95%, medium-range [10 30]which represents 50.74% and long-range[30 80] which represents 11.30%. The results shows that the proposed method is able to forecast good depth and outperform the baseline at short and medium forecasting range.

Results with respect to distance

In order to further analyze the depth forecasting results, an assessment based on the ground-truth LiDAR distance is done. Table IV shows the comparison of the non-forecasting method Monodepth2 [50], ForecastMonodepth2 and the proposed methods.

The results suggest that the proposed method outperform the adaptation of Monodepth2 for short-range with a improvement of the Abs Rel of -16.7% and medium-range with an improvement of the Abs Rel of -8.8%. These regions are the most significant regions of the forecasting as they have enough parallax for the ego-motion and dynamic object motion. Besides, this region assesses several challenges, including out-of-view objects and occlusion. For the long-range forecasting, the results show that the two methods perform badly due to the lack of parallax in this region and down-sampling that ignores small objects. Moreover, this region has a high likelihood of new-objects entering the scene, which the forecasting is unable to handle by definition. The reported performances and the qualitative results suggest that the two forecasting networks only fit the road and completely ignore any other object. These results are shown qualitatively in Fig. 4.4.

5.4.3 Ego-Motion forecasting results

Table 5.2 shows the results of the proposed network on the KITTI odometry benchmark [49]. Similar to depth, the assessment is made by comparing with non-forecasting prior works. To avoid data leakage, the network is trained from scratch on the sequences 00-08 of the KITTI odometry benchmark. The network takes only the context images $\mathbf{I}_c$ and forecasts ${}^{t}\mathbf{T}_{t+n}$. The ATE results in Table 5.2 show the proposed network achieved a competitive result relative to other non-forecasting approaches. All the methods are trained in the monocular setting, and therefore scaled at test time using ground-truth. These results suggest that using the proposed architecture along with the self-supervised loss function successfully learns the future joint depth and ego-motion.

5.4.4 Ablation study

To further analyse the network, several ablations are made. Table 5.4 depicts a comparison of the proposed model with several variants. The evaluation is done for short-term forecasting $n = 5$ using $k = 4$ context frames.

Effect of the Temporal Aggregation Module

In order to evaluate the contribution of the multi-head attention, a variant of the proposed method is designed by replacing the TAM module by a simple concatenation of the last layer features. From Table 5.4, the improvement induced by the TAM module is significant across all metrics. These results suggest that the performance obtained by the proposed method is achieved through the TAM module. Since the TAM aggregates the temporal information across all frames using a

Method	Abs Rel	Sq Rel	RMSE log	RMSE
Ours	0.178	1.645	0.257	6.196
(a) Without TAM	0.205	1.745	0.296	6.565
(b) Shared pose/depth features	0.208	1.745	0.282	6.529
(c) Single scale	0.208	1.950	0.283	6.595
(d) Disable auto-masking	0.193	1.774	0.273	6.374

Table 5.4: Ablation study results showcasing the effects of different modules in the proposed method. (a) Effect of the Temporal Aggregation Module (TAM) on performance metrics. The TAM module significantly improves performance across all metrics by better encoding the spatio-temporal relationship between images. (b) Effect of sharing the encoder of depth and ego-motion networks. Sharing the encoder leads to degradation in performance as it restricts the network from finding the best local optima for both tasks. (c) The benefit of using multiple scales in the proposed method. The network benefits from the multiscale approach, as demonstrated by improved results compared to using a single scale. (d) Effect of auto-masking on forecasted depth. Auto-masking improves all evaluation criteria by rejecting outliers that hinder optimization and consequently enhancing accuracy.

learned attention, the temporal features are better correlated and the final representation successfully encodes the spatio-temporal relationship between the images.

Effect of sharing the encoder of depth and ego-motion

Since both pose and depth networks encode the future motion and geometry of the scene, it is expected that sharing the encoders of these networks yield better results. However, as reported in Table 5.4, the degradation is significant. Even though these tasks are collaborative, sharing the encoder will result in a set of parameters $\widehat{\boldsymbol{\theta}}$ that are neither the best local optima for the depth nor for the pose. By alleviating this restriction and separating the encoders, the network learns better local optima for both pose and depth.

The benefit of using multiple scales

In order to evaluate the multiscale extension, a variant of the proposed method that uses only one scale is trained. As illustrated in the Table 5.4, the network benefits from the multiscale. The reverse warping uses bi-linear interpolation. As mentioned earlier, each depth point depends only on the four neighboring warped points. By using a multiscale depth at training-time the gradient is derived from a larger spatial region directly at each scale.

Effect of auto-masking

Table 5.4 compares the proposed method with a variant without using the auto-masking defined is Sec. 5.3.3. The results show that using auto-masking improves all four evaluation criteria. This demonstrates that, using auto-masking, rejects these outliers that inhibit the optimization. This leads to better accuracy of the forecasted depth.

5.5 Discussion

The work presented in this chapter proposed an approach for forecasting future depth and ego motion using only raw images as input. This problem is addressed as end-to-end self-supervised forecasting of the future depth and ego motion. Results showed significant performances on several KITTI dataset benchmarks [49]. The performance criteria are even comparable with non-forecasting self-supervised monocular depth inference methods [44, 101, 189, 181]. The proposed architecture demonstrates the effectiveness of combining the inductive bias of the CNN as a spatial feature extractor and the multi-head attention of transformers for temporal aggregation. The proposed method learns a spatio-temporal representation that captures the context and the motion of the scene.

5.5.1 Limitations and perspectives

Even though the proposed forecasting method yields good results, there exists a gap with respect to non-forecasting methods. Several limitations contribute to this:

- A common assumption across the presented methods is that the environment is deterministic and that there is only one possible future. However, this is not accurate since there are multiple plausible futures. Given the stochastic nature of the forecasting proposed here, the network will tend to forecast a blurry depth map that represents the mean of all the possible outcomes [4].
- The network does not forecast the correct boundaries of the objects. This is due to the formulation as a maximum likelihood problem with a Laplacian distribution assumption and the deterministic nature of the architecture. As a result, the boundaries of the dynamic objects are smoothed.
- Due to the problem formulation, the scale of the forecast depth is ambiguous. this is a fundamental problem to the monocular methods. As the distance of the camera to the floor is constant, this could be used to disambiguate the scale.
- The model fails to account for the motion of distant dynamic objects due to lack of parallax.

Chapter 6

Video-to-video future depth with spatio-temporal consistency

In the previous chapters, depth inference and forecasting were explored. However, our research was accompanied by several limitations that were discovered along the way. This chapter delves deeper into these limitations.

One of the primary limitations in the previous depth inference work was the fact that most methods do not take advantage of multiple frames as input. They rely on the scene clues such as the object shape prior. A model can better understand the geometry of the scene and better understand the motion of the ego and the dynamic objects by utilizing multiple frames as input. As for depth forecasting, one of the biggest limitations we encountered was that the model tends to output a blurry output that is a mean of all possible future situations and a single future depth. Moreover, these models do not take into account the motion of objects in the scene, this can have a significant impact on future depth estimates. To address the limitations, development of more sophisticated models is required to accurately predict the depth of a scene.

In this chapter, a self-supervised model that simultaneously predicts a sequence of future frames from video input with a novel spatial-temporal attention (ST) network is proposed. The ST transformer network allows constraining both temporal consistency across future frames whilst constraining consistency across spatial objects in the image at different scales. This was not the case in prior works for depth prediction, which focused on predicting a single frame as output. The proposed model leverages prior scene knowledge such as object shape and texture similar to single-image depth inference methods, whilst also constraining the motion and geometry from a sequence of input images. Apart from the transformer architecture, one of the main contributions with respect to prior works lies in the objective function that enforces spatio-temporal consistency across a sequence of output frames rather than a single output frame. As will be shown, this results in more accurate and

robust depth sequence forecasting. The model achieves highly accurate depth forecasting results that outperform existing baselines on the KITTI benchmark. Extensive ablation studies were performed to assess the effectiveness of the proposed techniques. One remarkable result of the proposed model is that it is implicitly capable of forecasting the motion of objects in the scene, rather than requiring complex models involving multi-object detection, segmentation, and tracking. In the Sec. 6.1, we motivate our method. We present our approach in Sec. 6.2 and experimental results in Sec. 6.3. We conclude in Sec. 6.4.

This chapter was based on the following paper:

- **To be submitted:** Boulahbal Houssem Eddine, Adrian Voicila, and Andrew Comport. "STDepthFormer: Predicting Spatio-temporal Depth from Video with a Self-supervised Transformer Model." arXiv preprint arXiv:2303.01196 (2023).

6.1 Introduction

State-of-the-art approaches, such as [173, 57, 13] (see Chapter 4), have developed models that output a single depth image. The underlying model is then used to perform inference or forecasting tasks separately. These approaches are, however, limited because they cannot enforce spatio-temporal consistency in the output, as they do not predict a sequence. By introducing a model that predicts a sequence of depth images, the model proposed here can apply motion and geometric constraints to the output which improves the accuracy and sharpness of the forecasting and forces the predicted images to be more deterministic (ie. it does not average across possible future outcomes as in prior works).

On one hand, the majority of self-supervised monocular depth inference methods [44, 14, 147, 92, 167, 140, 54, 51, 189, 50, 75, 141] rely on a single frame as input. While this approach is effective at leveraging prior knowledge such as object shape and textures, it is limited in its ability to learn the geometry and the motion of the scene. By contrast, using multiple frames [173, 57, 70] as input has the potential to provide a more comprehensive view of the scene and to help the model better understand the relationships between objects and their motions.

Depth forecasting self-supervised methods [113, 134, 67, 13], on the other hand, often produce a blurry depth map that represents the mean of all possible future scenarios [13]. This approach fails to produce an accurate depth, which limits its usefulness in decision-making contexts.

To address these limitations, a self-supervised model is proposed that can simultaneously output a depth sequence encompassing inference and forecasting. By using multiple image frames as input and output, the model can learn about the geometric consistency of the scene, which enables it to predict more accurate depth sequences as output. The proposed model enforces a spatio-temporal consistency in the output depth sequence by warping neighboring images onto the target image using a geometric and photometric warping operator that depends on the output depths. As will

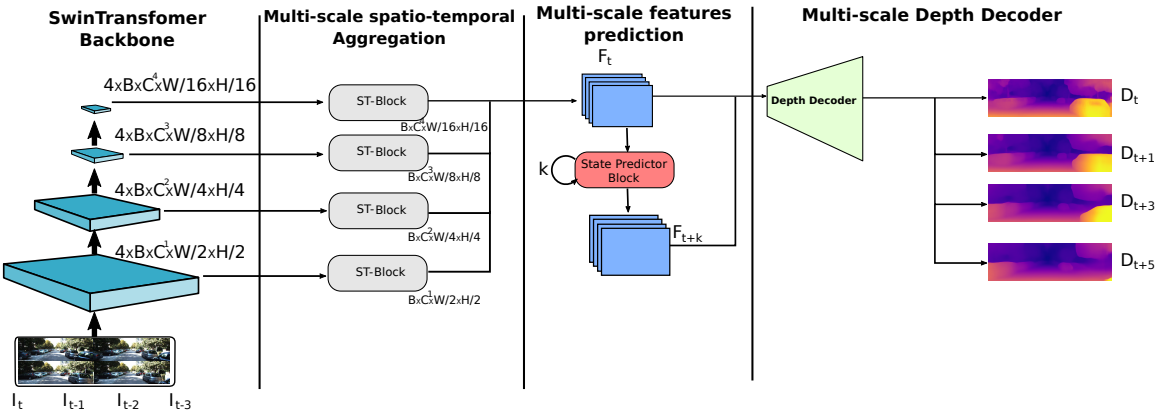


Figure 6.1: Architecture of the proposed method. The network comprises four stages. Firstly, the spatial feature of each frame is extracted using a SwinTransformer backbone shared across the context frames. Secondly, the features are correlated spatio-temporally using the ST-block shown in Fig. 6.2. Thirdly, a learned function f is used to transition from F_{t+k-1} to F_{t+k}, and this module consists of SwinTransformer blocks as well. Finally, the depth decoder employs skip connections to utilize multi-scale features and outputs 4 depth states: $(\mathbf{D}_t, \mathbf{D}_{t+1}, \mathbf{D}_{t+3}\mathbf{D}_{t+5})$

be detailed further in Sec. 6.2.3. The results show that this effectively constraints the output depth forecasting to choose the most probable outcome of the future depth, instead of using the mean of all outcomes, avoiding the issue of blurry depth maps and leading to more precise depth.

