

Introducing Generative Models to Facilitate Multi-Task Visual Learning

Zhipeng Bao

CMU-RI-TR-21-15

April, 2021



The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:

Martial Hebert, *chair*

Yu-Xiong Wang

Jun-Yan Zhu

Nadine Chang

*Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Robotics.*

Copyright © 2021 Zhipeng Bao. All rights reserved.

Abstract

Generative modeling has recently shown great promise in computer vision, but it has mostly focused on synthesizing visually realistic images. During my graduate study and research, motivated by multi-task learning of shareable feature representations, we consider a novel problem of learning a shared generative model that can facilitate multi-task visual learning.

We first start with a simple problem setting— learning a generative model for the joint task of few-shot recognition and novel-view synthesis: given only one or few images of a novel object from arbitrary views with only category annotation, we aim to simultaneously learn an object classifier and generate images of that type of object from new viewpoints. We focus on the *interaction and cooperation between a generative model and a discriminative model*, in a way that facilitates knowledge to flow across tasks in complementary directions. To this end, we propose *bowtie networks* that jointly learn 3D geometric and semantic representations *with a feedback loop*. Experimental evaluation on challenging fine-grained recognition datasets demonstrates that our synthesized images are realistic from multiple viewpoints and significantly improve recognition performance as ways of data augmentation, *especially in the low-data regime*.

Then, we further extend the bowtie network and propose a general multi-task oriented generative modeling (MGM) framework, by coupling a discriminative multi-task network with a generative network. While it is challenging to synthesize both RGB images and pixel-level annotations in multi-task scenarios, our framework enables us to use synthesized images paired with only weak annotations (*i.e.*, image-level scene labels) to facilitate multiple visual tasks. Experimental evaluation on challenging multi-task benchmarks, including NYUv2 and Taskonomy, demonstrates that our MGM framework improves the performance of all the tasks by large margins, consistently outperforming state-of-the-art multi-task approaches.

Acknowledgments

I would like to first give my great thanks to my advisors Prof. Martial Hebert and Prof. Yu-Xiong Wang for their guidance and mentor-ship over the two years. Martial has been guiding me how to do scientific research and how to make more valuable and solid work. He taught me the important role of carefulness and rigor in scientific research and also guided me to do every project step by step. I firmly believe that these experiences will greatly benefit me on my future research career. Yu-Xiong gives me detailed guidance in each of my projects. I learned from him how to think about research ideas, design the experiments, evaluate the approach, and how to present the final work. I wish I can become a qualified researcher like Yu-Xiong some day.

I also would like to appreciate the whole committee, Jun-Yan and Nadine are so kind and are always available to discuss and give useful feedback for my work. I also learned a lot from Jun-Yan's course *Learning Based Image Synthesis*, where I gained a lot of insights to potentially get involved in our framework.

To my family, I would like to express my gratitude for their constant support, both materially and spiritually. They trust me and always stand with me for all my decisions. I love them so much.

I spent two wonderful years at Pittsburgh, beyond research and study, I would thank my friends, especially Kelly Ning and Zifan Wang. Without them, I could not imagine what how boring what my life would be and how tough I was when I encountered an bottleneck. Thanks for the company.

Last, but not least, this work was supported in part by ONR MURI N000014-16-1-2007 and by AFRL Grant FA23861714660. We also thank NVIDIA for donating GPUs and AWS Cloud Credits for Research program.

Contents

1	Introduction	1
1.1	Generative Modeling for Joint View-Synthesis and Recognition	1
1.2	Generative Modeling for Multi-Task Visual Learning	4
2	Background	7
2.1	Few-Shot Recognition	7
2.2	Novel-View Synthesis	8
2.3	Multi-Task Learning and Task Relationship	8
2.4	Generative Modeling for Visual Learning	9
2.5	Feedback-Based Architectures and Task Model Learning	9
2.6	Reduced-Supervision Methods	10
3	Feedback-Based Network	11
3.1	Our Approach	11
3.1.1	Joint Task of Few-Shot Recognition and Novel-View Synthesis	11
3.1.2	Feedback-Based Bowtie Networks	12
4	Experimental Verification for FBNet	19
4.1	Experimental Setting	19
4.2	Main Results	20
4.2.1	View Synthesis Facilitates Recognition	20
4.2.2	Recognition Facilitates View Synthesis	21
4.2.3	Shared Generative Model vs. Shared Feature Representation .	23
4.3	Ablation Study	24
4.3.1	Different Recognition Networks	24
4.3.2	Categorical Loss	25
4.3.3	Resolution Distillation and Prototypical Classification	25
4.4	Qualitative Results on the CelebA-HQ Dataset	25
4.5	Discussion and Future Work	26
5	Multi-Task Oriented Generative Modeling	27
5.1	Problem Setting	27
5.2	Framework and Architecture	29
5.2.1	Multi-task Network (M)	29

5.2.2	Image Generation Network (G)	29
5.2.3	Refinement Network (R)	30
5.2.4	Self-supervision Network	31
5.3	Interaction Among Networks	32
5.3.1	Training Procedure:	33
6	Experimental Verification for MGM	35
6.1	Datasets and Compared Methods	35
6.2	Main Results	37
6.2.1	Quantitative Results	37
6.2.2	Qualitative Results	38
6.3	Ablation Study	39
6.3.1	Impact of Parameters	39
6.3.2	Impact of Self-supervision Task and Refinement Network . . .	39
6.3.3	Number of Synthesized Images vs. Real images	40
6.4	Extension	41
7	Conclusions	43
	Bibliography	45

List of Figures

1.1	Left: Given a single image of a novel visual concept (<i>e.g.</i> , a gadwall), a person can generalize in various ways, including imagining what this gadwall would look like from different viewpoints (top) and recognizing new gadwall instances (bottom). Right: Inspired by this, we introduce a general feedback-based bowtie network that facilitates the interaction and cooperation between a generative module and a discriminative module, thus simultaneously addressing few-shot recognition and novel-view synthesis in the low-data regime.	2
1.2	Left: Traditional multi-task learning framework (that learns shared feature representations) vs. Right: our proposed multi-task oriented generative modeling (that learns a shared generative model across various visual perception tasks)	4
3.1	Architecture of our feedback-based bowtie network. The whole network consists of a view synthesis module and a recognition module, which are linked through feedback connections in a bowtie fashion.	13
4.1	Synthesized images from multiple viewpoints. Images in the same row/column are from the same viewpoint/object. Our approach captures the shape and attributes well <i>even in the extremely low-data regime</i>	23
4.2	Ablation on λ_{cat} . Categorical loss trades off the performance between view synthesis and recognition.	24
4.3	Synthesized images by HoloGAN and FBNet on CelebA-HQ. Few-shot attributes (left to right): Black Hair, Gray Hair, Bald, Wearing Hat, and Aging. FBNet synthesizes images of higher quality and diversity.	26
5.1	Architecture of our proposed multi-task oriented generative modeling (MGM) framework. There are four main components in the framework: Multi-task network to address the target multiple pixel-level prediction tasks; self-supervision network to facilitate representation learning using images without any annotation; refinement network to perform scene classification using weak annotation; image generation network to synthesize useful images that benefit multiple tasks.	28

5.2	Joint training of the multi-task network and the image generation network. The multi-task network provides useful feature representation to guide the image generation process, while the generation network refines the shared representation through back-propagation.	32
6.1	Visualization and error comparison of the multi-task prediction outputs in the 50% data setting. The prediction results of MGM is quite close to the ground-truth, significantly outperforming the state-of-the-art results.	38
6.2	Performance change with different ratios of weakly labeled data. Joint learning significantly improves the performance. The performance of MGM keeps increasing with the number of the weakly labeled <i>synthesized</i> images, achieving results almost comparable to that of MGM_r trained with all the available weakly labeled <i>real</i> images. . . .	41

List of Tables

4.1	Top-1 (%) recognition accuracy on the CUB and CompCars datasets. For base classes: 150-way classification on CUB and 240-way classification on CompCars; for K -shot novel classes: 50-way classification on CUB and 120-way classification on CompCars. Our FBNet consistently achieves the best performance for both base and novel classes, and joint training significantly outperforms training each module individually.	21
4.2	Novel-view synthesis results under the FID and IS metrics. \uparrow indicates that higher is better, and \downarrow indicates that lower is better. As a reference, FID and IS of <i>Real Images</i> represent the best results we could expect. FBNet consistently outperforms the baselines, achieving 18% improvements for FID and 19% for IS.	22
4.3	Few-shot recognition accuracy consistently improves with different feature extraction networks.	24
4.4	Ablation studies on CUB regarding (i) learning a shared feature representation through standard multi-task learning, (ii) FBNet without resolution distillation, and (iii) FBNet using a regular classification network without prototypical classification. Our full model achieves the best performance.	24
6.1	Main results (mean \pm std) on the NYUv2 and Tiny-Taskonomy datasets. SS: semantic segmentation; DE: depth estimation; SN: surface normal prediction. \uparrow means higher is better; \downarrow means lower is better. We use different metrics on the two datasets, following existing protocol. Our MGM consistently and significantly outperforms both single-task (ST) and multi-task (MT) baselines, <i>even reaching the performance upper-bound of training with weakly annotated real images</i> (MGM _r).	37

6.2	Comparison of our MGM model with its variants. $MGM_{/G}$: <i>without</i> generating synthesized images; $MGM_{/j}$: <i>without</i> joint learning. Our MGM outperforms single-task and multi-task baselines <i>even without synthesized data</i> , showing its effectiveness as a general multi-task learning framework. The model performance further improves with joint learning.	39
6.3	Ablation study. (1) ST_1 and MT_1 : baselines with a larger number of parameters (with deeper backbones); (2) $MGM_{/self}$: <i>without</i> self-supervision task; (3) $MGM_{/refine}$: <i>without</i> classification refinement network; and (4) MGM_{recon} : <i>with</i> a simple reconstruction task as self-supervision. The two proposed components are complementary and both benefit the multiple tasks. The refinement network works better for surface normal; the self-supervision network works better for semantic segmentation. Their combination achieves the best. . . .	40
6.4	Mean test losses for six tasks on Tiny-Taskonomy. Again, our MGM outperforms the baselines, indicating its flexibility, generability, and scalability.	41

Chapter 1

Introduction

Seeing with the mind’s eye — creating internal images of objects and scenes not actually present to the senses, is perhaps one of the hallmarks in human cognition [63]. For humans, this visual imagination integrates learning experience and facilitates learning by solving different problems [21, 22, 62, 63]. We argue that achieving similar level of generalization is a crucial but important problem for machine vision. Therefore, during my graduate research and study, we aim to achieve similar level of generalization for machine vision— learning a shared generative model that is useful for different visual tasks. We first start with a simple problem: introducing generative modeling to facilitate the simple recognition task. Then we further considered learning a shared generative model for multiple visual tasks. In the following sections, we will further explain the motivations and our attempts for these two problems.

1.1 Generative Modeling for Joint View-Synthesis and Recognition

Given a never-before-seen object (*e.g.*, a gadwall in Figure 1.1), humans are able to generalize even from a single image of this object in different ways, including recognizing new object instances and imagining what the object would look like from different viewpoints. Achieving similar levels of generalization for machines is a fundamental problem in computer vision, and has been actively explored in areas such

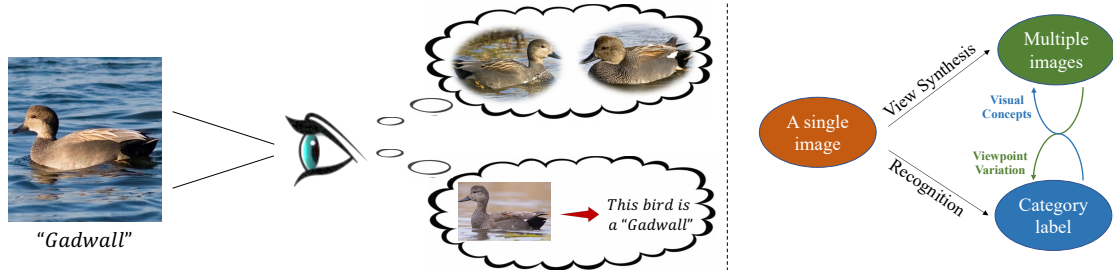


Figure 1.1: **Left:** Given a single image of a novel visual concept (*e.g.*, a gadwall), a person can generalize in various ways, including imagining what this gadwall would look like from different viewpoints (top) and recognizing new gadwall instances (bottom). **Right:** Inspired by this, we introduce a general feedback-based bowtie network that facilitates the interaction and cooperation between a generative module and a discriminative module, thus simultaneously addressing few-shot recognition and novel-view synthesis in the low-data regime.

as few-shot object recognition [24, 25, 80, 89, 90] and novel-view synthesis [54, 60, 79]. However, such exploration is often limited in *separate* areas with specialized algorithms *but not jointly*.

We argue that synthesizing images and recognizing them are inherently interconnected with each other. Being able to *simultaneously* address both tasks with a *single* model is a crucial step toward human-level generalization. This requires learning a richer, shareable internal representation for more comprehensive object understanding than it could be within individual tasks. Such “cross-task” knowledge becomes particularly critical in the low-data regime, where identifying 3D geometric structures of input images facilitates recognizing their semantic categories, and vice versa.

Inspired by this insight, here we propose a novel task of *joint few-shot recognition and novel-view synthesis*: given only one or few images of a novel object *from arbitrary views with only category annotation*, we aim to simultaneously learn an object classifier and generate images of that type of object from new viewpoints. This joint task is challenging, because of its (i) *weak supervision*, where we do not have access to any 3D supervision, and (ii) *few-shot setting*, where we need to effectively learn both 3D geometric and semantic representations from minimal data.

While existing work copes with two or more tasks mainly by multi-task learning or meta-learning of a shared feature representation [38, 104, 106], we take a different

perspective in this paper. Motivated by the nature of our problem, we focus on the *interaction and cooperation between a generative model (for view synthesis) and a discriminative model (for recognition)*, in a way that facilitates knowledge to flow across tasks *in complementary directions*, thus making the tasks help each other. For example, the synthesized images produced by the generative model provide viewpoint variations and could be used as additional training data to build a better recognition model; meanwhile, the recognition model ensures the preservation of the desired category information and deals with partial occlusions during the synthesis.

To this end, we propose a *feedback-based bowtie network (FBNet)*, as illustrated in Figure 1.1. The network consists of a view synthesis module and a recognition module, which are linked through feedback connections in a bowtie fashion. This is a general architecture that can be used on top of any view synthesis model and any recognition model. The view synthesis module explicitly learns a 3D geometric representation from 2D images, which is transformed to target viewpoints, projected to 2D features, and rendered to generate images. The recognition module then leverages these synthesized images from different views together with the original real images to learn a semantic feature representation and produce corresponding classifiers, leading to *the feedback from the output of the view synthesis module to the input of the recognition module*. The semantic features of real images extracted from the recognition module are further fed into the view synthesis module as conditional inputs, leading to *the feedback from the output of the recognition module to the input of the view synthesis module*.

One potential difficulty, when combining the view synthesis and the recognition modules, lies in the mismatch in their level of image resolutions. Deep recognition models can benefit from high-resolution images, and the recognition performance greatly improves with increased resolution [7, 31, 92]. By contrast, it is still challenging for modern generative models to synthesize very high-resolution images [52, 69]. To address this challenge, while operating on a resolution consistent with state-of-the-art view synthesis models [52], we further introduce *resolution distillation* to leverage additional knowledge in a recognition model that is learned from higher-resolution images.

We further evaluate our method in several standard datasets. The proposed FBNet significantly improves both view synthesis and recognition performance, *especially in*

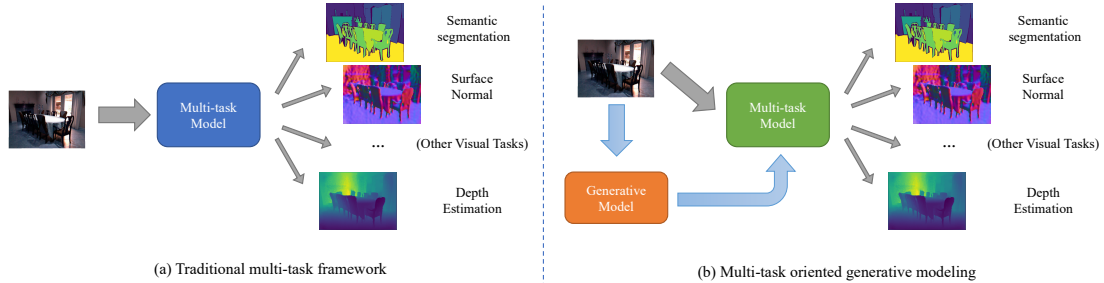


Figure 1.2: **Left:** Traditional multi-task learning framework (that learns shared feature representations) **vs. Right:** our proposed multi-task oriented generative modeling (that learns a shared generative model across various visual perception tasks)

the *low-data regime*, by enabling direct manipulation of view, shape, appearance, and semantics in generative image modeling.

1.2 Generative Modeling for Multi-Task Visual Learning

Human beings can generalize from a single image in different ways. Inspired by such ability, there has been increasing interest in building generative models that can synthesize images [28]. Yet, most of the effort has focused on generating visually realistic images [6, 110], which are still far from useful for machine perception tasks [5, 77, 96]. Even though recent work has started improving the “usefulness” of synthesized images, this line of investigation is often limited to a single specific task [54, 79, 81, 113]. Could we guide generative models to benefit *multiple* visual tasks?

While similar spirits of shared feature representations have been widely studied as multi-task learning or meta-learning [25, 106], here we are taking a different perspective — *learning a shared generative model across various tasks* (as illustrated in Figure 1.2). Leveraging multiple tasks allows us to capture the underlying image generation mechanism for more comprehensive object and scene understanding than being done within individual tasks. Taking simultaneous semantic segmentation, depth estimation, and surface normal prediction as an example (Figure 1.1), successful

generative modeling requires understanding not only the semantics but also the 3D geometric structure and physical property of the input image. Meanwhile, a learned common generative model facilitates the flow of knowledge across tasks, so that they benefit one another. For instance, the synthesized images provide meaningful variations in existing images and could be used as additional training data to build better task-specific models.

We thus explore *multi-task oriented generative modeling* (MGM), by coupling a discriminative multi-task network with a generative network. To make them cooperate with each other, a straightforward solution would be to synthesize both RGB images and corresponding *pixel-level annotations* (e.g., pixel-wise class labels for semantic segmentation and depth map for depth estimation). In the single task scenario, existing work trains a separate generative model to synthesize paired pixel-level labeled data [13, 73] and produce an augmented set. However, the quality and distribution of the generated annotations are not guaranteed. Moreover, these models are still highly task-dependant, and extending them to multi-task scenarios becomes difficult. A natural question then is: Do we actually need to synthesize paired image and multi-annotation data to be useful for multi-task visual learning?

Our MGM addresses this question by proposing a *general* framework that uses synthesized images paired with *only weak annotations* (i.e., image-level scene labels) to facilitate multiple visual tasks. Our key insight is to introduce *auxiliary discriminative tasks* that (i) only require image-level annotation or no annotation, and (ii) correlate with the original multiple tasks of interest. To this end, as additional components of the discriminative multi-task network, we introduce a *refinement* network and a *self-supervision* network that satisfies these properties. Through joint training, the discriminative network *explicitly* guides the image synthesis process. The generative network also contributes to further refining the shared feature representation. Meanwhile, the synthesized images of the generative network are used as additional training data for the discriminative network.

In more detail, the refinement network performs scene classification on the basis of the multi-task network predictions, which requires only scene labels for images. The self-supervision network can be operationalized on both real and synthesized images without reliance on annotations. With these two modules, our MGM is able to learn from both (pixel-wise) fully-annotated real images and synthesized

CHAPTER 1. INTRODUCTION

(image-level) weakly labeled images. We instantiate MGM with the state-of-the-art encoder-decoder based multi-task network [106], self-attention GAN [110], and contrastive learning-based self-supervision network [8]. Note that our framework is *agnostic to the choice of these model components*.

We evaluate our approach on standard multi-task benchmarks, including the NYUv2 [51] and Taskonomy [106] datasets. Consistent with the previous work [82, 85], we focus on three tasks of great practical importance: semantic segmentation, depth estimation, and normal prediction. The evaluation shows that our MGM consistently outperforms state-of-the-art multi-task approaches by large margins, almost reaching the *performance upper-bound* that trains with weakly annotated *real* images. Finally, we show the scalability of our approach to more visual tasks.

Chapter 2

Background

2.1 Few-Shot Recognition

Few-shot recognition is a classic problem in computer vision [24, 86]. Many algorithms have been proposed to address this problem [25, 80, 89, 90], including the recent efforts on leveraging generative models [12, 41, 42, 75, 83, 87, 91, 111, 112]. A hallucinator is introduced to generate additional examples in a pre-trained feature space as data augmentation to help with low-shot classification [91]. MetaGAN improves few-shot recognition by producing fake images as a new category [112]. However, these methods either do not synthesize images directly or use a pre-trained generative model that is not optimized towards the downstream task. By contrast, our approach performs joint training of recognition and view synthesis, and enables the two tasks to cooperate through feedback connections. In addition, while there has been work considering both classification and exemplar generation in the few-shot regime, such investigation focuses on simple domains like handwritten characters [38] but we address more realistic scenarios with natural images. Note that *our effort is largely orthogonal to designing the best few-shot recognition or novel-view synthesis method*; instead, we show that the joint model outperforms the original methods addressing each task in isolation.