As a result, the proposed approach produces an accurate depth sequence. In summary, the contributions of the proposed method are:

- A self-supervised model that predicts a spatially and temporally consistent depth sequence that captures both present and future depth information, allowing for more comprehensive and accurate depth.
- A transformer-based multi-frame architecture that implicitly learns the geometry of the scene in an image-based end-to-end manner. Interestingly, the proposed model is capable of forecasting the motion of objects in the scene, even in the absence of explicit motion supervision.
- The method achieves highly accurate depth forecasting results that outperform existing baselines in the KITTI [49] benchmark.
- Improved generalization for depth inference tasks over SOTA.
- A comprehensive analysis of the proposed method is conducted through several ablation studies.

6.2 Method

6.2.1 Problem formulation

The aim of monocular depth inference and forecasting is to predict an accurate depth sequence through the mapping, $\mathbf{D}_{t:t+n} = f(\mathbf{I}_{t-k:t}; \boldsymbol{\theta})$ where $\mathbf{I}_{t-k:t}$ are the k context images and $\mathbf{D}_{t:t+s}$ are the s depth target states. In self-supervised learning, this model is trained via novel view synthesis by warping a set of source frames $\mathbf{I}_{src}$ to the target frame $\mathbf{I}_{tgt}$ using the learned depth $\mathbf{D}_{tgt}$ and the target to source pose ${}^{src}\mathbf{T}_{tgt} \in \mathbb{SE}[3]$ [72]. The differentiable warping is defined in 4.2

$$\widehat{\mathbf{p}}_{src} \sim \pi(\mathbf{K}\,{}^{src}\mathbf{T}_{tgt} H(\mathbf{D}_{tgt}\mathbf{K}^{-1}\mathbf{p}_{tgt})) \tag{6.1}$$

The depth network takes $k = 4$ context images as input. With $k = 4$, it is possible to learn the velocity and the acceleration without exploding the memory. The network produces $tgt = \{0, 1, 3, 5\}$ depth outputs. As the pose network is only used for supervision during training, providing the future images will help the pose network to learn better. Therefore, for each depth state tgt, the pose network input is the triplet of images $(tgt-1, tgt, tgt+1)$. It outputs two poses, ${}^{tgt-1}\mathbf{T}_{tgt}$ and ${}^{tgt}\mathbf{T}_{tgt+1}$.

6.2.2 Architecture

The proposed model is related to classic *structure-from-motion*. During training, self-supervision is achieved by using an image warping function, and two networks are used: a pose network and a depth network. At test time, only the depth network is used to output the depth.

The depth network

Fig. 6.1 shows the architecture of the proposed method. The depth network uses $k = 4$ context inputs. The architecture comprises four stages:

1. Spatial feature extraction: SwinTransformer backbone [107] is used to extract the features of each frame. The swin-tiny variant is used with a number of layers : $[2, 2, 6, 2]$, with depths of $[3, 6, 12, 24]$, a patch embedding channel of 7, and an embedding dimension of 96. It is pretrained on the ImageNet dataset [37]. See Sec. 2.5.3 for more details. This feature extractor is shared across the context frames. The feature map at each scale is extracted as input for the next module. As the purpose of this module is to extract spatial information only, calculating the gradient for only one context frame is sufficient. Experimentally, no differences were observed between calculating the gradient for all four frames and only one frame. Therefore, backpropagation is carried out only on the first frame to minimize the memory footprint.

2. Multi-scale spatio-temporal aggregation: Next, the features are correlated spatio-temporally using the proposed novel ST-block. Fig. 6.2 shows the architecture of this fusion block.

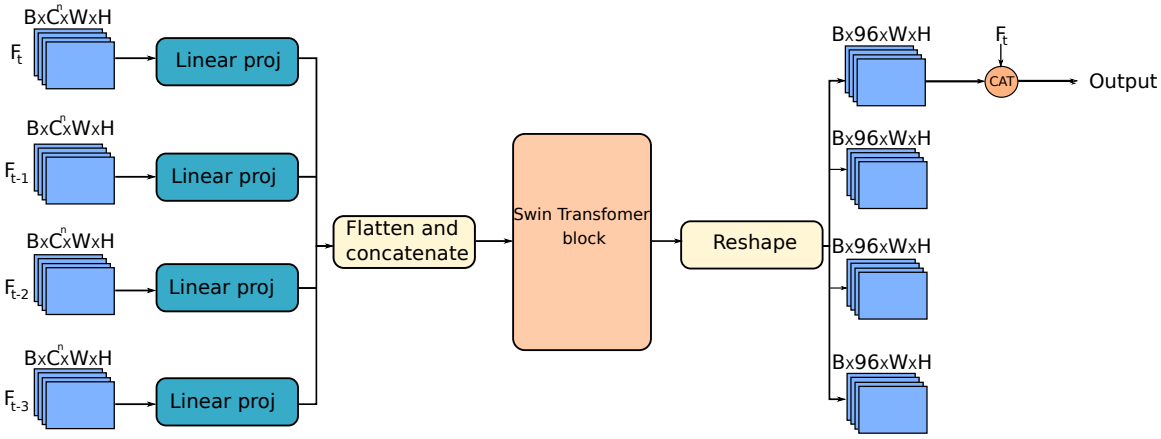


Figure 6.2: Architecture of a multiscale spatio-temporal aggregation network using linear projection and SwinTransformer layers for feature spatio-temporal correlation.

At each feature scale, each feature map of each frame is projected using a *Conv2D* with $kernal = 1$ outputting an embedding of dimension 96. These features are concatenated as a sequence of patches to construct embeddings that will be used as input to the transformers. This sequence is then provided to the transformer [107]. This block has a depth of 2 and embeddings of 96 and the number of heads at each scale is $[3, 6, 12, 24]$ from high to low resolution. The attention map performs the spatio-temporal correlation of these features. The sequence is reshaped to its original shape and the first feature map is contacted with the context feature F_t. Finally, another projection layer outputs the spatio-temporal features to recover the channel to C^n.

3. Multiscale feature prediction: A transition function f is used to relate each feature to a state in the output sequence. At each scale, this learned function f is used to transition from F_{t+k-1} to F_{t+k}. This function is recursive and defined as:

$$F_{t+k} = f(F_{t+k-1}) \tag{6.2}$$

This module is composed of SwinTransformer blocks [107]. It is shared and used recursively across all n frames to be forecast. Fig. 6.3 shows the architecture of the state predictor block. The input feature map F_t is projected to have an embedding dimension of 96. This map is flattened to patches of size 1 to be used as input to the SwinTransfomer block. Similarly, this block has a depth of 2, embeddings of 96 and the number of heads of each scale is $[3, 6, 12, 24]$ from high to low resolution. The output is reshaped to its original dimensions and concatenated with the input with a skip connection. A linear projection is used to obtain the features of F_{t+1} with size: $B \times C^n \times W \times H$.

4. Depth decoder: This module is shared across all state features. It consists of Transposed2DConvolution with ReLU as activation and a kernel size of $k = 3$, which is similar to [50]. Skip connections are employed since the previous stage outputs multi-scale features. In this method,

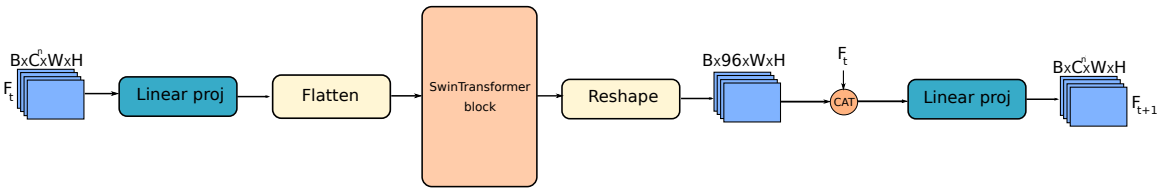


Figure 6.3: SwinTransformer-based state predictor block. The input feature map F_t is projected onto an embedding dimension of size 96 and flattened into patches for the SwinTransformer block. The output is reshaped and concatenated with the input using a skip connection. A linear projection generates the features of $Ft+1$ with size $B \times C_n \times W \times H$ where C_n is the original channel

four depth states are output: $(\mathbf{D}_t, \mathbf{D}_{t+1}, \mathbf{D}_{t+3}, \mathbf{D}_{t+5})$.

The pose network

This network is an off-the-shelf model taken from [50] that takes the triplet $(tgt-1, tgt, tgt+1)$ and outputs two poses: ${}^{tgt-1}\mathbf{T}_{tgt}$ and ${}^{tgt}\mathbf{T}_{tgt+1}$. This model is used only for self-supervised training and is discarded at evaluation.

It is worth noting that current state-of-the-art methods utilize a plane sweep approach, such as the one proposed by [173, 57], that involve explicitly providing the pose and camera parameters to the depth network and constructing a matching volume during training and evaluation. Alternatively, the proposed method adopts a different approach that learns this information implicitly. This presents several benefits, most notably the ability for the two networks, the depth and the pose, to operate independently. This independence from the pose network and camera parameters is particularly significant, as it allows the proposed network to generalize better and perform more robustly. Empirical evidence supporting this claim is presented in Sec. 6.3.4, where the experimental results demonstrate the superiority of the proposed approach.

6.2.3 Objective functions

As the self-supervision is done by reconstructing the frames $\mathbf{I}_{tgt}$ such as $tgt \in \{0, 1, 3, 5\}$ using the depth and the pose with the warping, this can be formulated as a maximum likelihood problem:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\operatorname{argmin}} L(\mathbf{I}_{t+5}, \mathbf{I}_{t+3}, \mathbf{I}_{t+1}, \mathbf{I}_t | \mathbf{I}_{t-4:t+6}; \boldsymbol{\theta}) \tag{6.3}$$
$$\widehat{\boldsymbol{\theta}} \equiv \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\operatorname{argmin}} \sum_m P_{model}(\mathbf{I}^m_{t+n} | \mathbf{I}^m_c, \mathbf{I}^m_f)$$

$\boldsymbol{\theta}$ is the model parameters, $\mathbf{I}^m_c$ are the context frames that will be provided to the depth network and $\mathbf{I}^m_f$ are the future frames that will be provided to the pose network. As presented in the Sec. 4.2.2, the photometric loss and structural similarity index measure SSIM [171], along with the depth

smoothness, are used to optimize the parameters.

$$pe(\mathbf{I}_{tgt}, \widehat{\mathbf{I}}_{(tgt\pm1\rightarrow tgt)}) = \sum_{\mathbf{p}} [(1-\alpha)\ \text{SSIM}[\mathbf{I}_{tgt}(\mathbf{p}) - \widehat{\mathbf{I}}_{(tgt\pm1\rightarrow tgt)}(\mathbf{p})] + \alpha|\mathbf{I}_{tgt}(\mathbf{p}) - \widehat{\mathbf{I}}_{(tgt\pm1\rightarrow tgt)}(\mathbf{p})|] \tag{6.4}$$

Such that $\widehat{\mathbf{I}}_{(tgt\pm1\rightarrow tgt)}$ is reconstructed from two views : $\mathbf{I}_{tgt-1}$ and $\mathbf{I}_{tgt+1}$. Similar to [50], the minimum projection loss of the two frames is used to handle occlusions leading to:

$$\mathcal{L}_{\text{ph}}(\mathbf{I}_{tgt}) = min\big[pe(\mathbf{I}_{tgt}, \widehat{\mathbf{I}}_{(tgt-1\rightarrow tgt)}), pe(\mathbf{I}_{tgt}, \widehat{\mathbf{I}}_{(tgt+1\rightarrow tgt)})\big] \tag{6.5}$$

To further improve the training, outlier rejection is performed. Similar to [50] this is done using auto-masking which is defined as:

$$\mu = [min_{tgt}(pe(\mathbf{I}_{tgt}, \widehat{\mathbf{I}}_{(tgt\pm1\rightarrow tgt)}), pe(\mathbf{I}_{tgt}, \mathbf{I}_{(tgt\pm1)}))] \tag{6.6}$$

where [ ] is the Iverson bracket. μ is set to only include the loss of pixels where the re-projection error of the warped image $\widehat{\mathbf{I}}_{(tgt\pm1\rightarrow tgt)}$ is lower than that of the original, unwarped image $\mathbf{I}_{(tgt\pm1)}$. An edge-aware gradient smoothness constraint is used to regularize the photometric loss. The disparity map is constrained to be locally smooth.

$$\mathcal{L}_s(D_{tgt}) = \sum_{p} \big[|\partial_x D_{tgt}(\mathbf{p})|e^{-|\partial_x \mathbf{I}_{tgt}(\mathbf{p})|} + \quad |\partial_y D_{tgt}(\mathbf{p})|e^{-|\partial_y \mathbf{I}_{tgt}(\mathbf{p})|}\big] \tag{6.7}$$

Temporal consistency is enforced during training through a loss function that enforces geometric constraints simultaneously across multiple output frames, namely $\mathbf{D}_{t+5}$, $\mathbf{D}_{t+3}$, $\mathbf{D}_{t+1}$, and $\mathbf{D}_t$ via a warping function. The image-based loss function minimizes the pair-wise photometric consistency by warping neighboring images for each central target ($\mathbf{I}_{t+5}$, $\mathbf{I}_{t+3}$, $\mathbf{I}_{t+1}$, and $\mathbf{I}_t$). The warping function depends on the output depth and pose outputs from the network. This constrains the model to respect image consistency between these frames. For example, $\mathbf{I}_{t+1}$ is minimized with respect to the warped $\widehat{\mathbf{I}}_t$ and the warped $\widehat{\mathbf{I}}_{t+2}$. The gradient locality problem [189] is handled using a pyramid of depth outputs, and the optimization is done on all these levels. The final loss function is defined as:

$$\mathcal{L} = \frac{1}{m} \sum_{m} \sum_{tgt} \sum_{l=1}^{l=4} \mu \mathcal{L}_{\text{ph}}(\mathbf{I}_{tgt}) + \alpha_s \mathcal{L}_s(D_{tgt}) \tag{6.8}$$

where m is the batch size, $tgt \in 0, 1, 3, 5$ and l represents the multiscale output depth.