2.2 Novel-View Synthesis

Novel-view synthesis aims to generate a target image with an arbitrary camera pose from one given source image [88]. It is also known as “multiview synthesis.” For this task, some approaches are able to synthesize lifelike images [35, 54, 60, 79, 94, 95, 102, 103]. However, they heavily rely on pose supervision or 3D annotation, which is not applicable in our case. An alternative way is to learn a view synthesis model in an unsupervised manner. Pix2Shape learns an implicit 3D scene representation by generating a 2.5D surfel based reconstruction [68]. HoloGAN proposes an unsupervised approach to learn 3D feature representations and render 2D images accordingly [52]. Nguyen-Phuoc et al. [53] learn scene representations from 2D unlabeled images through foreground-background fragmenting. Different from them, not only can our view synthesis module learn from weakly labeled images, but it also enables conditional synthesis to facilitate recognition.

2.3 Multi-Task Learning and Task Relationship

Multi-task learning (MTL) aims to leverage information coming from signals of related tasks so that each individual task can gain benefit [18]. A lot of work have been proposed to tackle this problem [36, 49, 58, 71, 98]. [71] identifies that most recent works use two clusters of strategies for MTL: hard parameter sharing techniques [18, 37, 65] and soft parameter sharing techniques [11, 49, 76]. These strategies have achieved good performance for MTL with similar tasks. Researchers have also carefully studied the task relationships among different tasks to make the best cooperations among them.

Task relationships have also been studied [82, 106]. *Taskonomy* exploits the relationships among various visual tasks to benefit the transfer or multi-task learning [106]. [58] proposes a meta-learning algorithm to adapt existing models to zero-shot learning tasks. [82] considers task cooperation and competition, and proposes a method to assign tasks to a few neural networks to balance all of them. Some other following works also explores task relationships among different types of tasks [2, 85, 107].

Some recent work investigates the connection between recognition and view synthesis, and makes some attempt to combine them together [48, 74, 84, 91, 97, 99].

For example, Xiong et al. [99] use multiview images to tackle fine-grained recognition tasks. However, their method needs strong pose supervision to train the view synthesis model, while we do not. Also, these approaches do not treat the two tasks of equal importance, *i.e.*, one task as an auxiliary task to facilitate the other. On the contrary, our approach targets the joint learning of the two tasks and improves both of their performance. *Importantly*, we focus on learning a shared generative model, rather than a shared feature representation as is normally the case in multi-task learning.

2.4 Generative Modeling for Visual Learning

Besides the initial goal of synthesizing realistic images, some recent work has explored the potential to leverage generative models to synthesize “useful” images for other visual tasks [78]. The most straightforward way is to generate images and the corresponding annotations as data augmentation for the target visual task [3, 13, 73]. Besides, [91] proposes to generate imaginary latent features rather than images to better benefit the low-shot classification. Another strategy to leverage generative models is through well-designed error feedback or adversarial training [14, 46, 50]. There have been works that apply generative models for different visual tasks including classification [27, 108, 113], semantic segmentation [46, 81] and depth estimation [1, 66]. These methods are limited to a single specific task and have relatively low generalizability for more tasks. In comparison, MGM is applicable to various multiple visual tasks and different generative networks.

2.5 Feedback-Based Architectures and Task Model Learning

Feedback occurs where the full or partial output of a system is routed back into the input as part of an iterative cause-and-effect process [26], have been recently introduced into neural networks [4, 101, 105]. Compared with prior work, our FBNet contains two complete sub-networks, and the output of *each* module is fed into the other as one of the inputs. Therefore, FBNet is essentially a *bi-directional* feedback-based framework which optimizes the two sub-networks jointly.

Joint data augmentation and task model learning leverage generative networks to improve other visual tasks [33, 47, 64, 109]. A generative network and a discriminative pose estimation network are trained jointly through adversarial loss in Peng et al. [64], where the generative network performs data augmentation to facilitate the downstream pose estimation task. Luo et al. [47] design a controllable data augmentation method for robust text recognition, which is achieved by tracking and refining the moving state of the control points. Zhang et al. [109] study and make use of the relationship among facial expression recognition, face alignment, and face synthesis to improve training. Mustikovela et al. [50] leverage a generative model to boost viewpoint estimation. The main difference from these work and our FBNet is that we focus on the joint task of synthesis and recognition and achieve bi-directional feedback, while existing work only considers optimizing the target discriminative task using adversarial training or with a feedforward network.

2.6 Reduced-Supervision Methods

Recent works take advantage of weakly labeled data by assigning some self-created labels (*e.g.* colorization, rotation, reconstruction) [8, 19, 56, 57, 61]. Similar self-supervised techniques have been proved useful for multi-task learning [18, 40, 44, 70]. Among these techniques, a famous one is the *Expectation-Maximization (EM)* algorithm [16], which leverages the information of weakly or unlabelled data by iteratively estimating and refining their labels. [59] further applies *EM* algorithm for semi-supervised semantic segmentation. We adopt a similar spirit and introduce the refinement network for MGM framework.

Chapter 3

Feedback-Based Network

In this Chapter, we mainly describe the problem setting of the joint recognition and view-synthesis and our approach, FBNet, for this problem. In Chapter 4, we will further report and discuss our evaluation for this network.

3.1 Our Approach

3.1.1 Joint Task of Few-Shot Recognition and Novel-View Synthesis

Problem Formulation: Given a dataset $\mathcal{D} = \{(x_i, y_i)\}$, where $x_i \in \mathcal{X}$ is an image of an object and $y_i \in \mathcal{C}$ is the corresponding category label (\mathcal{X} and \mathcal{C} are the image space and label space, respectively), we address the following two tasks *simultaneously*. (i) Object recognition: learning a discriminative model $R : \mathcal{X} \rightarrow \mathcal{C}$ that takes as input an image x_i and predicts its category label. (ii) Novel-view synthesis: learning a generative model $G : \mathcal{X} \times \Theta \rightarrow \mathcal{X}$ that, given an image x_i of category y_i and an arbitrary 3D viewpoint $\theta_j \in \Theta$, synthesizes an image in category y_i viewed from θ_j . Notice that we are more interested in *category-level consistency*, for which G is able to generate images of not only the instance x_i but also other objects of the category y_i from different viewpoints. This joint-task scenario requires us to improve the performance of both 2D and 3D tasks under weak supervision *without any ground-truth 3D annotations*. Hence, we need to exploit the *cooperation* between

them.

Few-Shot Setting: The few-shot dataset consists of one or only a few images per category, which makes our problem even more challenging. To this end, following the recent work on knowledge transfer and few-shot learning [9, 30], we leverage a set of “base” classes $\mathcal{C}_{\text{base}}$ with a large-sample dataset $\mathcal{D}_{\text{base}} = \{(x_i, y_i), y_i \in \mathcal{C}_{\text{base}}\}$ to train our initial model. We then fine-tune the pre-trained model on our target “novel” classes $\mathcal{C}_{\text{novel}}$ ($\mathcal{C}_{\text{base}} \cap \mathcal{C}_{\text{novel}} = 0$) with its small-sample dataset $\mathcal{D}_{\text{novel}} = \{(x_i, y_i), y_i \in \mathcal{C}_{\text{novel}}\}$ (e.g., a K -shot setting corresponds to K images per class).

3.1.2 Feedback-Based Bowtie Networks

To address the joint task, we are interested in learning a generative model that can synthesize realistic images of different viewpoints, which are also useful for building a strong recognition model. We propose a feedback-based bowtie network (FBNet) for this purpose. This model consists of a view synthesis module and a recognition module, trained in a joint, end-to-end fashion. Our key insight is to explicitly introduce feedback connections between the two modules, so that they cooperate with each other, thus enabling the entire model to simultaneously learn 3D geometric and semantic representations. This general architecture can be used on top of any view synthesis model and any recognition model. Here we focus on a state-of-the-art view synthesis model – HoloGAN [52], and a widely adopted few-shot recognition model – prototypical network [80], as shown in Figure 3.1.

View Synthesis Module

The view synthesis module V is shown in the blue shaded region in Figure 3.1. It is adapted from HoloGAN [52], a state-of-the-art model for unsupervised view synthesis. This module consists of a generator G which first generates a 3D feature representation from a latent constant tensor (initial cube) through 3D convolutions. The feature representation is then transformed to a certain pose and projected to 2D with a projector. The final color image is then computed through 2D convolutions. This module takes two inputs: a latent vector input z and a view input θ . z characterizes the style of the generated image through adaptive instance normalization (AdaIN) [34] units. $\theta = [\theta^x, \theta^y, \theta^z]$ guides the transformation of the 3D feature representation. This

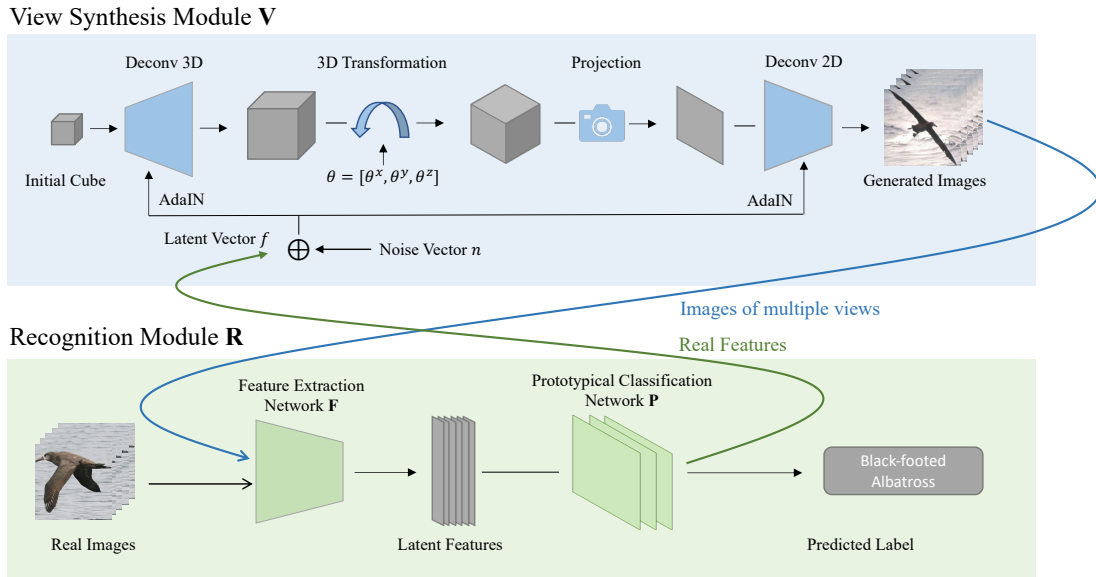


Figure 3.1: Architecture of our feedback-based bowtie network. The whole network consists of a view synthesis module and a recognition module, which are linked through feedback connections in a bowtie fashion.

module also contains a discriminator D to detect whether an image is real or fake (not shown in Figure 3.1). We use the standard GAN loss from DC-GAN [67], $\mathcal{L}_{\text{GAN}}(G, D)$. We make the following important modifications to make the architecture applicable to our joint task.

Latent Vector Formulation: To allow the synthesis module to get feedback from the recognition module (details are shown in Chapter 3.1.2), we first change HoloGAN from unconditional to conditional. To this end, we model the latent input z as: $z_i = f_i \oplus n_i$, where f_i is the conditional feature input derived from image x_i and n_i is a noise vector sampled from Gaussian distribution. \oplus is the combination strategy (*e.g.*, concatenation). By doing so, the synthesis module leverages additional semantic information, and thus maintains the category-level consistency with a target image and improves the diversity of the generated images.

Identity Regularizer: Inspired by Chen et al. [10], we introduce an identity regularizer to ensure that the synthesis module simultaneously satisfies two critical properties: (i) the identity of the generated image remains when we only change the view input θ ; (ii) the orientation of the generated image preserves when we only change

the latent input z , and this orientation should be consistent with the view input θ . Specifically, we leverage an encoding network H to predict the reconstructed latent vector z' and the view input θ' : $H(G(z, \theta)) = [z', \theta']$, where $G(z, \theta)$ is the generated image. Then we minimize the difference between the real and the reconstructed inputs as

$$\mathcal{L}_{\text{identity}}(G, H) = \mathbb{E}_z \|z - z'\|^2 + \mathbb{E}_\theta \|\theta - \theta'\|^2. \quad (3.1)$$

Here H shares the majority of the convolution layers of the discriminator D , but uses an additional fully-connected layer.

Recognition Module

The recognition module R (green shaded region in Fig. 3.1) consists of a feature extraction network F which transforms images to latent features, and a prototypical classification network P [80] which performs the final classification. Below we explain the design of these two components, focusing on how to address the technical challenges faced by joint training with view synthesis.

Feature Extraction with Resolution Distillation: We use a ResNet [31] as our feature extraction network F to transform images into latent features for the recognition module. One of the main obstacles to combining F with the synthesis module is that state-of-the-art synthesis models and recognition models operate on different resolutions. Concretely, to the best of our knowledge, current approaches to unsupervised novel-view synthesis still cannot generate satisfactory high-resolution images (*e.g.*, 224×224) [52]. By contrast, the performance of current well-performing recognition models substantially degrades with low-resolution images [7, 92]. To reconcile the resolution incompatibility, we introduce a simple distillation technique inspired by the general concept of knowledge distillation [32]. Specifically, we operate on the resolution of the synthesis module (*e.g.*, 64×64). But we benefit from an additional auxiliary feature extraction network F_{highR} that is trained on high-resolution images (*e.g.*, 224×224). We first pre-train F_{highR} following the standard practice with a cross-entropy softmax classifier [45]. We then train our feature extraction network F_{lowR} (the one used in the recognition module), under the guidance of F_{highR}

through matching their features:

$$\mathcal{L}_{\text{feature}}(F_{\text{lowR}}) = \mathbb{E}_x \|F_{\text{highR}}(x) - F_{\text{lowR}}(x)\|^2, \quad (3.2)$$

where x is a training image. With the help of resolution distillation, the feature extraction network re-captures information in high-resolution images but potentially missed in low-resolution images.

Prototypical Classification Network: We use the prototypical network P [80] as our classifier. The network assigns class probabilities \hat{p} based on distance of the input feature vector from class centers μ ; and μ is calculated by using support images in the latent feature space:

$$\hat{p}_c(x) = \frac{e^{-d(P(F_{\text{lowR}}(x)), \mu_c)}}{\sum_j e^{-d(P(F_{\text{lowR}}(x)), \mu_j)}}, \quad \mu_c = \frac{\sum_{(x_i, y_i) \in S} P(F_{\text{lowR}}(x_i)) \mathbf{I}[y_i = c]}{\sum_{(x_i, y_i) \in S} \mathbf{I}[y_i = c]}, \quad (3.3)$$

where x is a real query image, \hat{p}_c is the probability of category c , and d is a distance metric (*e.g.*, Euclidean distance). S is the support dataset. P operates on top of the feature extraction network F , and consists of 3 fully-connected layers as additional feature embedding (the classifier is non-parametric). Another benefit of using the prototypical network lies in that it enables the recognition module to explicitly leverage the generated images in a way of data augmentation, *i.e.*, S contains both real and generated images to compute the class mean. Notice that, though, the module parameters are updated based on the loss calculated on the *real query images*, which is a cross-entropy loss $\mathcal{L}_{\text{rec}}(R)$ between their predictions \hat{p} and ground-truth labels.

Feedback-Based Bowtie Model

As shown in Figure 3.1, we leverage a bowtie architecture for our full model, where the output of each module is fed into the other module as one of its inputs. Through joint training, such connections work as explicit feedback to facilitate the communication and cooperation between different modules.

Feedback Connections: We introduce two complementary feedback connections between the view synthesis module and the recognition module: (1) **recognition output** \rightarrow **synthesis input** (green arrow in Figure 3.1), where the features of the

real images extracted from the recognition module are fed into the synthesis module as conditional inputs to generate images from different views; (2) **synthesis output** \rightarrow **recognition input** (blue arrow in Figure 3.1), where the generated images are used to produce an augmented set to train the recognition module.

Categorical Loss for Feedback: The view synthesis module needs to capture the categorical semantics in order to further encourage the generated images to benefit the recognition. Therefore, we introduce a categorical loss to update the synthesis module with the prediction results of the generated images:

$$\mathcal{L}_{\text{cat}}(G) = \mathbb{E}_{y_i} \| -\log(R(G(z_i, \theta_i))) \|, \quad (3.4)$$

where y_i is the category label for the generated image $G(z_i, \theta_i)$. This loss also implicitly increases the diversity and quality of the generated images.

Final Loss Function: The final loss function is:

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{GAN}} + \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{feature}} + \lambda_{\text{id}} \mathcal{L}_{\text{identity}} + \lambda_{\text{cat}} \mathcal{L}_{\text{cat}}, \quad (3.5)$$

where λ_{id} and λ_{cat} are trade-off hyper-parameters.

Training Procedure: We first pre-train F_{highR} on the high-resolution dataset and save the computed features. These features are used to help train the feature extraction network F_{lowR} through $\mathcal{L}_{\text{feature}}$. Then the entire model is first trained on $\mathcal{C}_{\text{base}}$ and then fine-tuned on $\mathcal{C}_{\text{novel}}$. The training on the two sets are similar. During each iteration, we randomly sample some images per class as a support set and one image per class as a query set. The images in the support set, together with their computed features via the entire recognition module, are fed into the view synthesis module to generate multiple images from different viewpoints. These synthesized images are used to augment the original support set to compute the prototypes. Then, the query images are used to update the parameters of the recognition module through \mathcal{L}_{rec} ; the view-synthesis module is updated through \mathcal{L}_{GAN} , $\mathcal{L}_{\text{identity}}$, and \mathcal{L}_{cat} . The entire model is trained in an end-to-end fashion. Algorithm 1 shows the whole training process.

Algorithm 1 Training process of FBNet on base classes.

```

1: procedure TRAINING ON A MINIBATCH
2:   Initialization:
3:   max_it: Maximum iteration for the training
4:   R: Recognition module
5:   V: View synthesis module
6:   F: Feature extraction network
7:   Fhigh: Feature extraction network with
8:   high-resolution images
9:   G: Generator of view synthesis module
10:  D: Discriminator of view synthesis module
11:  n: Number of support images per class, n = 5
12:  for iter ← 1 to max_iter do
13:    Ssupport = {}, Squery = {}, Saugmented = {}
14:    for c ∈ Cbase do
15:      support_ims ← sample n images in c
16:      query_ims ← sample 1 image in c
17:      Ssupport ← Ssupport ∪ support_ims
18:      Squery ← Squery ∪ query_ims
19:    end for
20:    fhigh ← Fhigh(Ssupport ∪ Squery)
21:    flow ← F(Ssupport ∪ Squery)
22:    for img in Ssupport do
23:      f = R(img)
24:      z = f ⊕ N
25:      θ ← sample a view angle
26:      img' ← G(z, θ)
27:      y = D(img)
28:      [y', z', θ'] = D(img')
29:      LGAN(y, img, y', img') → update G, D
30:      Lid(z, θ, z', θ') → update D
31:      Saugmented ← Saugmented ∪ img'
32:    end for
33:    Swhole ← Ssupport ∪ Saugmented
34:    Lrec(Swhole, Squery) → update R
35:    Lfeature(fhigh, freal) → update F
36:    Lcat(Ssupport, Saugmented) → update G
37:  end for
38: end procedure

```

Chapter 4

Experimental Verification for FBNet

4.1 Experimental Setting

Datasets: We focus on two datasets here: the Caltech-UCSD Birds (CUB) dataset which contains 200 classes with 11,788 images [93], and the CompCars dataset which contains 360 classes with 25,519 images [100]. These are challenging fine-grained recognition datasets for our joint task. The images are resized to 64×64 . We randomly split the entire dataset into 75% as the training set and 25% as the test set. For CUB, 150 classes are selected as base classes and 50 as novel classes. For CompCars, 240 classes are selected as base classes and 120 as novel classes. Note that we focus on simultaneous recognition and synthesis over *all* base or novel classes, which is significantly more challenging than typical 5-way classification over sampled classes in most of few-shot classification work [9, 80].

Implementation Details: We set $\lambda_{\text{id}} = 10$ and $\lambda_{\text{cat}} = 1$ via cross-validation. We use ResNet-18 [31] as the feature extraction network, unless otherwise specified. To match the resolution of our data, we change the kernel size of the first convolution layer of ResNet from 7 to 5. The training process requires hundreds of examples at each iteration, which may not fit in the memory of our device. Hence, inspired by Wang et al. [91], we make a trade-off to first train the feature extraction network

through resolution distillation. We then freeze its parameters and train the other parts of our model.