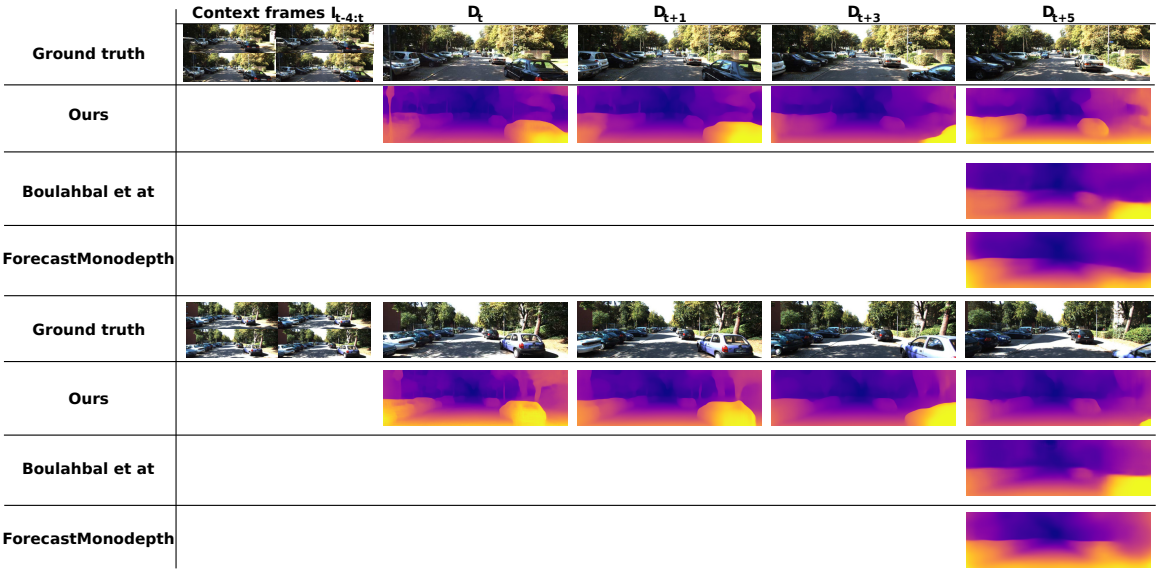


Figure 6.4: Qualitative comparison of the proposed method and the prior work on KITTI Eigen test benchmark. The proposed method is able to generate an accurate future depth sequence that exhibits significantly more details compared to the prior work. The depth map generated by the proposed method is remarkably sharp and not blurry. This superior performance can be attributed to the fact that the proposed method was specifically trained for depth inference with spatio-temporal consistence across the forecast range, resulting in an enforced deterministic output. As a result, the proposed approach predicts the most probable future instead of averaging all possible futures, as done in the prior work.

6.3 Results

6.3.1 Experimental setup

Datasets: The KITTI benchmark [49] is the defacto benchmark for evaluating depth methods. The Eigen *et al* [44] is used with Zhou *et al* [189] preprocessing to remove static frames. In order to test the generalization of the method, the Cityscapes [35] and the Robotcar [112] datasets are used. The Cityscapes dataset does not provide ground truth LiDAR depth and uses the classical SGM method [64] to obtain the depth. The LiDAR depth is projected onto the image to obtain the ground truth for the Robotcar dataset.

Baselines: Several depth inference method were used for the comparison [173, 57]. For forecasting, a comparison is made only with methods that perform self-supervision with respect to frame 5 as done in [13]. To test the performance of the method with respect to dynamic objects, the analysis provided in [14] is used.

Hyperparameters: The networks are trained for 6 epochs, with a batch size of 4. The Adam optimizer [83] is used with a learning rate of $lr = 10^{-4}$ and $(\beta_1, \beta_2) = (0.9, 0.999)$. The SSIM weight is set to $\alpha = 0.15$ and the smoothing regularization weight to $\alpha_s = 0.001$. $l = 4$ scales are used for

Predicted frame	Method	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
$\mathbf{t = 0}$	SfMLearner [189]	0.198	1.836	0.275	6.565	0.718	0.901	0.960
	Yang *et al* [181]	0.182	1.481	0.267	6.501	0.725	0.906	0.963
	GeoNet [182]	0.155	1.296	5.857	0.233	0.793	0.931	0.973
	CC [140]	0.140	1.070	5.326	0.217	0.826	0.941	0.975
	Monodepth2 [50]	0.115	0.903	4.863	0.193	0.877	0.959	0.981
	Lee *et al* [92]	0.113	0.835	4.693	0.191	0.879	0.961	0.981
	PackNetSfm [141]	0.111	0.785	4.601	0.189	0.878	0.960	0.982
	Manydepth [173]	**0.098**	**0.770**	**4.459**	**0.176**	**0.900**	**0.965**	**0.983**
	Ours	0.110	0.805	4.678	0.187	0.879	0.961	0.983
$\mathbf{t = 1}/0.1sec$	**Ours**	0.121	0.989	5.026	0.203	0.863	0.951	0.978
$\mathbf{t = 3}/0.3sec$	**Ours**	0.146	1.295	5.493	0.227	0.824	0.935	0.971
$\mathbf{t = 5}/0.5sec$	ForecastMonodepth2 [13]	0.201	1.588	6.166	0.275	0.702	0.897	0.960
	Boulahbal *et al* [13]	0.178	1.645	6.196	0.257	0.761	0.914	0.964
	Ours	**0.165**	**1.489**	**5.805**	**0.245**	**0.792**	**0.921**	**0.964**

Table 6.1: Quantitative performance of the proposed method on the KITTI benchmark [49] with eigen [44] benchmark for the frames $D_t, D_{t+1}, D_{t+3}, D_{t+5}$. for Abs Rel, Sq Rel, RMSE and RMSE log lower is better. For $\delta < 1.25$, $\delta < 1.25^2$, $\delta < 1.25^3$ higher is better. The proposed method is able to output an accurate depth at different time steps. The performance of the future depth is even comparable to depth inference method that have access to the target frame.

each output of the decoder. At each scale, the depth is upscaled to the target image size. The input images are resized to 192×640. Two data augmentations were performed: horizontal flips with probability $p = 0.5$ and color jitter with p = 1. The activation of depth decoder, σ, is re-scaled to obtain the depth $D = \frac{1}{a\sigma+b}$, where a and b are chosen to constrain D between 0.1 and 100 units for training and $(0.5, 100)$ for evaluation, similar to [14]. The scale ambiguity is resolved using median scaling, similar to the prior work [50, 173].

6.3.2 Multi-step depth forecasting results

The findings of the study reveal that the proposed method exhibits a faster convergence rate, with a reduced number of epochs compared to prior works. Specifically, the proposed method achieves convergence in just 6 epochs, whereas previous approaches required 20 epochs.

It is of particular interest to examine Table 6.1, which displays the performance of the proposed method across different predicted future steps. Notably, it can be observed that as time progresses, the uncertainty of the future increases, leading to larger errors in the predictions.

Furthermore, Table 6.1 presents the results of comparing the depth forecasting of the proposed method with prior works. As expected, the proposed method outperforms the prior work, with a significant gap of $\Delta AbsRel = 7.3\%$. This finding is further substantiated by the qualitative observations portrayed in Fig. 6.4, where the generated output depth maps produced by our proposed method demonstrate superior precision and intricate details when compared to prior approaches. The forecasted depth at the different time steps is even comparable to other methods that does depth inference and have access to the target frame image. $t = 1$ it is better than [140, 182, 181, 189]. and $t = 5$ is better than [181, 189]. This demonstrates that the enforcing the spatio-temporal consistency

Evaluation	Model	Abs Rel	Sq Rel	RMSE	RMSE log
All points mean	ManyDepth [173]	0.098	0.770	4.459	0.176
	Boulahbal *et al* [14]	0.110	0.719	4.486	0.184
	Ours	0.110	0.805	4.677	0.187
	Ours + stereo	0.107	0.751	4.805	0.189
Only dynamic	ManyDepth [173]	0.192	2.609	7.461	0.288
	Boulahbal *et al* [14]	0.167	1.911	6.724	0.271
	Ours	0.178	2.089	6.963	0.278
	Ours + stereo	0.155	1.668	6.401	0.260
Only static	ManyDepth[173]	0.085	0.613	4.128	0.150
	Boulahbal *et al* [14]	0.101	0.624	4.269	0.163
	Ours	0.099	0.684	4.462	0.165
	Ours + stereo	0.099	0.684	4.679	0.173
Per category mean	ManyDepth[173]	0.139	1.611	5.794	0.219
	Boulahbal *et al* [14]	0.134	1,267	5,496	0,217
	Ours	0.138	1.386	5.712	0.222
	Ours + stereo	0.127	1.176	5.540	0.217

Table 6.2: Quantitative performance comparison for dynamic and static objects at $t = 00$. The proposed method outperforms the SOTA [173] on the dynamic objects. The stereo variant is the best model for the dynamic and the per category mean.

results in an accurate depth sequence.

6.3.3 Handling dynamic objects

In the interest of conducting a thorough analysis of the proposed method, it is important to consider its ability to handle dynamic objects in the scene. One limitation of a previous approach ([173]) was its inability to handle such objects, and thus we conducted an experiment to address this issue.

The proposed model will be evaluated against [173, 14] using the methodology introduced in [14]. Furthermore, a variant of the proposed method is employed which leverages stereo images during training. The pose network is completely discarded, and the extrinsic parameters of the stereo pair are used to warp the one view into the other. This variant model operates under the assumption of a rigid scene, thereby avoiding any issues related to warping dynamic objects.

The results, presented in Table 6.2, show that the proposed method outperforms the ManyDepth baseline on dynamic objects with a significant improvement of $\Delta AbsRel = 13.0\%$, and it has a comparable result when the unbiased per category mean is used. Although the proposed variant with stereo images during evaluation performs almost equally well for the static scenes, most of the improvement is observed on the dynamic objects. These findings suggest that although the proposed model is not explicitly trained on dynamic objects, it is able to learn their dynamics implicitly. One possible explanation for this is that the as model utilizes multi-scale attention, which allows it to

capture the motion of dynamic objects even without being specifically supervised to do so. This highlights the effectiveness of attention mechanisms in capturing spatio-temporal dependencies and modeling the dynamics of the scene. Overall, these experiments highlight the importance of handling the dynamic objects in depth inference and forecasting.

6.3.4 Depth inference generalization

In order to compare the proposed architecture with other methods that perform single depth-image inference, it is proposed to train the model only for this (output only depth at D_t). The comparison is only made with respect to methods that leverage multi-fame input for the depth network. Prior methods [173, 57] perform a plane sweep operation to compute a cost volume. The plane sweep algorithm explicitly uses the pose of the scene and requires the camera intrinsic parameters. The proposed depth network model, on the other hand, performs the matching implicitly using the transformers and does not depend on any other network.

Table 6.1 shows the comparison results of the proposed method with ManyDepth [173] on KITTI benchmark [49]. As expected, the models that explicitly use the pose information have better performance on the KITTI benchmark for depth inference. These assessments, however, do not evaluate the ability of the networks to generalize to new scenes. Therefore, a generalization study was performed to better assess the models in this respect:

- **Testing the domain gap:** The models pre-trained on the KITTI dataset are directly evaluated on the Cityscapes [35] dataset without retraining.
- **Testing the sensibility to the camera parameters:** The focal length of the camera is replaced with a focal length $f = 1$ and the optical center is chosen as $(\frac{W}{2}, \frac{H}{2})$. This evaluation is performed on the KITTI dataset.
- **Testing weather perturbation:** The evaluation is done on 3 sequences of the Robotcar dataset: overcast, snow and rain sequences.