Compared Methods: Our feedback connections enable the two modules to cooperate through joint training. Therefore, to evaluate the effectiveness of the feedback connections, we focus on the following comparisons. (1) For the novel-view image synthesis task, we compare our approach **FBNet** with the state-of-the-art method **HoloGAN** [52]. We also consider a variant of our approach **FBNet-view**, which has the same architecture as our novel-view synthesis module, but takes the *constant* features extracted by a pre-trained ResNet-18 as latent input. FBNet-view can be also viewed as a *conditional* version of HoloGAN. (2) For the few-shot recognition task, we compare our full model **FBNet** with its two variants: **FBNet-rec** inherits the architecture of our recognition module, which is essentially a prototypical network [80]; **FBNet-aug** uses the synthesized images from *individually trained* FBNet-view as data augmentation for the recognition module. Note that, while conducting comparisons with other few-shot recognition (*e.g.*, Chen et al. [9], Finn et al. [25]) or view synthesis models (*e.g.*, Wiles et al. [94], Yoon et al. [103]) is interesting, *it is not the main focus of this paper*. We aim to validate that the feedback-based bowtie architecture outperforms the single-task models upon which it builds, rather than designing the best few-shot recognition or novel-view synthesis method. All the models are trained following the same few-shot setting described in Chapter 3.1.1.

4.2 Main Results

4.2.1 View Synthesis Facilitates Recognition

Table 4.1 presents the top-1 recognition accuracy for the base classes and the novel classes, respectively. We focus on the challenging 1, 5-shot settings, where the number of training examples per novel class K is 1 or 5. For the novel classes, we run five trials for each setting of K , and report the average accuracy and standard deviation for all the approaches. Table 4.1 shows that our FBNet *consistently* achieves the best few-shot recognition performance on the two datasets. Moreover, the significant improvement of FBNet over FBNet-aug (where the recognition model uses additional

	Model	Base	Novel- $K=1$	Novel- $K=5$
CUB	FBNet-rec	57.91	47.53 \pm 0.14	71.26 \pm 0.26
	FBNet-aug	58.03	47.20 \pm 0.19	71.51 \pm 0.33
	FBNet	59.43	48.39 \pm 0.19	72.76 \pm 0.24
CompCars	FBNet-rec	46.05	20.83 \pm 0.03	50.52 \pm 0.11
	FBNet-aug	47.41	21.59 \pm 0.05	51.07 \pm 0.14
	FBNet	49.63	23.28 \pm 0.05	53.12 \pm 0.09

Table 4.1: Top-1 (%) recognition accuracy on the CUB and CompCars datasets. For base classes: **150-way** classification on CUB and **240-way** classification on CompCars; for K -shot novel classes: **50-way** classification on CUB and **120-way** classification on CompCars. Our FBNet consistently achieves the best performance for both base and novel classes, and joint training significantly outperforms training each module individually.

data from the *conditional* view synthesis model, but they are trained separately) indicates that the feedback-based *joint training* is the key to improve the recognition performance.

4.2.2 Recognition Facilitates View Synthesis

We investigate the novel-view synthesis results under two standard metrics. The **FID** score computes the Fréchet distance between two Gaussians fitted to feature representations of the source (real) images and the target (synthesized) images [20]. The **Inception Score (IS)** uses an Inception network pre-trained on ImageNet [17] to predict the label of the generated image and calculate the entropy based on the predictions. IS seeks to capture both the quality and diversity of a collection of generated images [72]. A higher IS or a lower FID value indicates better realism of the generated images. A larger variance of IS indicates more diversity of the generated images. We generate images of random views in one-to-one correspondence with the training examples for all the models, and compute the IS and FID values based on these images. The results are reported in Table 4.2. As a reference, we also show the results of *real images* under the two metrics, which are the best results we could expect from synthesized images. Our FBNet consistently achieves the best performance under both metrics. Compared with HoloGAN, our method brings up to 18% improvement under FID and 19% under IS. Again, the significant performance

CHAPTER 4. EXPERIMENTAL VERIFICATION FOR FBNET

	Model	IS (\uparrow)			FID (\downarrow)		
		Base	Novel- $K=1$	Novel- $K=5$	Base	Novel- $K=1$	Novel- $K=5$
CUB	<i>Real Images</i>	4.55 \pm 0.30	3.53 \pm 0.22	3.53 \pm 0.22	0	0	0
	HoloGAN	3.55 \pm 0.09	2.44 \pm 0.07	2.58 \pm 0.08	79.01	106.56	94.73
	FBNet-view	3.60 \pm 0.12	2.53 \pm 0.03	2.64 \pm 0.05	75.38	107.36	103.25
	FBNet	3.69 \pm 0.17	2.79 \pm 0.06	2.83 \pm 0.12	70.86	104.04	92.97
CompCars	<i>Real Images</i>	2.96 \pm 0.12	2.80 \pm 0.13	2.80 \pm 0.13	0	0	0
	HoloGAN	1.85 \pm 0.08	1.41 \pm 0.04	1.65 \pm 0.07	51.49	93.48	83.17
	FBNet-view	2.03 \pm 0.09	1.44 \pm 0.05	1.71 \pm 0.07	49.94	92.01	83.58
	FBNet	2.33 \pm 0.14	1.89 \pm 0.07	1.91 \pm 0.10	44.70	89.39	78.38

Table 4.2: Novel-view synthesis results under the FID and IS metrics. \uparrow indicates that higher is better, and \downarrow indicates that lower is better. As a reference, FID and IS of *Real Images* represent the best results we could expect. FBNet consistently outperforms the baselines, achieving 18% improvements for FID and 19% for IS.

gap between FBNet and FBNet-view shows that the feedback-based *joint training* substantially improves the synthesis performance.

IS and FID cannot effectively evaluate whether the generated images maintain the category-level identity and capture different viewpoints. Therefore, Figure 4.1 visualizes the synthesized multiview images. Note that, in our problem setting of limited training data under weak supervision, we could not expect that the quality of the synthesized images would match those generated based on large amounts of training data, *e.g.* Brock et al. [6]. This demonstrates the general difficulty of image generation in the few-shot setting, which is worth further exploration in the community.

Notably, even in this challenging setting, our synthesized images are of significantly higher visual quality than the state-of-the-art baselines. Specifically, (1) our FBNet is able to perform *controllable* conditional generation, while HoloGAN cannot. Such conditional generation enables FBNet to better capture the shape information of different car models on CompCars, which is crucial to the recognition task. On CUB, FBNet captures both the shape and attributes well even in the extremely low-data regime (1-shot), thus generating images of higher quality and more diversity. (2) Our FBNet also better maintains the identity of the objects in different viewpoints. For both HoloGAN and FBNet-view, it is hard to tell whether they keep the identity, but FBNet synthesizes images well from all the viewpoints while maintaining the main color and shape. (3) In addition, we notice that there is just a minor improvement for

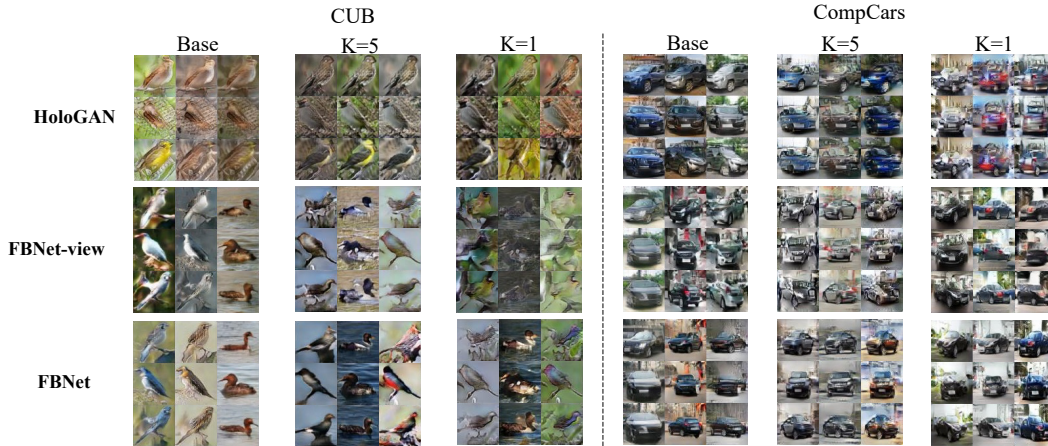


Figure 4.1: Synthesized images from multiple viewpoints. Images in the same row/column are from the same viewpoint/object. Our approach captures the shape and attributes well *even in the extremely low-data regime*.

the visual quality of the synthesis results from HoloGAN to FBNet-view, indicating that simply changing the view synthesis model from unconditional to conditional versions does not improve the performance. However, through our feedback-based joint training with recognition, the quality and diversity of the generated images significantly improve.

4.2.3 Shared Generative Model vs. Shared Feature Representation

We further compare with a standard multi-task baseline [71], which learns a shared feature representation across the joint tasks, denoted as ‘Multitask-Feat’ in Table 4.4. We treat the feature extraction network as a shared component between the recognition module and the view synthesis module, and update its parameters using both tasks *without* feedback connections. Table 4.4 shows that, through the feedback connections, our shared generative model captures the *underlying image generation mechanism* for more comprehensive object understanding, outperforming direct task-level shared feature representation.

Setting	Model	ResNet-10	ResNet-18	ResNet-34	ResNet-50
$K=1$	FBNet-view	46.28	47.53	46.79	45.68
	FBNet	48.85	48.39	47.65	47.03
$K=5$	FBNet-view	71.66	71.26	70.69	70.00
	FBNet	72.49	72.76	71.28	70.95

Table 4.3: Few-shot recognition accuracy consistently improves with different feature extraction networks.

Setting	$K=1$			$K=5$		
	Acc	FID (\downarrow)	IS (\uparrow)	Acc	FID (\downarrow)	IS (\uparrow)
Multitask-Feat	34.71	110.03	2.19 ± 0.03	52.54	99.61	2.44 ± 0.04
FBNet w/o Dist	22.47	108.73	2.31 ± 0.05	34.15	97.64	2.42 ± 0.07
FBNet w/o Proto	44.62	105.81	2.61 ± 0.07	70.04	95.15	2.76 ± 0.10
FBNet	48.39	104.04	2.79 ± 0.06	72.76	92.97	2.83 ± 0.12

Table 4.4: Ablation studies on CUB regarding (i) learning a shared feature representation through standard multi-task learning, (ii) FBNet without resolution distillation, and (iii) FBNet using a regular classification network without prototypical classification. Our full model achieves the best performance.





$\lambda_{cat} = 0$	$\lambda_{cat} = 0.1$	$\lambda_{cat} = 1$	$\lambda_{cat} = 5$
			
Top 1 Acc : 68.84 ± 0.29 FID : 103.75 IS : 2.58 ± 0.10	Top 1 Acc : 70.93 ± 0.27 FID : 99.50 IS : 2.75 ± 0.09	Top 1 Acc : 72.76 ± 0.24 FID : 92.97 IS : 2.83 ± 0.12	Top 1 Acc : 73.09 ± 0.21 FID : 114.85 IS : 2.36 ± 0.08

Figure 4.2: Ablation on λ_{cat} . Categorical loss trades off the performance between view synthesis and recognition.