As observed in Fig. 6.5 the proposed method outperforms the baselines for these generalization settings. This suggests that while the baselines are able to perform better on the KITTI dataset, they do generalize better in other settings. This could be explained by the fact that the generalization of these methods depends on both the generalization of the pose and depth network. The proposed model, on the other hand, performs matching implicitly using the attention of transformers and does not depend on any other network, which makes it less sensitive to variations in pose and camera parameters.

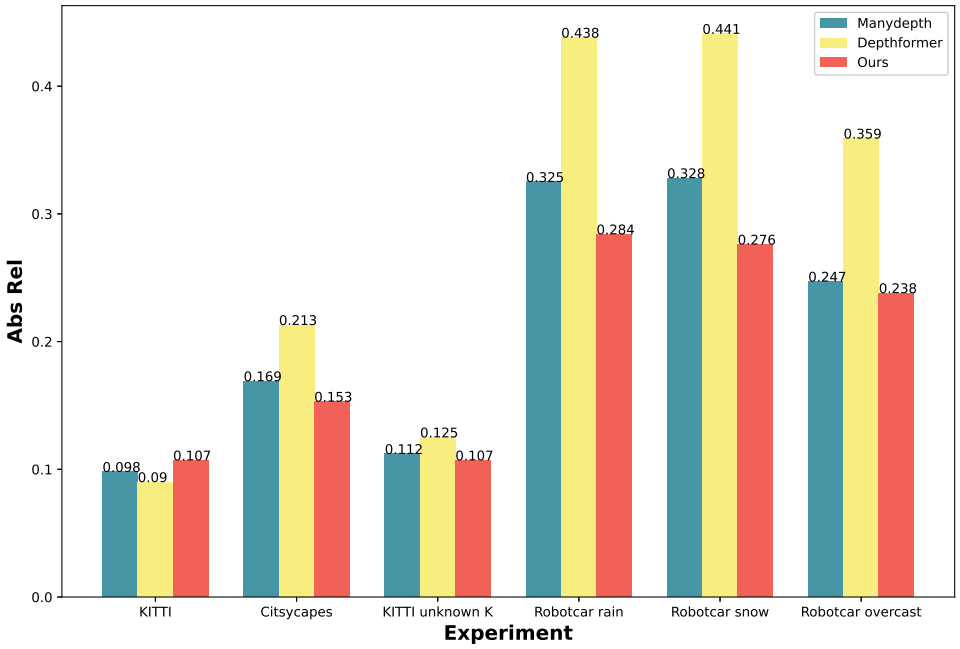


Figure 6.5: Depth inference generalization study. The proposed architecture is compared to Many-Depth and DeptFormer on different generalization scenarios: Domain gap evaluation on Cityscapes, sensitivity to camera parameters, and weather perturbations on the Robotcar dataset. As shown, the proposed method outperforms the baselines in all three generalization settings, suggesting its ability to generalize well to different scenarios.

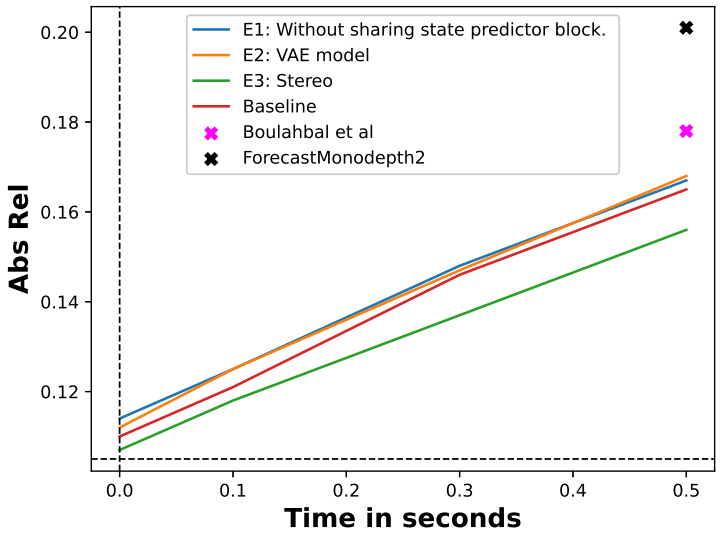


Figure 6.6: Ablation studies for improving depth forecasting performance. The *Abs Rel* performance of various evaluated models is shown in the figure. (i) E1, tests the model without sharing the state predictor block. (ii) E2, involves the use of a VAE model to output multi-hypothesis future depth (iii) E3, assess the model with the stereo pose.

6.3.5 Ablation study

Several ablations were performed in an effort to improve the performance of the proposed model. Figure 6.6 displays the *Abs Rel* performance of the various evaluated models. Specifically, the following ablation studies were carried out:

- E1: Tests the model without sharing the state predictor block. As observed, sharing the state predictor helps the model to output a better depth as multiple passes helps the network to generalize better.
- E2: This experiment involved the use of a VAE model, where the latent variables of the state predictor block were assumed to follow a Gaussian distribution. The aim of this experiment was to output a multi-hypothesis future depth. However, the first observation we made was that the model collapsed to a single modality and failed to output multiple hypotheses. As the decoder was perturbed with the Gaussian distribution, the output is less accurate with respect to the baseline.
- E3: Aim to assess the model with dynamic objects. More details are provided in Sec. 6.3.3

These experiments demonstrate that the proposed method holds several advantages: providing a spatial-temporal consistent depth sequence that represents present and future depth, superior depth

forecasting compared to the prior work, and better generalization for depth inference.

6.4 Discussion

In conclusion, this chapter presents a novel self-supervised model that predicts a sequence of future frames from video-input using a spatial-temporal attention (ST) network. The proposed model outperforms existing baselines on the KITTI benchmark for depth forecasting and achieves highly accurate and robust depth inference results. The novelty of the proposed model lies in its use of a transformer-based multi-frame architecture that implicitly learns the geometry and motion of the scene, while also leveraging prior scene knowledge such as object shape and texture. Furthermore, the proposed model enforces spatio-temporal consistency across a sequence of output frames rather than a single output frame, resulting in more accurate and robust depth sequence forecasting. Several ablation studies were conducted to assess the effectiveness of the proposed techniques. The proposed model provides a significant contribution to the field of depth prediction, and holds great promise for a wide range of applications in computer vision. Future research could explore the generation of accurate multi-hypotheses future depth, building upon the promising results presented in this chapter.

Chapter 7

Conclusion

7.1 Summary of the thesis

In this thesis, I have thoroughly examined the potential of self-supervised approaches for depth prediction and demonstrated their ability to provide a rich representation of a scene, enabling a more comprehensive understanding of motion and geometry. The task of predicting the future depth of a scene is undoubtedly challenging, yet it has significant implications for intelligent systems, particularly in the field of autonomous driving (AD) and advanced driver assistance systems (ADAS). The research presented in this thesis addressed the challenges of depth prediction using self-supervised learning techniques. Various scenarios were explored:

- In Chapter 3, the generalization of deep learning models was explored. This was done through domain adaptation methods, by utilizing generative adversarial networks, specifically conditional GANs for style transfer. During this exploration, a fundamental limitation of cGANs was revealed, superficially their lack of complete conditionality. To address this issue, the chapter presents an innovative solution referred to as the "*a contrario* method". The main objective of the *a contrario* method is to enhance conditional GANs and empower them with full conditionality.

- Chapter 4 explored image-to-depth map inference and extended the classical methods with dynamic objects. We have presented a solution for the static-scene assumption of the classical SFM model, using a novel transformer-based method that outputs a pose for each dynamic object.

- Chapter 5 explored video-to-depth mapping. This was the first attempt to forecast the future depth using self-supervision. The proposed model used a sequence of the past and present frames, and the model outputs a depth map that represents the future depth at step k. A

novel transformer-based architecture was proposed to aggregate the temporal information, this enabled the network to learn a rich spatio-temporal representation.

- Chapter 6 presented video-to-video depth. This model takes a sequence of images of past and present images and outputs a sequence of the present and future depth maps. This method addressed the limitations of the previous methods and extended the forecasting into a sequence of future depth. We have presented our self-supervised model that simultaneously predicts a sequence of future frames from video input with a novel spatial-temporal attention (ST) network.

Accurately predicting future depth can help these systems better anticipate and react to changes in their environment, which is crucial for their safe and effective operation. The applications of self-supervised depth prediction extend beyond AD and ADAS, as this method offers an efficient way to enable good understanding of videos. Given the promising results of self-supervised depth prediction, it is worth considering the possibility of applying this technique at scale to create vision models akin to the popular GPT-4 language model. Such models would have broad applications, ranging from autonomous systems to robotics, and beyond.

7.2 Perspective and future work

To build upon the findings of this thesis, future research could explore the following areas:

- **Predicting multiple plausible future depth:**
  Depth prediction self-supervision uses a differentiable warping and an image reconstruction as pretext task. This warping assume is that the environment is deterministic and that there is only one possible future. However, the future is stochastic by nature. The uncertainty of motion grows with time and there exists a multiple possible outcomes. In Fig. 7.1 the pedestrians could decide to cross the road or not. While the past context could provide a hint of the future actions of the pedestrian, his actions are not deterministic. Therefore, generating multiple hypotheses for the future is crucial for path planning and for safety applications. In Chapter 6, We tried to use a VAE to model to generate multiple future scenarios, but we observed that the model had collapsed into a single mode. We suspect that the warping function and the image reconstruction with a deterministic video (there is only one scenario observed) makes the training collapse into the single mode (*i. e.* the most likely mode based on the context frames). One solution is to train on a dataset with multiple hypothesis. We believe that this avenue of research will have a high impact on safety applications.

- **Scaling the training for large-scale datasets:**
  As we mentioned in Chapter 3 the advantage of the self-supervision is the ability to train on

Figure 7.1: An example of the future multi-hypothesis. The pedestrians may or may not cross the street, the situation is uncertain.

large-scale dataset as only videos are required. The current results show that even for a small and limited dataset such as KITTI the model is able to obtain accurate depth with a closing gap with respect to the supervised methods. Training with large-scale datasets is challenging for data collection, storage, and especially training. However, this approach could enable the model to not only achieve outstanding performance and generalization, but also to attain a genuine scene understanding akin to GPT models. As such, this research direction has the potential to make substantial contributions to the field of computer vision.

- **Domain generalization:**
  While training on large scale-datasets could be a way to go forward with improving the generalization, it is possible to improve the generalization with other techniques like domain adaptation and style transfer. If we have prior knowledge on what domains the model will encounter, it is possible to adapt the model directly to these domains. However, generative techniques are susceptible to hallucination. One possible solution is to use the acontrario cGAN [11] to help the generative model to be consistent with the conditioning input and the depth model to generalize better. The pursuit of improved generalization in depth models is an ongoing endeavor with potential to improve downstream safety applications.

- **Improving depth with semi-supervised learning:**

Although this thesis has primarily focused on self-supervised methods, we believe that semi-supervised methods hold significant potential for enhancing performance and generalization in computer vision applications. The utilization of supervised labels provides a well-constrained signal for supervising the network and achieving improved accuracy in the predictions. Furthermore, self-supervision aids the network in achieving better generalization by enabling training on large-scale datasets that may feature significant domain shifts. The integration of semi-supervised methods into the training pipeline presents a promising research direction with the potential to advance the field of computer vision. However, to fully realize the potential of semi-supervised methods, additional research is required to investigate the optimal strategies for combining supervised and self-supervised methods.

In conclusion, this thesis has demonstrated the potential of self-supervised depth prediction as a powerful tool for enabling a more comprehensive understanding of scenes and their dynamics. The implications of this technique for intelligent systems, particularly in the field of autonomous driving and advanced driver assistance systems, are significant and promising.

Appendix A

Computer vision basics

Computer vision deals with how computers can understand and interpret images and videos. It involves the development of algorithms that can analyze and understand the environment represented as a set of **image**, Understanding this environment involves recognizing the entities and their **motion** and reasoning about their interactions. this enables to perform useful tasks such as object recognition, image classification, scene understanding, object tracking, 3D reconstruction and more. In the following section, we begin to define the rigid-body transformation and the process of acquiring images and the geometry of multiple-views.

A.1 Rigid-body transformation

In order to describe the motion of an object, in principle, the trajectory of all points of that objects should be specified. However, as this object do not have any deformation or change in its shape, specifying the motion of one point is sufficient. This known as rigid-body transformation. This type of transformation can include rotations, translations, and combinations of both. In the context of computer vision, rigid-body transformation is often used to describe the movement of objects within an image or video, and can be used to track the motion of those objects over time. It could be defined formally as :

Rigid-body transformation: A map $g : \mathbb{R}^3 \longrightarrow \mathbb{R}^3$ is a rigid-body transformation if it preserves the norm and the cross product of any two vectors :

1. norm : $||g(\mathbf{v})|| = ||\mathbf{v}||, \mathbf{v} \in \mathbb{R}$.
2. cross product: $g(\mathbf{v}) \times g(\mathbf{u}) = g(\mathbf{v} \times \mathbf{u})\mathbf{v}, \mathbf{u} \in \mathbb{R}^3$.