4.3 Ablation Study

4.3.1 Different Recognition Networks

While we used ResNet-18 as the default feature extraction network, our approach is applicable to different recognition models. Table 4.3 shows that the recognition performance with different feature extraction networks consistently improves. Interestingly, ResNet-10/18 outperform the deeper models, indicating that the deeper models might suffer from over-fitting in few-shot regimes, consistent with the observation in [9].

4.3.2 Categorical Loss

In addition to the feedback connections, our synthesis and recognition modules are linked by the categorical loss. To analyze its effect, we vary λ_{cat} among 0 (without the categorical loss), 0.1, 1, and 5. Figure 4.2 shows the quantitative and qualitative results on CUB. With λ_{cat} increasing, the recognition performance improves gradually. Meanwhile, a too large λ_{cat} reduces the visual quality of the generated images: checkerboard noise appears. While these images are not visually appealing, they still benefit the recognition task. This shows that the categorical loss trades off the performance between the two tasks, and there is a “sweet spot” between them.

4.3.3 Resolution Distillation and Prototypical Classification

Our proposed resolution distillation reconciles the resolution inconsistency between the synthesis and recognition modules, and further benefits from a recognition model trained on high-resolution images. The prototypical network leverages the synthesized images, which constitutes one of the feedback connections. We evaluate their effect by building two variants of our model without these techniques: ‘FBNet w/o Dist’ trains the feature extraction network directly from low-resolution images; ‘FBNet w/o Proto’ uses a regular classification network instead of the prototypical network. Table 4.4 shows that the performance of full FBNet significantly outperforms these variants, verifying the importance of our techniques.

4.4 Qualitative Results on the CelebA-HQ Dataset

We further show that the visual quality of our synthesized images significantly gets improved on datasets *with better aligned poses*. For this purpose, we conduct experiments on CelebA-HQ [39], which contains 30,000 aligned human face images regarding 40 attributes in total. We randomly select 35 attributes as training attributes and 5 as few-shot test attributes. While CelebA-HQ does not provide pose annotation, the aligned faces mitigate the pose issue to some extent. Figure 4.3 shows that both the visual quality and diversity of our synthesized images substantially improve, while



Figure 4.3: Synthesized images by HoloGAN and FBNet on CelebA-HQ. Few-shot attributes (left to right): Black Hair, Gray Hair, Bald, Wearing Hat, and Aging. FBNet synthesizes images of higher quality and diversity.

consistently outperforming HoloGAN.

4.5 Discussion and Future Work

Our experimental evaluation has focused on fine-grained categories, mainly because state-of-the-art novel-view synthesis models still cannot address image generation for a wide spectrum of general images [43]. Meanwhile, our feedback-based bowtie architecture is general. With the advance in novel-view synthesis, such as the recent work of BlockGAN [53] and RGBD-GAN [55], our framework could be potentially extended to deal with broader types of images. Additional further investigation includes exploring more architecture choices and dealing with images with more than one object.

Chapter 5

Multi-Task Oriented Generative Modeling

We propose multi-task oriented generative modeling (MGM) to leverage generative networks for multi-task visual learning, as summarized in Figure 5.1. In this section, we first formalize the novel problem setting of MGM. Then, we explain the general framework and an instantiation of the MGM model with state-of-the-art multi-task learning and image generation approaches. Finally, we discuss the detailed training strategy for the framework.

5.1 Problem Setting

Multi-task discriminative learning: Given n visual tasks $\mathcal{T} = \{T_1, T_2, \dots, T_n\}$, we aim to learn a discriminative multi-task model \mathbf{M} that is able to address all of these tasks simultaneously: $\mathbf{M}(x) \rightarrow \hat{\mathbf{y}} = (\hat{y}^1, \hat{y}^2, \dots, \hat{y}^n)$, where x is an input image and \hat{y}^i is the prediction for task T_i . Here we focus on the type of per-pixel level prediction tasks (*e.g.*, semantic segmentation or depth estimation). We treat image classification as a special task, which provides global semantic description (*i.e.*, scene labels) of images and only requires image-level category annotation c . Therefore, the set of fully annotated real data is denoted as $\mathcal{S}_{\text{real}} = \{(x_j, y_j^1, y_j^2, \dots, y_j^n, c_j)\}$.

Generative learning: Meanwhile, we aim to learn a generative model \mathbf{G} that produces a set of synthesized data but with only corresponding image-level scene

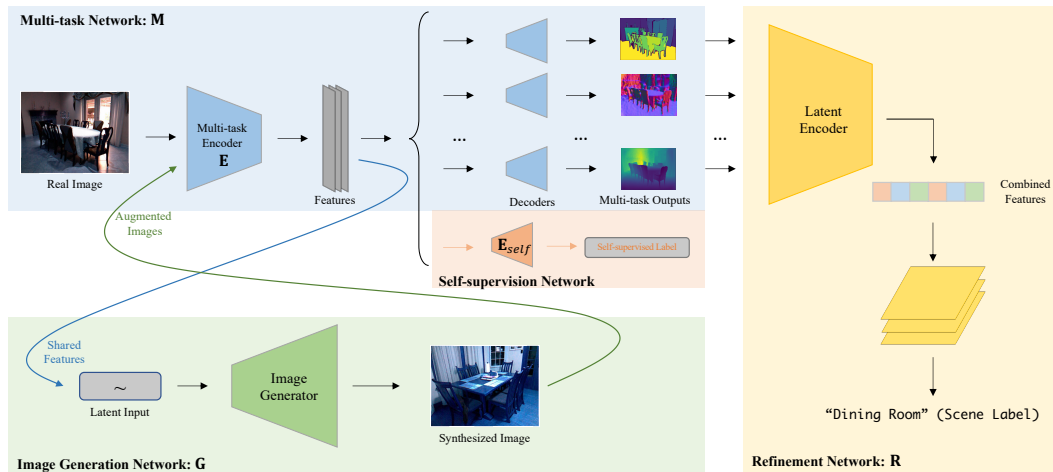


Figure 5.1: Architecture of our proposed multi-task oriented generative modeling (MGM) framework. There are four main components in the framework: Multi-task network to address the target multiple pixel-level prediction tasks; self-supervision network to facilitate representation learning using images without any annotation; refinement network to perform scene classification using weak annotation; image generation network to synthesize useful images that benefit multiple tasks.

labels (weak annotation): $\mathbf{G}(c, z) \rightarrow \tilde{x}$, where z is a random input, and \tilde{x} is a synthesized image. The scene label of \tilde{x} is denoted as $\tilde{c} = c$. We denote the set of synthesized images and their corresponding scene labels as $\tilde{\mathcal{S}}_{\text{syn}} = \{(\tilde{x}_k, \tilde{c}_k)\}$.

Cooperation between discriminative and generative learning: Our objective is that the discriminative model \mathbf{M} and the generative model \mathbf{G} cooperate with each other to improve the performance on the multiple visual tasks \mathcal{T} . During the whole process, the full model only gets access to the real fully-labeled data $\mathcal{S}_{\text{real}}$, then the generative network is trained to produce the synthesized set $\tilde{\mathcal{S}}_{\text{syn}}$. Finally, \mathbf{M} effectively learns from both $\mathcal{S}_{\text{real}}$ and $\tilde{\mathcal{S}}_{\text{syn}}$. Note that, unlike most of the existing work on image generation [6, 110], we do not focus on the visual realism of the synthesized images \tilde{x} . Instead, we hope \mathbf{G} to capture the underlying mechanism that benefits \mathbf{M} .

5.2 Framework and Architecture

Figure 5.1 shows the architecture of our proposed MGM framework. It contains four components: the main multi-task discriminative network \mathbf{M} , the image generation network \mathbf{G} , the refinement network \mathbf{R} , and the self-supervision network. By introducing the refinement network and the self-supervision network, the full model can leverage both fully-labeled real images and weakly-labeled synthesized images to facilitate the learning of latent feature representation. These two networks thus allow \mathbf{M} and \mathbf{R} to better cooperate with each other. Notice that our MGM is a *model-agnostic* framework, and here we instantiate its components with state-of-the-art models. In the ablation study (Sec. 6.3), we show that our MGM works well with different choices of the model components.

5.2.1 Multi-task Network (\mathbf{M})

The multi-task network aims to make predictions for multiple target tasks based on an input image. Consistent with the most recent work on multi-task learning, we instantiate an encoder-decoder based architecture [85, 106, 110]. Considering the trade-off between model complexity and performance, we use a shared encoder \mathbf{E} to extract features from input images, and individual decoders for each target task. We adopt a ResNet-18 [31] for the encoder and symmetric transposed decoders following [106]. For each task, we have its own loss function to update the corresponding decoder and the shared encoder.

5.2.2 Image Generation Network (\mathbf{G})

The generative model \mathbf{G} is a variant of generative adversarial networks (GANs). We include the generator in our framework, but this module also has a discriminator during its own training. \mathbf{G} takes as input a latent vector z and a category label c , and synthesizes an image belonging to category c . Considering the trade-off between performance and training cost, we instantiate \mathbf{G} with self-attention generative adversarial network (SAGAN) [110]. We achieve conditional image generation by

applying conditional batch normalization (CBN) layers [15]:

$$\text{CBN}(f_{i,c,h,w} \mid \gamma_c, \beta_c) = \gamma_c \frac{f_{i,c,h,w} - \mathbb{E}[f_{\cdot,c,\cdot,\cdot}]}{\sqrt{\text{Var}[f_{\cdot,c,\cdot,\cdot}] + \epsilon}} + \beta_c, \quad (5.1)$$

where $f_{i,c,h,w}$ is an extracted c -channel 2D feature for the i -th sample, and ϵ is a small value to avoid collapse. γ_c and β_c are two parameters to control the mean and variance of the normalization, which are learned by the model for each class. We use hinge loss for the adversarial training. Notice that the proposed framework is flexible with different generative models, and we instantiate with a state-of-the-art module.

5.2.3 Refinement Network (**R**)

As one of our key contributions, we introduce the refinement network **R** to further refine the shared representation using the global scene category labels. **R** takes the predictions of the multi-task network as input and predicts the category label of the input image. Importantly, because **R** only requires category labels, it can be effortlessly operationalized on the “weakly-annotated” synthesized images. Meanwhile, **R** also enforces the semantic consistency of the synthesized images with **G**.

We apply an algorithm inspired by Expectation-Maximum (EM) [16] to train the refinement network **R**. For the fully-annotated real images (x, \mathbf{y}, c) , we use the scene classification loss to update **R** and refine the encoder **E** in the multi-task network **M**. Then for the synthesized images (\tilde{x}, \tilde{c}) , since their multi-task predictions produced by **M** might not be reliable, we only refine **E** with **R** frozen using the scene classification loss. Through refining the share feature representation with the synthesized images, this process also provides implicit guidance to the image generation network.