Rigid-body transformation can include rotations, translations, and combinations of both. For example, A point in the world frame at instance p_1^w can be transformed with a rotation and translation

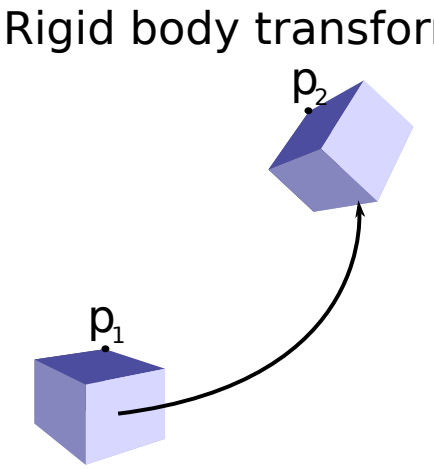


Figure A.1: An example of a rigid-body transformation

as shown in the Fig. A.1. The equation that relates the two points can be expressed as:

$$p_2^w =^2 R_1 p_1^w +^2 T_1 \tag{A.1}$$

where ${}^2R_1$ is the rotation matrix and ${}^2T_1$ is the translation matrix.

A.1.1 Rotation matrix representation

The rotation matrix is 3×3. However, a rotation have only 3-DOF. Therefore, this 9 parameters matrix could be expressed using only 3 parameters. There exists several minimal parametrization of the rotation matrix such as Euler angles, quaternions and axis-angle parametrization.

- **Euler angles :** a commonly used method for rotation representation, where a rotation is decomposed into three consecutive rotations around different axes, as shown in Fig. A.2. The simplicity and ease of interpretation of Euler angles make them a popular choice in many applications. However, it is important to note that Euler angles are not unique and can produce the same rotation with different parametrization depending on the order of the rotations.

- **Axis-angle representation** It represents the rotations with a single angle and axis of rotation. The axis of rotation is defined as a unit vector in 3D space, and the angle represents the magnitude of rotation about this axis. Fig. A.3 represents an example of axis-angle rotation convention. The rotation is parameterized by the vector $\boldsymbol{\theta} = \theta\mathbf{e}$ where the vector $\mathbf{e}$ gives the direction and θ is a scalar that gives the angle.

Axis-angle representation of rotation is generally considered to be better than Euler angles in several ways:

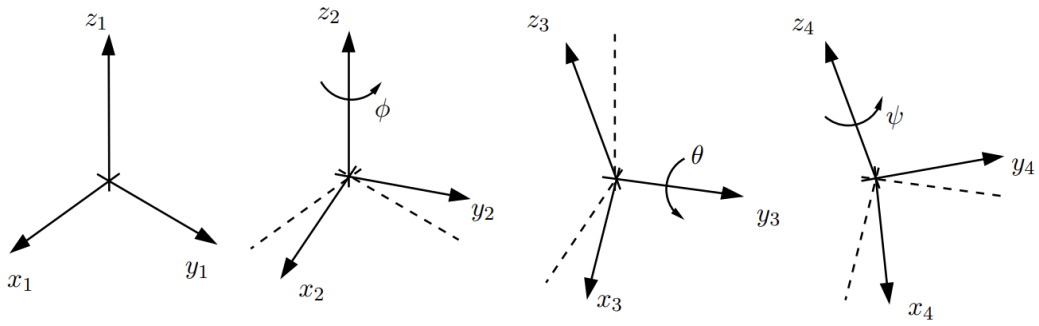


Figure A.2: An example of an Euler rotation representation $Z_1Y_2Z_3$. Consider a Cartesian coordinate. In order to define Euler angles, three canonical rotations are applied. First rotate around the z-axis by ϕ, then around the new y-axis by $\boldsymbol{\theta}$, and finally around the new z-axis by ψ.

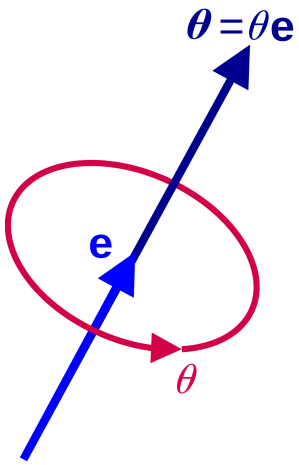


Figure A.3: An example of the axis-angle rotation convention. The rotation is parameterized by the vector $\boldsymbol{\theta} = \theta\mathbf{e}$ where the vector $\mathbf{e}$ gives the direction and θ is a scalar that gives the angle.

- **Unique representation:** Axis-angle provides a unique representation for a rotation, while Euler angles can lead to singularities and result in multiple solutions for a single rotation.
- **Avoiding Gimbals lock:** Euler angles can suffer from Gimbals lock, a phenomenon where two of the rotational degrees of freedom become locked to each other, causing the rotation to become ambiguous. Axis-angle does not suffer from Gimbals lock.

A.1.2 Homogeneous representation

Since Eq. A.1 combine an addition and multiplication, it is possible to represent that equation with a single matrix multiplication. This is achieved by introducing a homogeneous vector, which is a 4-dimensional vector that includes an additional element of 1 at the end $\mathbf{p} = (x, y, z, 1)$.

The homogeneous transformation is represented as a 4×4 matrix, where the first 3×3 elements represents the rotation and the last column represents the translation. This matrix can be used to transform a $3D$ point in one coordinate system to another coordinate system by matrix multiplication. The advantage of using a homogeneous transformation is that it allows for a compact and efficient way to represent both rotation and translation, as well as combining multiple transformations into a single matrix multiplication.

$$\mathbf{p}_1 = \mathbf{T}\mathbf{p}_2 = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_3 & 1 \end{bmatrix} \mathbf{p}_2 \tag{A.2}$$

$\mathbf{T}$ is the homogeneous matrix that transforms the homogeneous point p_1 into the point $\mathbf{p}_2$

It is important to note that the pose of an object refers to its position and orientation in space. It is often represented by a combination of translation (x, y, z) and rotation (roll, pitch, yaw) parameters, or as a 4×4 transformation matrix that describes the same information. The rigid-body transformation, on the other hand, is a mathematical operation that describes how points in one coordinate system can be transformed to another coordinate system, while preserving the distances and angles between points. It is often used to describe the relationship between two different coordinate systems. Therefore, the pose of an object or a camera describes its location and orientation in a particular coordinate system, while the rigid-body transformation describes how to transform points between two different coordinate systems.

A.2 Pinhole camera model

A camera is a device that captures images by detecting and measuring the intensity of electromagnetic radiation, such as light. It consists of a lens and a light sensor. The lens is used to control the direction and intensity of the incoming light, while the light sensor measures the amount of light that falls on it and converts it into an electrical signal. This measurement, known as irradiance, is a measure of the power per unit area of the light incident on the sensor, and it is typically expressed in watts per square meter (W/m^2). There are several types of cameras that use different approaches to capture and process images, including pinhole cameras, fish-eye cameras, and event-based cameras. Pinhole cameras are the most common and widely used type of camera, and they are found in a wide range of applications, including smartphones, webcams and even cars. These cameras are cheap and well-documented.

An image is a representation of the visual perception of the world. This representation encodes the world in a $2D$ array of pixels. Each of these pixels stores the color intensity for that location of

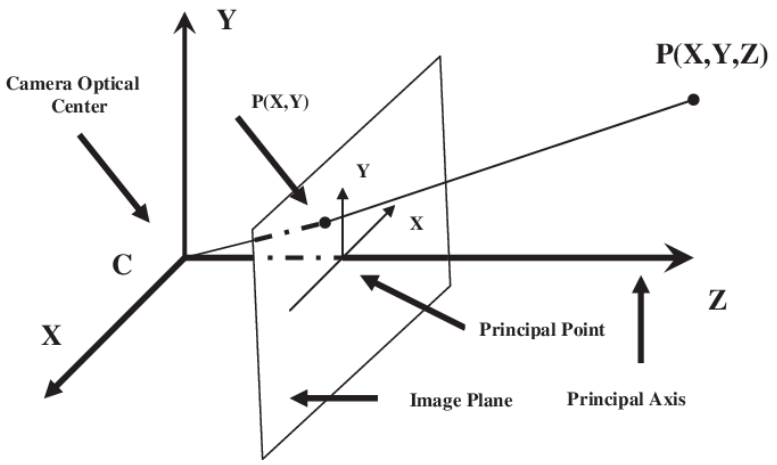


Figure A.4: Fronto-prallel pinhole camera model. The point P in the world frame is projected to the image coordinates. This process is defined using the extrinsic matrix that relates the world frame and camera frame, The projection to the image frame using the focal length and The image frame transformation using the optical center.

the image. More Formally, an image is mapping I that assign to a location (x, y) a positive value :

$$\mathbf{I}(x, y) : \mathbb{R}^2 \Longrightarrow \mathbb{R}^{3+} \tag{A.3}$$

An RGB image stores the color intensity of red, green and blue. For a digital image, the value of the intensity $\mathbf{I}(x, y)$ is discretized in 8 or 16 bit representation. The domain of (x, y) is also discretized $(x, y) \in ((0, W) \in \mathcal{N}, (0, H) \in \mathcal{N})$. Where W is the width and H is the height of the image.

Geometric model for pinhole camera image formation

In order to accurately model and predict the behavior of a pinhole camera, it is necessary to define a mathematical model that describes the relationship between the input light and the output image. This model takes into account the properties of the lens and sensor. By understanding and applying this mathematical model, it is possible to establish a correspondence between the points in the 3D space and their project 2D image. The pinhole camera model is shown in Fig. A.4

One mathematical model that can describe the image formation includes 3 transformations :

1. **Projection into camera frame:** it transforms the point from the world frame into the camera frame : if P_c have the coordinates $P_w = [X_w, Y_w, Z_w]$ we could obtain the coordinates of this point relative to the camera frame given by the rigid body transformation :

$$\mathbf{P_c} =^c \mathbf{T}_w \mathbf{P}_w = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_3 & 1 \end{bmatrix} \mathbf{P}_w \tag{A.4}$$

2. **Projection into the image coordinate:** Using the fronto-parallel pinhole camera model, the 3D point P_c is projected to the image frame coordinates :

$$\mathbf{p} = \begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z} \begin{bmatrix} X \\ Y \end{bmatrix} \tag{A.5}$$

$$Z\mathbf{p} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{K}_f \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{A.6}$$

f represents the focal length of the camera.

3. **Coordinates transformation from normalized coordinates to pixel coordinates:** first converting from metric to pixels and converting the origin to be the top-left of the image. This transformation can be expressed as :

$$Z \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & s_{\boldsymbol{\theta}} & o_x \\ 0 & s_y & o_x \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{A.7}$$

s_x and s_y converts the metric coordinates into the pixel coordinates. $s_{\boldsymbol{\theta}}$ is the *skew* factor usually close to zero for digital cameras. o_x and o_y are the coordinates of the optical center.

In summary, the pinhole camera model can be defined as follows:

$$Z \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} fs_x & fs_{\boldsymbol{\theta}} & o_x \\ 0 & fs_y & o_x \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_3 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{A.8}$$

$$Z\mathbf{p}' = \mathbf{K}\,\mathbf{\Pi}\,\mathbf{T}\,\mathbf{P}_w \tag{A.9}$$

Where $\mathbf{K}$ is the intrinsic camera matrix. $\mathbf{\Pi}$ is the projection matrix, and $\mathbf{T}$ is the extrinsic parameters.