More specifically, we model the whole multi-task network and refinement network as a joint probability graph:

$$P(x, \mathbf{y}, c; \theta, \theta') = P(x) \left(\prod_{i=1}^n P(y^i \mid x; \theta) \right) P(c \mid \mathbf{y}; \theta'), \quad (5.2)$$

where x is an input image, \mathbf{y} is the vector of multi-task predictions, c is the scene label, θ is the vector of parameters of the multi-task network, and θ' is the vector of parameters of the refinement network. The parameters θ and θ' are learned to

maximize the joint probability. For data samples in $\mathcal{S}_{\text{real}}$, we maximize the joint probability and update θ' to train the refinement network.

$$\theta'^{\star} = \operatorname{argmax}_{\theta'} P(\tilde{c}_k | \mathbf{y}; \theta'). \quad (5.3)$$

For data samples in $\tilde{\mathcal{S}}_{\text{syn}}$, we update the parameters of \mathbf{M} (θ) in an EM-like manner. During the **E** step, we estimate the latent multi-task ground-truth by:

$$\mathbf{y}^{\dagger} = \operatorname{argmax}_{\mathbf{y}} P(\mathbf{y} | \tilde{x}_k; \theta) P(\tilde{c}_k | \mathbf{y}; \theta'). \quad (5.4)$$

Then for the **M** step, we back-propagate the error between \mathbf{y}^{\dagger} and $\hat{\mathbf{y}}$ (the multi-task predictions) to the multi-task encoder.

$$\theta^{\star} = \operatorname{argmax}_{\theta} P(\mathbf{y}^{\dagger} | \tilde{x}_k; \theta). \quad (5.5)$$

We use cross-entropy as the classification loss function.

5.2.4 Self-supervision Network

The self-supervision network facilitates the representation learning of the encoder \mathbf{E} by performing self-supervised learning tasks on images without any annotation so that can be operationalized on both real and synthesized images. We modify SimCLR [8], one of the state-of-the-art approaches, as our self-supervision network.

This network contains an additional embedding network \mathbf{E}_{self} , working on the output of the multi-task encoder \mathbf{E} , to obtain a 1D latent feature of the input image: $\mu = \mathbf{E}_{\text{self}}(\mathbf{E}(x))$. Then, it performs contrastive learning with these latent vectors. Specifically, given a minibatch of N images, this network first randomly samples two transformed views of each source image as augmented images, resulting in $2N$ augmented images. For each augmented image, there is only one pair of positive augmented examples from the same source image, and other $2(N - 1)$ negative pairs. Then the network jointly minimizes the distance of positive pairs and maximizes the distance of negative pairs in the latent space, through the normalized temperature-

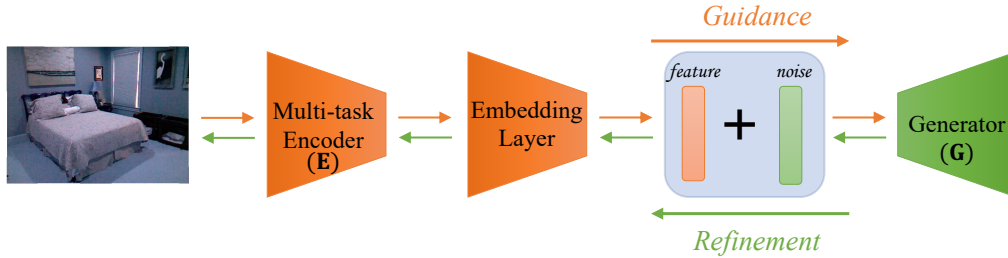


Figure 5.2: Joint training of the multi-task network and the image generation network. The multi-task network provides useful feature representation to guide the image generation process, while the generation network refines the shared representation through back-propagation.

scaled cross-entropy (NT - $Xent$) loss [8]:

$$\ell_{i,j} = -\log \frac{\exp(\text{dis}(\mu_i, \mu_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{dis}(\mu_i, \mu_k)/\tau)}, \quad (5.6)$$

where $\ell_{i,j}$ is the NT - $Xent$ loss for a positive pair of examples in the latent space (μ_i, μ_j) . $\mathbb{1}_{[k \neq i]} \in 0, 1$ is an indicator function evaluating to 1 if $k \neq i$, and τ is a temperature parameter. $\text{dis}(\mu_i, \mu_j)$ is a distance function, and we use cosine distance following [8]. This loss is further back-propagated to refine the multi-task encoder \mathbf{E} . Notice that other types of self-supervised tasks are applicable as well. To demonstrate this, in Sec. 6.3 we also report the result with another task — image reconstruction.

5.3 Interaction Among Networks

Cooperation Through Joint Training: We propose a simple but effective joint training algorithm shown in Figure 5.2. The image generation network \mathbf{G} takes the transferred feature representation of the multi-task encoder \mathbf{E} , added with some Gaussian noise, as the latent input z to conduct conditional image generation. Hence, the generation network obtains *additional, explicit guidance* (*i.e.*, extra effective features) from the multi-task network to facilitate the generation of “better images”—images that may not look more realistic but are more useful for the multiple target tasks. Then, the generation error of \mathbf{G} will be back-propagated to \mathbf{E} to further refine the shared representation. This process can be also viewed as introducing image

Algorithm 2 The training procedure of MGM.

```

1: procedure TRAINING ON A MINIBATCH
2:   Initialization:
3:   max_epoch: Maximum epoch for the training
4:   M: Multi-task Network, G: Image Generation Network
5:   E: Multi-task Encoder, R: the Refinement Network
6:   Eself: Self-supervision Network Encoder
7:   N: minibatch size
8:   for epoch  $\leftarrow 1$  to max_epoch do
9:     Split  $\mathcal{S}_{\text{real}}$  into minibatches with size N:  $\mathcal{S}_{\text{mini}}$ 
10:    for  $(x, \mathbf{y}, c) \in \mathcal{S}_{\text{mini}}$  do
11:       $\hat{\mathbf{y}} = \mathbf{M}(x)$ 
12:       $\mathcal{L}_{\text{multi}}(\mathbf{y}, \hat{\mathbf{y}}) \rightarrow$  update  $\mathbf{M}$ 
13:       $\hat{c} = \mathbf{R}(\hat{\mathbf{y}})$ 
14:       $\mathcal{L}_{CE}(c, \hat{c}) \rightarrow$  update  $\mathbf{R}, \mathbf{E}$ 
15:      Sample  $2N$  augmented images  $x_{\text{aug}}$ 
16:       $\mathcal{L}_{NT-Xent}(x_{\text{aug}}) \rightarrow$  update  $\mathbf{E}, \mathbf{E}_{\text{self}}$ 
17:      Use  $\mathcal{L}_{GAN}$  to train  $\mathbf{G}$ 
18:       $(\tilde{x}, \tilde{c}) = \mathbf{G}(x, c)$ 
19:       $\mathcal{L}_{CE}(\tilde{c}, \mathbf{R}(\mathbf{M}(\tilde{x}))) \rightarrow$  update  $\mathbf{E}$ 
20:      Sample  $2N$  augmented images for the synthesized data  $\tilde{x}_{\text{aug}}$ 
21:       $\mathcal{L}_{NT-Xent}(\tilde{x}_{\text{aug}}) \rightarrow$  update  $\mathbf{E}$ .
22:    end for
23:  end for
24: end procedure

```

generation as an additional task in the multi-task learning framework.

5.3.1 Training Procedure:

We summarized the training procedure in Algorithm 2. Here we further explain the training procedure in more details. Given a minibatch of data in $\mathcal{S}_{\text{real}}$, we conduct the following training procedure.

1. For the input images x , we predict $\hat{\mathbf{y}} = \mathbf{M}(x)$, and then use the task-specific losses between \mathbf{y} and $\hat{\mathbf{y}}$ to update the multi-task network \mathbf{M} .
2. We predict the scene labels by $\hat{c} = \mathbf{R}(\hat{\mathbf{y}})$, and update the refinement network \mathbf{R} and the multi-task encoding network \mathbf{E} using the cross-entropy loss between

c and \hat{c} .

3. We randomly sample pairs of augmented images, process them with the self-supervision network, and then update the self-supervision network and the multi-task encoder \mathbf{E} with the *NT-Xent* loss in Eqn. (6).
4. We train the image generation network \mathbf{G} through adversarial training with (x, c) , and back-propagate the adversarial error and update \mathbf{E} at the same time.
5. We sample another minibatch of synthesized data (\tilde{x}, \tilde{c}) , and use these data to update \mathbf{E} by performing both the EM-like algorithm described in Chapter 5.2.3 with \mathbf{R} and the self-supervised learning as in step 3.

Chapter 6

Experimental Verification for MGM

To evaluate our proposed MGM model and investigate the impact of each component, we conduct a variety of experiments on two standard multi-task learning datasets. We also perform detailed analysis and ablation studies here.

6.1 Datasets and Compared Methods

Datasets: Following the work of [85] and [82], we mainly focus on three representative visual tasks in the main experiments: semantic segmentation (SS), surface normal prediction (SN), and depth estimation (DE). At the end of this section, we will show that our approach is scalable to an additional number of tasks. We evaluate all the models on two widely-benchmarked datasets: **NYUv2** [23, 51] containing 1,449 images with 40 types of objects [29]; **Tiny-Taskonomy** which is the standard tiny split of the Taskonomy dataset [106]. Since a certain amount of images for each category is required to train a generative network, we keep the images of the top 35 scene categories on Tiny-Taskonomy, with each one consisting of more than 1,000 images. This resulting dataset contains 358,426 images in total. For NYUv2, we randomly select 1,049 images as the full training set and 200 images each as the validation/test set. For Tiny-Taskonomy, we randomly pick 80% of the whole set as the full training set and 10% each as the validation/test set.

Compared Methods: We mainly focus on our comparison with two state-of-the-art discriminative baselines: **Single-Task (ST)** model follows the architecture of Taskonomy single task network [106], and address each task individually; **Multi-Task (MT)** model refers to the sub-network for the three tasks of interest in [82]. These two baselines can be viewed as using our multi-task network without the proposed refinement, self-supervision, and generation networks. Note that *our work is the first that introduces generative modeling for multi-task learning, and there is no existing baseline in this direction.*

Our **MGM** is the full model trained with both fully-labeled *real* data and weakly-labeled *synthesized* data, which are produced by the generation network through joint training. In addition, to further validate the effectiveness of our **MGM** model, we consider its variant model **MGM_r** that is trained with both fully and weakly labeled *real* data. **MGM_r** is used to show *the performance upper bound* in the semi-supervised learning scenario, where the synthesized images are replaced by the real images in the dataset. The resolution is set to 128 for all the experiments. For all the compared methods, we use a ResNet-18 like architecture to build the encoder and use the standard decoder architecture of Taskonomy [106].