A.2.1 Epipolar geometry

Epipolar geometry is a mathematical concept that describes the relationship between two views of the same scene. Consider two images ($\mathbf{I}_1$, $\mathbf{I}_2$) of the same scene from a different view. If a point $\mathbf{X}$ have coordinates $\mathbf{x}_1$ and $\mathbf{x}_2$ relative to the frames of each camera and ${}^1T_2$ is the pose of the second

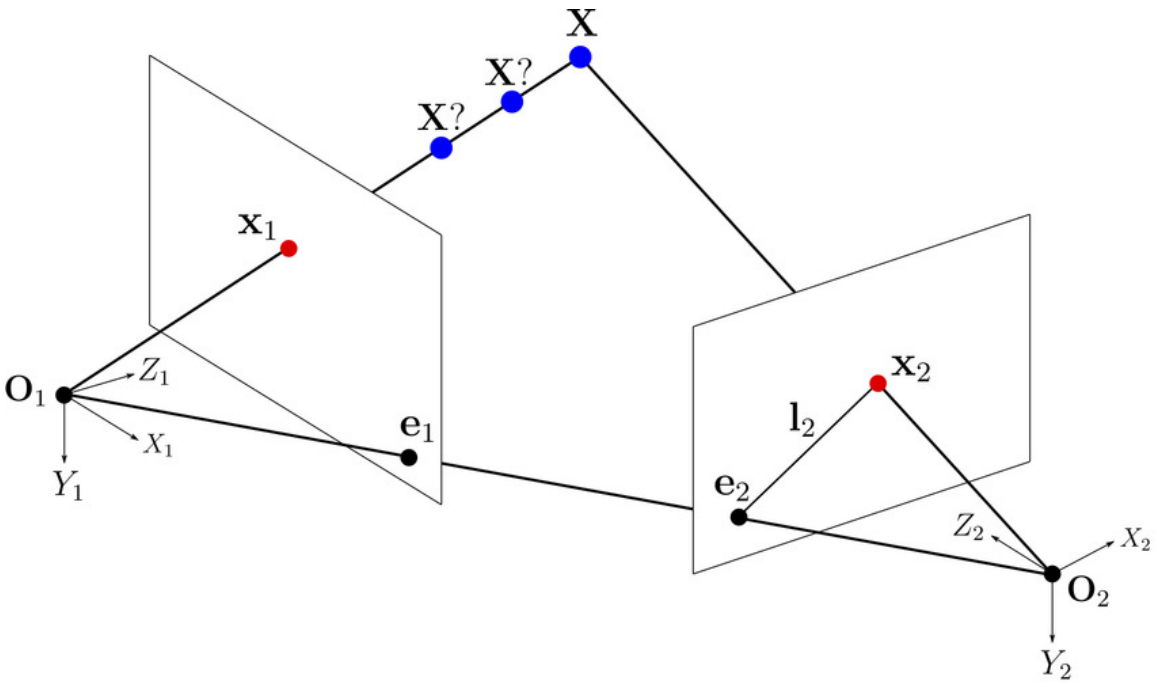


Figure A.5: Illustration of the epipolar geometry model of two cameras with optical centers O_1 and O_2. The point X is projected as x_1 for the first camera and x_2 for the second camera. The epipoles are defined at the intersection of the image planes and the plane (O_1, O_2, X). The projection of the line (O, x_1) on the other camera is called the epipolar line. The corresponding point x_2 is situated at that line.

camera with respect to the first then:

$$\mathbf{x}_1 =^1 T_2 \mathbf{x}_2 \tag{A.10}$$

Epipolar constraint: The epipolar constraint that relates the two images x_1 and x_2 is defined as follows:

$$\mathbf{x}_2^T \hat{\mathbf{T}} \mathbf{R} \mathbf{x}_1 = 0$$

The matrix

$$\mathbf{E} = \hat{\mathbf{T}} \mathbf{R}$$

is called the essential matrix. It encodes the relative pose of the two cameras. Fig. A.5 show the projection, the point $\mathbf{X}$ in the view. The intersection of the line (o_1, o_2) is called **epipoles** denoted by e_1, e_2. The lines l_1 and l_2 are called the **epipolar lines** which are the intersection of the plane (O_1, O_2, X) with the two image plane.

The epipolar geometry is a powerful tool for establishing correspondences between stereo pairs and resolving the scale ambiguity in depth prediction with deep learning. It is possible to project a

IR pattern to replace one of the images and to establish the correspondence based on the captured pattern and the ground truth pattern, this is the principle of an active depth sensor.

A.2.2 Classical methods for depth prediction

In computer vision, depth perception has been studied extensively as a means of understanding and interpreting visual scenes. We could distinguish two approaches for depth prediction: monocular methods and multi-view methods

Classical methods for monocular depth prediction

The depth is recovered from the motion of the camera. The scene is captured from different view and by knowing the relative position of camera it possible to recover the depth up to a certain scale. This algorithm is known as Structure from motion (SFM). Here is a typical structure from motion algorithm :

1. Load a set of images and detect keypoints and extract the descriptors (such as SIFT or ORB) for each image.
2. Find correspondences between the keypoints of different images using descriptor matching.
3. Estimate the fundamental matrix for each pair of images with correspondences.
4. Compute the essential matrix for each pair of images.
5. Use the essential matrix to compute the camera pose for each image.
6. Triangulate the 3D positions of the corresponding points using the camera pose for each image.

Classical methods for multi-view depth prediction

One example to perform multi-view depth prediction is stereo matching. It is a computer vision technique used to estimate the 3D structure of a scene from two or more images taken from different viewpoints. It involves finding corresponding points between the images and using these correspondences to compute the depth of each point in the scene.

There are many algorithms for stereo matching, but one of the most popular is the block matching algorithm. This algorithm works by dividing each image into small blocks, and then comparing the blocks from one image to the blocks in the other image to find the best match. The difference in position between the matching blocks is used to estimate the depth of the points in the scene. Here is a typical stereo matching algorithm :

1. Load two images of the same scene taken from different viewpoints.
2. Rectify the images.

3. Set the search window size and block size.
4. For each block in the left image:
   (a) Search for the best matching block in the right image within the search window.
   (b) Calculate the difference in position between the matching blocks.
   (c) Use the difference in position to estimate the depth of the points in the block.
5. Repeat the process for each block in the right image.

These classical methods are limited in their ability to handle complex and varied scenes, and are prone to errors and ambiguities. With the advent of deep learning, it has become possible to learn more robust and effective features for depth perception from large amounts of data. Instead of relying on hand-crafted features that may not encode relevant information for depth prediction. Deep learning approaches learn from the data the optimal features for depth prediction. These methods have achieved significant progress and outperform classical methods.

Appendix B

Acontrario conditional GAN

Supplementary material is presented here as follows, Section B.1 provides an additional evaluation of mode collapse for the depth prediction model. Section B.2 looks into the choice of weighting the different parts of the proposed loss function. Details are provided for reproducibility in Section B.3. Finally, an analysis of the training procedure is provided in Section B.4 to show that the training procedures did not encounter any degenerate situations.

B.1 Mode collapse analysis

Mode collapse is the setting in which the generator learns to map several inputs to the same output. A collapsing model is by construction unconditional. Only a few measures have been designed to explicitly evaluate this issue [145, 172, 3]. MS-SSIM [171, 172] measures a multi-scale structural similarity index and birthday paradox [3] concerns the probability that, in a set of n randomly chosen outputs, some pair of them will be duplicates. Another approach, NDB [145], presents a simple method to evaluate generative models based on relative proportions of samples that fall into predetermined bins.

The analysis provided in this section is an extension of the experiments done on depth prediction. Figure B.1 shows the evolution of the NDB measure over training iterations using the NDB score (the less, the better) for both pix2pix baseline and *a contrario* cGAN models trained on the NYU Depth V2 training set [124]. Out of the 12 trained models, the best model (in terms of RMSE log) is chosen for the evaluation. For clustering and evaluating NDB, non overlapping patches of 64×64 are considered. At the end of the training the NDB/k ($k = 100$) of the *a contrario* cGAN is 0.550 while the baseline achieves only 0.645. This indicates that *a contrario* model **generalizes better**. This is also observed qualitatively in Figure 3.8. Training with the counter examples helps the discriminator to model conditionality. Thus, the generator search space is restricted to only conditional space. The generator is penalized for non-conditionality even if the generation is realistic.

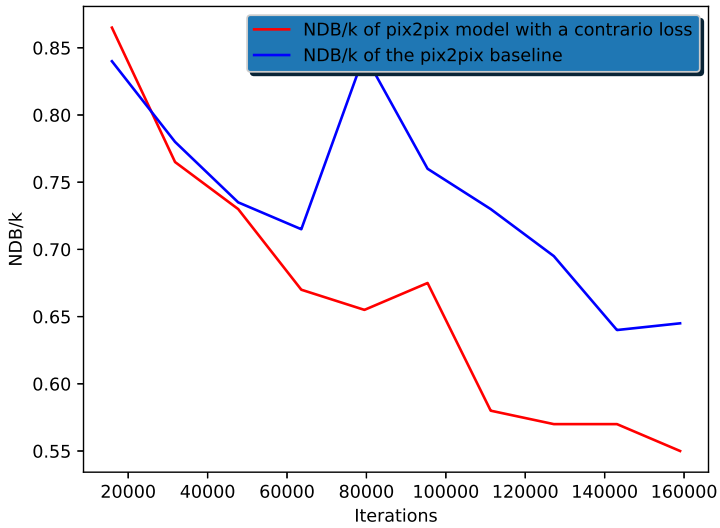


Figure B.1: An analysis of mode collapse using the NDB criteria (lower values are better) throughout training on the NYU depthV2 dataset. It can be concluded from this evaluation that the proposed approach is much better at avoiding mode collapse due to the restricted search space of the generator.

B.2 Loss function analysis

An ablation study on Eq 3.7 was performed. Each term that contributes to the adversarial loss is weighted by λ_i. Eq 3.7 becomes:

$$\begin{aligned}\mathcal{L}_{adv} = \min_G \max_D \Big[\lambda_1 \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}),\mathbf{y}\sim p(\mathbf{y}|\mathbf{x})}\big[log(D(\mathbf{x},\mathbf{y})]\big] + \lambda_2 \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}\big[log[1-D(\mathbf{x},G(\mathbf{x}))]\big]\Big] + \\ \max_D \Big[\lambda_3 \mathbb{E}_{\tilde{\mathbf{x}}\sim p(\tilde{\mathbf{x}}),\mathbf{y}\sim p(\mathbf{y})}\big[log(1-D(\tilde{\mathbf{x}},\mathbf{y}))\big] + \lambda_4 \mathbb{E}_{\tilde{\mathbf{x}}\sim p(\tilde{\mathbf{x}}),\mathbf{x}\sim p(\mathbf{x})}\big[log(1-D(\tilde{\mathbf{x}},G(\mathbf{x})))\big]\Big]\end{aligned} \tag{B.1}$$

Three strategies were considered for the weighting. The models were trained on the Cityscapes label-to-image dataset with the same settings described earlier (Section 3.4.3). Figure B.2 shows the mIoU for different *a contrario* cGAN models trained with different choices for λ_i.

- **Strategy 1:** Equal contribution for each term : $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4$.
- **Strategy 2:** Balancing the "fake" and "true" contributions. Since there are 3 data pairings classified as fake and only 1 real pair as true, equal balancing of true/fake gives: $\lambda_1 = 1, \lambda_2 = \lambda_3 = \lambda_4 = 0.33$
- **Strategy 3:** Testing the significance of both *a contrario* error terms for fake and real images. In this case only 3 terms with real-a-contrario is tested : $\lambda_1 = \lambda_2 = \lambda_3 = 0.5, \lambda_4 = 0$.

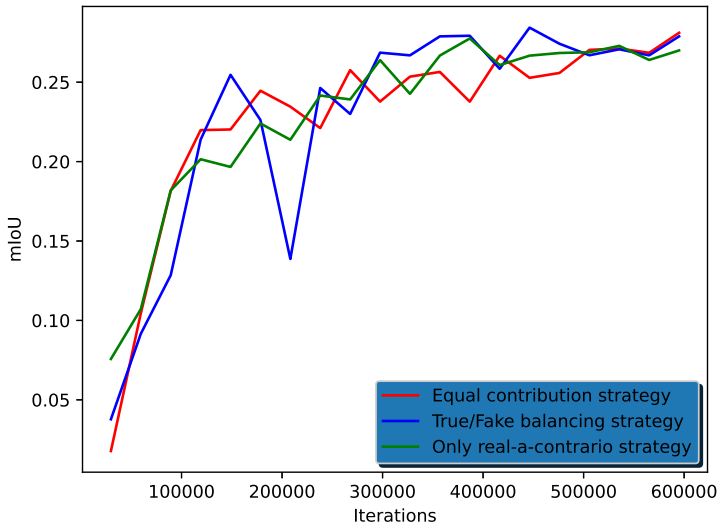


Figure B.2: The mIoU evaluation for different choice of λ_i. The strategy 1 of giving equal contribution yield the best results. However, there is no major difference on the convergence or the performances at epoch 200 between the different strategies

In this simple test, Strategy 1 gives the best results. Strategy 2 seems less stable. Strategy 3 succeeds to learn conditionality, however, it may not capture conditionality for generated images during training. Each of these strategies succeed to model conditionality, however, Strategy 1 converges faster and yields a better final result in terms of mIOU.

B.3 Reproducibility

Various experiments were performed using different datasets and input-output modalities. Some extra detail is provided here for reproducibility purposes. In all the experiments using the pix2pix baseline, random jitter was applied by resizing the 256×256 input images to 286×286 and then randomly cropping back to size 256×256. All networks were trained from scratch. Weights were initialized from a Gaussian distribution with mean 0 and standard deviation 0.02. The Adam optimizer was used with a learning rate of 0.0002, and momentum parameters $\beta_1 = 0.5$, $\beta_2 = 0.999$. A linear decay is applied starting from epoch 100, reaching 0 at epoch 200. Dropout is used during training. As in the original implementation [71], the discriminator is a PatchGan with a receptive field of 70×70. Similarly pix2pixHD [170], SPADE [130] and CC-FPSE [106] were trained with the same hyper-parameters as mentioned is their respective papers. For label-to-image, a U-Net256 with skip connections was used for the generator. A U-Net with 9 ResNet blocks was used for depth prediction, the last channel is 1 instead of 3 and the activation of the last convolution layer generator

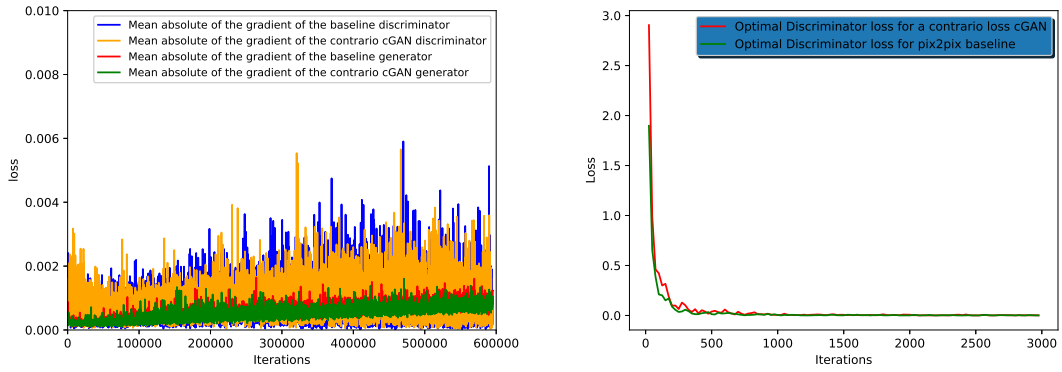


Figure B.3: (a) The mean absolute value of the gradients of the generator and discriminator for both baseline and *a contrario* cGAN models trained on Cityscapes[35]. The gradient is stable and it is neither vanishing nor exploding. (b) The loss function of the optimal discriminators when the generator is fixed. Both losses converge rapidly to 0.

is *Relu* instead of *Tanh*.

For the image-to-label task, a U-Net256 with skip connections was used for the generator but the output channel size was chosen to be 19 instead of 3 for segmentation of 19 classes. The activation of the last convolution layer of the generator was changed to a softmax to predict class probability for segmentation purposes.

B.4 Training details

Figure B.3(a) shows the gradient of the classic and proposed *a contrario* cGANs trained on Cityscapes [35] label-to-image with and without *a contrario* (see Section 3.4.3). The mean absolute value of the gradient is reported in order to demonstrate the stability of the training. Neither vanishing nor exploding gradient is observed for both models. Figure B.3(b) shows the training loss of the optimal discriminator trained as described in Section 3.4.1 for both models with the generator fixed at epoch 200. Both models converge rapidly to 0. Allowing the discriminator to converge for one epoch is enough to obtain the optimal discriminator with a fixed generator.

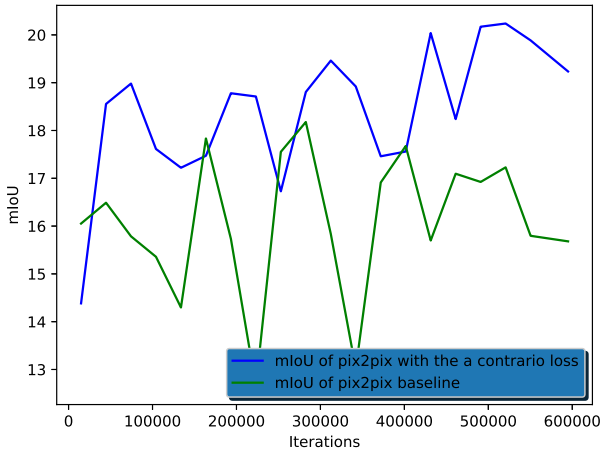


Figure B.4: mIoU for the Cityscape image-to-label dataset throughout training. The proposed method consistently obtains more accurate results and finishes with a largely different score at the end of training 19.23 versus for the baseline 15.97.

Bibliography

[1] Yasin Almalioglu, Muhamad Risqi U. Saputra, Pedro P.B. De Gusmao, Andrew Markham, and Niki Trigoni. GANVO: Unsupervised deep monocular visual odometry and depth estimation with generative adversarial networks. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume 2019-May, pages 5474–5480, 2019.

[2] Martín Arjovsky and L. Bottou. Towards principled methods for training generative adversarial networks. *ArXiv*, abs/1701.04862, 2017.

[3] Sanjeev Arora, Andrej Risteski, and Yi Zhang. Do gans learn the distribution? some theory and empirics. In *International Conference on Learning Representations*, 2018.

[4] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy Campbell, and Sergey Levine. Stochastic variational video prediction. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 2018.

[5] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[6] Apratim Bhattacharyya, Mario Fritz, and Bernt Schiele. Bayesian prediction of future street scenes using synthetic likelihoods. In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[7] Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. *Advances in neural information processing systems*, 32:35–45, 2019.

[8] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9157–9166, 2019.

[9] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[10] Houssem Boulahbal, Adrian Voicila, and Andrew Comport. Stdepthformer: Predicting spatio-temporal depth from video with a self-supervised transformer model. *arXiv preprint arXiv:2303.01196*, 2023.

[11] Houssem-eddine Boulahbal, Adrian Voicila, and Andrew Comport. Are conditional gans explicitly conditional? *arXiv preprint arXiv:2106.15011*, 2021.

[12] Houssem Eddine Boulahbal, Adrian Voicila, and Andrew I. Comport. Un apprentissage de bout-en-bout d'adaptateur de domaine avec des réseaux antagonistes génératifs de cycles consistants. In *Journée des Jeunes Chercheurs en Robotique*, Visioconference, France, November 2020.

[13] Houssem Eddine Boulahbal, Adrian Voicila, and Andrew I Comport. Forecasting of depth and ego-motion with transformers and self-supervision. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 3706–3713. IEEE, 2022.

[14] Houssem Eddine Boulahbal, Adrian Voicila, and Andrew I Comport. Instance-aware multi-object self-supervision for monocular depth prediction. *IEEE Robotics and Automation Letters*, 7(4):10962–10968, 2022.

[15] A. Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. *ArXiv*, abs/1809.11096, 2019.

[16] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 2020-December, 2020.

[17] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[18] C. Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A. Efros. Everybody dance now. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5932–5941, 2019.

[19] Hao Chen, Kunyang Sun, Zhi Tian, Chunhua Shen, Yongming Huang, and Youliang Yan. Blendmask: Top-down meets bottom-up for instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8573–8581, 2020.

[20] Liang-Chieh Chen, Maxwell Collins, Yukun Zhu, George Papandreou, Barret Zoph, Florian Schroff, Hartwig Adam, and Jon Shlens. Searching for efficient multi-scale architectures for dense image prediction. *Advances in neural information processing systems*, 31, 2018.

[21] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[22] Po-Yi Chen, Alexander H Liu, Yen-Cheng Liu, and Yu-Chiang Frank Wang. Towards scene understanding: Unsupervised monocular depth estimation with semantic-aware representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2624–2632, 2019.

[23] Qiang Chen, Anda Cheng, Xiangyu He, Peisong Wang, and Jian Cheng. Spatialflow: Bridging all tasks for panoptic segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(6):2288–2300, 2020.

[24] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1511–1520, 2017.

[25] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. *Advances in neural information processing systems*, 29:730–738, 2016.

[26] Wuyang Chen, Xianzhi Du, Fan Yang, Lucas Beyer, Xiaohua Zhai, Tsung-Yi Lin, Huizhong Chen, Jing Li, Xiaodan Song, Zhangyang Wang, et al. A simple single-scale vision transformer for object localization and instance segmentation. *arXiv preprint arXiv:2112.09747*, 2021.

[27] Xia Chen, Jianren Wang, and Martial Hebert. Panonet: Real-time panoptic segmentation through position-sensitive feature embedding. *arXiv preprint arXiv:2008.00192*, 2020.

[28] Yifeng Chen, Guangchen Lin, Songyuan Li, Omar Bourahla, Yiming Wu, Fangfang Wang, Junyi Feng, Mingliang Xu, and Xi Li. Banet: Bidirectional aggregation network with occlusion handling for panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3793–3802, 2020.

[29] Yuhua Chen, Cordelia Schmid, and Cristian Sminchisescu. Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7063–7072, 2019.

[30] Yuhua Chen, Cordelia Schmid, and Cristian Sminchisescu. Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-October:7062–7071, 2019.

[31] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*, 2022.

[32] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. *arXiv preprint arXiv:2112.01527*, 2021.

[33] Hsu Kuang Chiu, Ehsan Adeli, and Juan Carlos Niebles. Segmenting the Future. *IEEE Robotics and Automation Letters*, 5(3):4202–4209, 2020.

[34] Jaehoon Choi, Dongki Jung, Donghwan Lee, and Changick Kim. Safenet: Self-supervised monocular depth estimation with semantic-aware feature extraction. *arXiv preprint arXiv:2010.02893*, 2020.

[35] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[36] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.

[37] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[38] Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1(Mlm):4171–4186, 2019.

[39] Jinhao Dong and Tong Lin. Margingan: Adversarial training in semi-supervised learning. In *NeurIPS*, 2019.

[40] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[41] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[42] Vincent Dumoulin, Jonathon Shlens, and M. Kudlur. A learned representation for artistic style. *ArXiv*, abs/1610.07629, 2017.

[43] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in Neural Information Processing Systems*, 27:2366–2374, 2014.

[44] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems*, volume 3, pages 2366–2374, 2014.

[45] Farzan Farnia and A. Ozdaglar. Gans may have no nash equilibria. *ArXiv*, abs/2002.09124, 2020.

[46] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in Neural Information Processing Systems*, pages 64–72, 2016.

[47] Cheng-Yang Fu, Mykhailo Shvets, and Alexander C Berg. Retinamask: Learning to predict masks improves state-of-the-art single-shot detection for free. *arXiv preprint arXiv:1901.03353*, 2019.

[48] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European conference on computer vision*, pages 740–756. Springer, 2016.

[49] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[50] Clement Godard, Oisin Mac Aodha, Michael Firman, and Gabriel Brostow. Digging into self-supervised monocular depth estimation. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-October(1):3827–3837, 2019.

[51] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua:6602–6611, 2017.

[52] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[53] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2*, pages 2672–2680, 2014.

[54] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-Octob:8976–8985, 2019.

[55] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. In *Proceedings of the ieee/cvf International Conference on computer vision*, pages 6391–6400, 2019.

[56] Colin Graber, Grace Tsai, Michael Firman, Gabriel Brostow, and Alexander Schwing. Panoptic segmentation forecasting. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 2279–2288, 2021.

[57] Vitor Guizilini, Rareș Ambruș, Dian Chen, Sergey Zakharov, and Adrien Gaidon. Multi-frame self-supervised depth with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 160–170, 2022.

[58] Ishaan Gulrajani, Colin Raffel, and Luke Metz. Towards gan benchmarks which require generalization. *arXiv preprint arXiv:2001.03653*, 2020.

[59] Ruohao Guo, Dantong Niu, Liao Qu, and Zhenbo Li. Sotr: Segmenting objects with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7157–7166, 2021.

[60] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[61] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[62] Philipp Heise, Sebastian Klose, Brian Jensen, and Alois Knoll. Pm-huber: Patchmatch with huber regularization for stereo matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2360–2367, 2013.

[63] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, 2017.

[64] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008.

[65] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[66] Sara Hooker. The hardware lottery. *Communications of the ACM*, 64(12):58–65, 2021.

[67] Anthony Hu, Fergal Cotter, Nikhil Mohan, Corina Gurau, and Alex Kendall. Probabilistic Future Prediction for Video Scene Understanding. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 12361 LNCS, pages 767–785, 2020.

[68] X. Huang and Serge J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1510–1519, 2017.

[69] Junhwa Hur and Stefan Roth. Self-supervised monocular scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7396–7405, 2020.

[70] Seung-Jun Hwang, Sung-Jun Park, Joong-Hwan Baek, and Byungkyu Kim. Self-supervised monocular depth estimation using hybrid transformer encoder. *IEEE Sensors Journal*, 22(19):18762–18770, 2022.

[71] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2017.

[72] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28:2017–2025, 2015.

[73] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. *Advances in Neural Information Processing Systems*, 2015-Janua:2017–2025, 2015.

[74] J. Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.

[75] Adrian Johnston and Gustavo Carneiro. Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 4755–4764, 2020.

[76] Adrian Johnston and Gustavo Carneiro. Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 4756–4765, 2020.

[77] Minguk Kang and Jaesik Park. Contragan: Contrastive learning for conditional image generation. *arXiv: Computer Vision and Pattern Recognition*, 2020.

[78] Tero Karras, Miika Aittala, Janne Hellsten, S. Laine, J. Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *ArXiv*, abs/2006.06676, 2020.

[79] Ilya Kavalerov, Wojciech Czaja, and Rama Chellappa. A multi-class hinge loss for conditional gans. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1290–1299, 2021.

[80] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A convolutional network for real-time 6-dof camera relocalization. *Proceedings of the IEEE International Conference on Computer Vision*, 2015 Inter:2938–2946, 2015.

[81] Raymond A Kent. Estimation. *Data Construction and Data Analysis for Survey Research*, page 157, 2001.

[82] Minyoung Kim, Pritish Sahu, Behnam Gholami, and Vladimir Pavlovic. Unsupervised visual domain adaptation: A deep max-margin gaussian process approach. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2019-June, pages 4375–4385, 2019.

[83] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[84] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6399–6408, 2019.

[85] Marvin Klingner, Jan Aike Termöhlen, Jonas Mikolajczyk, and Tim Fingscheidt. Self-supervised Monocular Depth Estimation: Solving the Dynamic Object Problem by Semantic Guidance. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12365 LNCS:582–600, 2020.

[86] Naveen Kodali, James Hays, J. Abernethy, and Z. Kira. On convergence and stability of gans. *arXiv: Artificial Intelligence*, 2018.

[87] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). *""*, 2009.

[88] Manoj Kumar, Mohammad Babaeizadeh, Dumitru Erhan, Chelsea Finn, Sergey Levine, Laurent Dinh, and Durk Kingma. Videoflow: A conditional flow-based model for stochastic video generation. *arXiv preprint arXiv:1903.01434*, 2019.

[89] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiri Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8183–8192, 2018.

[90] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750, 2018.

[91] Seokju Lee, Sunghoon Im, Stephen Lin, and In So Kweon. Learning monocular depth in dynamic scenes via instance-aware projection consistency. *arXiv preprint arXiv:2102.02629*, 2021.

[92] Seokju Lee, Francois Rameau, Fei Pan, and In So Kweon. Attentive and contrastive learning for joint depth and motion field estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4862–4871, 2021.

[93] Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13906–13915, 2020.

[94] Chongxuan Li, T. Xu, J. Zhu, and B. Zhang. Triple generative adversarial nets. In *NIPS*, 2017.

[95] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. *arXiv preprint arXiv:2112.03857*, 2021.

[96] Wenbo Li, Pengchuan Zhang, Lei Zhang, Qiuyuan Huang, X. He, Siwei Lyu, and Jianfeng Gao. Object-driven text-to-image synthesis via adversarial training. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12166–12174, 2019.

[97] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Improved multiscale vision transformers for classification and detection. *arXiv preprint arXiv:2112.01526*, 2021.

[98] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2359–2367, 2017.

[99] Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017.

[100] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[101] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning Depth from Single Monocular Images Using Deep Convolutional Neural Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(10):2024–2039, 2016.

[102] Lingjie Liu, Weipeng Xu, M. Zollhöfer, H. Kim, F. Bernard, Marc Habermann, W. Wang, and C. Theobalt. Neural rendering and reenactment of human actor videos. *ACM Transactions on Graphics (TOG)*, 38:1 – 14, 2019.

[103] Ruiyang Liu, Yinghui Li, Dun Liang, Linmi Tao, Shimin Hu, and Hai-Tao Zheng. Are we ready for a new paradigm shift? a survey on visual deep mlp. *arXiv preprint arXiv:2111.04060*, 2021.

[104] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759–8768, 2018.

[105] Wei Liu, Andrew Rabinovich, and Alexander C Berg. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, 2015.

[106] Xihui Liu, Guojun Yin, Jing Shao, Xiaogang Wang, and hongsheng Li. Learning to predict layout-to-image conditional convolutions for semantic image synthesis. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[107] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[108] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

[109] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[110] Pauline Luc, Natalia Neverova, Camille Couprie, Jakob Verbeek, and Yann Lecun. Predicting Deeper into the Future of Semantic Segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2017-Octob, pages 648–657, 2017.

[111] Chenxu Luo, Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, Ram Nevatia, and Alan Yuille. Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2624–2641, 2019.

[112] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017.

[113] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Geometry-based next frame prediction from monocular video. In *IEEE Intelligent Vehicles Symposium, Proceedings*, pages 1700–1707, 2017.

[114] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised Learning of Depth and Ego-Motion from Monocular Video Using 3D Geometric Constraints. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 5667–5675, 2018.

[115] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

[116] Robert McCraith, Lukas Neumann, Andrew Zisserman, and Andrea Vedaldi. Monocular depth estimation with self-supervised instance adaptation. *arXiv preprint arXiv:2004.05821*, 2020.

[117] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. *arXiv preprint arXiv:2101.02702*, 2021.

[118] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[119] Tom M Mitchell and Tom M Mitchell. *Machine learning*, volume 1. McGraw-hill New York, 1997.

[120] Takeru Miyato and Masanori Koyama. cgans with projection discriminator. *ArXiv*, abs/1802.05637, 2018.

[121] Rohit Mohan and Abhinav Valada. Efficientps: Efficient panoptic segmentation. *International Journal of Computer Vision*, 129(5):1551–1579, 2021.

[122] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[123] Kevin P Murphy. *Probabilistic machine learning: an introduction*. MIT press, 2022.

[124] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[125] Weili Nie and A. Patel. Jr-gan: Jacobian regularization for generative adversarial networks. *ArXiv*, abs/1806.09235, 2018.

[126] Evangelos Ntavelis, A. Romero, I. Kastanis, L. Gool, and R. Timofte. Sesame: Semantic editing of scenes by adding, manipulating or erasing objects. *ArXiv*, abs/2004.04977, 2020.

[127] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *ICML*, 2017.

[128] OpenAI. Gpt-4 technical report, 2023.

[129] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[130] T. Park, Ming-Yu Liu, T. Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2332–2341, 2019.

[131] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[132] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[133] Horia Porav, Tom Bruls, and Paul Newman. Don't worry about the weather: Unsupervised condition-dependent domain adaptation. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 33–40. IEEE, 2019.

[134] Xiaojuan Qi, Zhengzhe Liu, Qifeng Chen, and Jiaya Jia. 3D motion decomposition for RGBD future dynamic scene synthesis. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2019-June, pages 7665–7674, 2019.

[135] Y. Qu, Yizi Chen, J. Huang, and Yuan Xie. Enhanced pix2pix dehazing network. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8152–8160, 2019.

[136] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[137] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

[138] Ruslan Rakhimov, Denis Volkhonskiy, Alexey Artemov, Denis Zorin, and Evgeny Burnaev. Latent video transformer. In *VISIGRAPP 2021 - Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, volume 5, pages 101–112, 2021.

[139] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[140] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2019-June, pages 12232–12241, 2019.

[141] Vitor Guizilini Rares, Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3D packing for self-supervised monocular depth estimation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2482–2491, 2020.

[142] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[143] S. Reed, Zeynep Akata, Xinchen Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In *ICML*, 2016.

[144] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[145] Eitan Richardson and Yair Weiss. On gans and gmms. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 5852–5863, 2018.

[146] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[147] Sadra Safadoust and Fatma Güney. Self-supervised monocular scene decomposition and depth estimation. In *2021 International Conference on 3D Vision (3DV)*, pages 627–636. IEEE, 2021.

[148] Tim Salimans, Ian J. Goodfellow, W. Zaremba, Vicki Cheung, A. Radford, and Xi Chen. Improved techniques for training gans. In *NIPS*, 2016.

[149] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser Nam Lim, and Rama Chellappa. Learning from Synthetic Data: Addressing Domain Shift for Semantic Segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3752–3761, 2018.

[150] Josip Saric, Marin Orsic, Tonci Antunovic, Sacha Vrazic, and Sinisa Segvic. Warp to the Future: Joint Forecasting of Features and Feature Motion. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 10645–10654, 2020.

[151] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[152] A. Srivastava, L. Valkov, Chris Russell, Michael U. Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In *NIPS*, 2017.

[153] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 573–580. IEEE, 2012.

[154] Weijie Su, Xizhou Zhu, Chenxin Tao, Lewei Lu, Bin Li, Gao Huang, Yu Qiao, Xiaogang Wang, Jie Zhou, and Jifeng Dai. Towards all-in-one pre-training via maximizing multi-modal mutual information. *arXiv preprint arXiv:2211.09807*, 2022.

[155] Vadim Sushko, Edgar Schönfeld, D. Zhang, Juergen Gall, B. Schiele, and A. Khoreva. You only need adversarial supervision for semantic image synthesis. *ArXiv*, abs/2012.04781, 2020.

[156] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[157] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[158] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020.

[159] Hao Tang, D. Xu, Yan Yan, P. H. S. Torr, and N. Sebe. Local class-specific and global image-level generative adversarial networks for semantic-guided scene generation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7867–7876, 2020.

[160] Adam M Terwilliger, Garrick Brazil, and Xiaoming Liu. Recurrent flow-guided semantic forecasting. In *Proceedings - 2019 IEEE Winter Conference on Applications of Computer Vision, WACV 2019*, pages 1703–1712, 2019.

[161] Thomas Unterthiner, Bernhard Nessler, G. Klambauer, Martin Heusel, Hubert Ramsauer, and S. Hochreiter. Coulomb gans: Provably optimal nash equilibria via potential fields. *ArXiv*, abs/1708.08819, 2018.

[162] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[163] Sudheendra Vijayanarasimhan, Susanna Ricco, Cordelia Schmid, Rahul Sukthankar, and Katerina Fragkiadaki. Sfm-net: Learning of structure and motion from video. *arXiv preprint arXiv:1704.07804*, 2017.

[164] H. D. Vries, Florian Strub, J. Mary, H. Larochelle, Olivier Pietquin, and Aaron C. Courville. Modulating early visual processing by language. In *NIPS*, 2017.

[165] Tuan Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Perez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2019-June, pages 2512–2521, 2019.

[166] Jacob Walker, Ali Razavi, and Aäron van den Oord. Predicting video with vqvae. *arXiv preprint arXiv:2103.01950*, 2021.

[167] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. PatchMatchNet: Learned multi-view patchmatch stereo. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, c:14189–14198, 2021.

[168] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV*, pages 108–126. Springer, 2020.

[169] T. Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, A. Tao, J. Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *NeurIPS*, 2018.

[170] T. Wang, Ming-Yu Liu, Jun-Yan Zhu, A. Tao, J. Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2018.

[171] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[172] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003.

[173] Jamie Watson, Oisin Mac Aodha, Victor Prisacariu, Gabriel Brostow, and Michael Firman. The temporal opportunist: Self-supervised multi-frame monocular depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1164–1174, 2021.

[174] Xinshuo Weng, Jianren Wang, Sergey Levine, Kris Kitani, and Nicholas Rhinehart. Inverting the pose forecasting pipeline with spf2: Sequential pointcloud forecasting for sequential pose forecasting. *arXiv preprint arXiv:2003.08376*, 2020.

[175] Gabriel Kreiman William Lotter and David Cox. Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning. In *Advances in Neural Information Processing Systems*, pages 64–72, 2016.

[176] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2, 2019.

[177] Dan Xu, Andrea Vedaldi, and Joao F Henriques. Moving slam: Fully unsupervised deep learning in non-rigid scenes. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4611–4617. IEEE, 2021.

[178] T. Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and X. He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1316–1324, 2018.

[179] Yihong Xu, Yutong Ban, Guillaume Delorme, Chuang Gan, Daniela Rus, and Xavier Alameda-Pineda. Transcenter: Transformers with dense queries for multiple-object tracking. *arXiv preprint arXiv:2103.15145*, 2021.

[180] Yanchao Yang and Stefano Soatto. FDA: Fourier domain adaptation for semantic segmentation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 4084–4094, 2020.

[181] Zhenheng Yang, Peng Wang, Wei Xu, Liang Zhao, and Ramakant Nevatia. Unsupervised learning of geometry from videos with edge-aware depth-normal consistency. In *Thirty-Second AAAI conference on artificial intelligence*, 2018.

[182] Zhichao Yin and Jianping Shi. GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1983–1992, 2018.

[183] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. *arXiv preprint arXiv:2106.04560*, 2021.

[184] H. Zhang, Vishwanath Sindagi, and V. Patel. Image de-raining using a conditional generative adversarial network. *IEEE Transactions on Circuits and Systems for Video Technology*, 30:3943–3956, 2020.

[185] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1947–1962, 2018.

[186] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[187] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.

[188] P. Zhou, Lingxi Xie, Xiaopeng Zhang, B. Ni, and Q. Tian. Searching towards class-aware generators for conditional generative adversarial networks. *ArXiv*, abs/2006.14208, 2020.

[189] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January:6612–6621, 2017.

[190] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.