Data Settings: We conduct experiments with three different data settings: (1) 100% data setting; (2) 50% data setting; and (3) 25% data setting. For each setting, we use 100%, 50%, or 25% of the entire labeled training set to train the model. For **MGM_r**, we add another 50% or 25% of weakly-labeled real data in the last two settings. For **MGM**, we include the same number of weakly-labeled synthesized data in all three settings.

Evaluation Metrics: For NYUv2, following the metrics in [23, 85], we measure the mean Intersection-Over-Union (mIOU) for the semantic segmentation task, the mean Absolute Error (mABSE) for the depth estimation task, and the mean Angular Distance (mAD) for the surface normal estimation task. For Tiny-Taskonomy, we follow the evaluation metrics of previous work [82, 85, 106] and report the averaged loss values on the test set.

	Data Setting	100% Data Setting			50% Data Setting				25% Data Setting			
	Models	ST	MT	MGM	ST	MT	MGM	MGM_r	ST	MT	MGM	MGM_r
NYU v2	SS-mIOU(\uparrow)	0.249 ± 0.008	0.256 ± 0.005	0.264 ± 0.005	0.230 ± 0.009	0.237 ± 0.006	0.251 ± 0.005	<i>0.258</i> ± 0.004	0.199 ± 0.004	0.207 ± 0.007	0.229 ± 0.004	<i>0.231</i> ± 0.005
	DE-mABSE(\downarrow)	0.748 ± 0.019	0.708 ± 0.021	0.698 ± 0.014	0.837 ± 0.017	0.819 ± 0.018	0.734 ± 0.011	<i>0.723</i> ± 0.010	0.908 ± 0.017	0.874 ± 0.015	0.844 ± 0.011	<i>0.821</i> ± 0.009
	SN-mAD(\downarrow)	0.273 ± 0.06	0.283 ± 0.008	0.255 ± 0.010	0.309 ± 0.008	0.291 ± 0.010	0.273 ± 0.009	<i>0.270</i> ± 0.006	0.312 ± 0.007	0.296 ± 0.007	0.277 ± 0.006	<i>0.274</i> ± 0.005
Tiny Taskonomy	SS-mLoss(\downarrow)	0.111 ± 0.002	0.137 ± 0.003	0.106 ± 0.003	0.120 ± 0.003	0.138 ± 0.002	0.114 ± 0.003	<i>0.112</i> ± 0.002	0.119 ± 0.003	0.141 ± 0.002	0.117 ± 0.002	<i>0.115</i> ± 0.002
	DE-mLoss(\downarrow)	1.716 ± 0.006	1.584 ± 0.008	1.472 ± 0.006	1.768 ± 0.007	1.595 ± 0.009	1.499 ± 0.008	<i>1.378</i> ± 0.007	1.795 ± 0.010	1.692 ± 0.008	1.585 ± 0.009	<i>1.580</i> ± 0.008
	SN-mLoss(\downarrow)	0.155 ± 0.003	0.153 ± 0.003	0.145 ± 0.002	0.157 ± 0.002	0.156 ± 0.002	0.147 ± 0.002	<i>0.140</i> ± 0.001	0.154 ± 0.002	0.152 ± 0.002	0.148 ± 0.003	<i>0.142</i> ± 0.002

Table 6.1: Main results (mean \pm std) on the NYUv2 and Tiny-Taskonomy datasets. SS: semantic segmentation; DE: depth estimation; SN: surface normal prediction. \uparrow means higher is better; \downarrow means lower is better. We use different metrics on the two datasets, following existing protocol. Our MGM consistently and significantly outperforms both single-task (ST) and multi-task (MT) baselines, *even reaching the performance upper-bound of training with weakly annotated real images* (MGM_r).

6.2 Main Results

6.2.1 Quantitative Results

We run all the models for 5 times and report the averaged results and the standard deviation on the two datasets in Table 6.1. From this table, we have the following key observations that support the effectiveness of our approach which combines generative learning with discriminative learning. (1) Existing discriminative multi-task learning approaches may not consistently benefit all the three individual tasks. However, our MGM consistently and significantly outperforms both the single-task and multi-task baselines across all the scenarios. (2) By using weakly-labeled synthesized data, the results of our model in the 50% data setting are even better than those of baselines in the 100% data setting. (3) More interesting, the performance of our MGM is close to MGM_r , which indicates that our synthesized images are *comparably useful* as real images for improving multiple visual perception tasks. The performance gap is especially minimal in the 25% labeled data setting, suggesting that our proposed MGM model is, in particular, helpful for low-data regime.

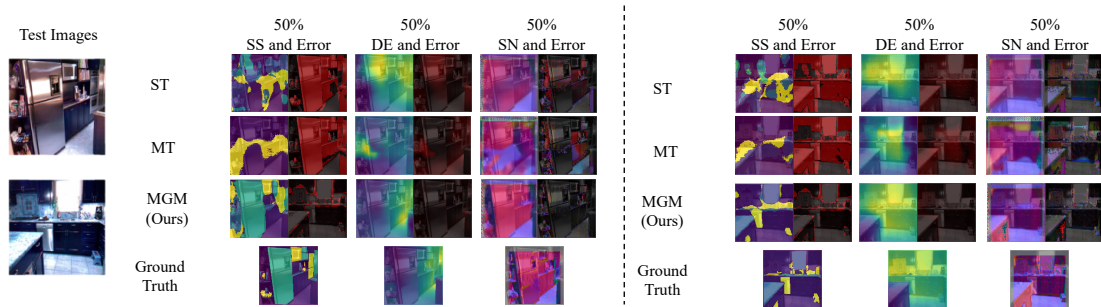


Figure 6.1: Visualization and error comparison of the multi-task prediction outputs in the 50% data setting. The prediction results of MGM is quite close to the ground-truth, significantly outperforming the state-of-the-art results.

6.2.2 Qualitative Results

We also visualize the prediction results on the three tasks for ST, MT, and MGM in the 50% data setting in Figure 6.1. While obvious defects can be found for all the baselines, the results of our proposed method are quite close to the ground-truth.

How Does Generative Modeling Benefit Multi-tasks? To have a better understanding of how the generative modeling and joint learning mechanism benefit multi-task visual learning, we also consider two variants of our MGM model and evaluate their performance. $\text{MGM}_{/G}$ is the MGM model trained with $\mathcal{S}_{\text{real}}$ only (without generative modeling), which shows the performance of our proposed multi-task learning framework in general (with the help from the auxiliary refinement and self-supervision networks), and helps to understand the gain of leveraging generative modeling. $\text{MGM}_{/j}$ is trained with synthesized images produced by a pre-trained SAGAN *without* the joint training mechanism. Table 6.2 shows the results on the two datasets.

Combining the results of Tables 6.2 and 6.1, we find: (1) MGM outperforms both ST and MT baseline even without generative modeling, indicating the benefit of the self-supervised task and the refinement network. (2) By introducing synthesized images that are trained separately, the multi-task performance slightly improves, which shows the effectiveness of involving generative modeling into multi-task discriminative learning, under the assistance of our refinement and self-supervision networks. (3) The joint learning mechanism further improves the cooperation between generative modeling and discriminative learning, thus enabling the generative model to better

	Data Setting	100% Data Setting			50% Data Setting			25% Data Setting		
	Models	MGM _{/G}	MGM _{/j}	MGM	MGM _{/G}	MGM _{/j}	MGM	MGM _{/G}	MGM _{/j}	MGM
NYU v2	SS-mIOU(↑)	0.261	0.262	0.264	0.243	0.243	0.251	0.215	0.220	0.229
	DE-mABSE(↓)	0.707	0.701	0.698	0.799	0.763	0.734	0.868	0.860	0.844
	SN-mAD(↓)	0.262	0.259	0.255	0.287	0.281	0.273	0.292	0.286	0.277
Tiny Taskonomy	SS-mLoss(↓)	0.108	0.108	0.106	0.116	0.115	0.114	0.119	0.121	0.117
	DE-mLoss(↓)	1.491	1.488	1.472	1.527	1.523	1.499	1.636	1.616	1.585
	SN-mLoss(↓)	0.151	0.151	0.145	0.153	0.152	0.147	0.154	0.152	0.148

Table 6.2: Comparison of our MGM model with its variants. MGM_{/G}: *without* generating synthesized images; MGM_{/j}: *without* joint learning. Our MGM outperforms single-task and multi-task baselines *even without synthesized data*, showing its effectiveness as a general multi-task learning framework. The model performance further improves with joint learning.

facilitate multi-task visual learning.

6.3 Ablation Study

For all the experiments in this section, models are trained in the 50% data setting, unless specifically mentioned.

6.3.1 Impact of Parameters

Introducing the refinement, self-supervision, and image generation networks also leads to more parameters. To validate that the performance improvements come from the novel design of our architecture rather than merely increasing the number of parameters, we provide two model variants as additional baselines: **ST**₁ and **MT**₁ use ResNet-34 as the encoder network and the corresponding decoder networks. These two networks have a similar amount of parameters as MGM. The top 4 rows in Table 6.3 show that simply increasing the number of parameters cannot significantly boost performance.

6.3.2 Impact of Self-supervision Task and Refinement Network

Two important components of the proposed framework are the self-supervision task and the refinement network. We evaluate their impact individually in Table 6.3.

Model	SS-mIOU (\uparrow)	DE-mABSE (\downarrow)	SN-mAD (\downarrow)
ST	0.230	0.837	0.309
MT	0.237	0.819	0.291
ST ₁	0.232	0.841	0.304
MT ₁	0.236	0.804	0.288
MGM _{/self}	0.239	0.776	0.279
MGM _{/refine}	0.254	0.808	0.290
MGM _{recon}	0.241	0.768	0.285
MGM	0.251	0.734	0.273

Table 6.3: Ablation study. (1) ST₁ and MT₁: baselines with a larger number of parameters (with deeper backbones); (2) MGM_{/self}: *without* self-supervision task; (3) MGM_{/refine}: *without* classification refinement network; and (4) MGM_{recon}: *with* a simple reconstruction task as self-supervision. The two proposed components are complementary and both benefit the multiple tasks. The refinement network works better for surface normal; the self-supervision network works better for semantic segmentation. Their combination achieves the best.

MGM_{/self} is the model trained *without* the self-supervision task; MGM_{/refine} is the model *without* the refinement network; for MGM_{recon}, we replace the SimCLR based self-supervision method with a weaker reconstruction task, and use Mean Square Error as the loss function.

We could see that the refinement network works better for the surface normal task, and the self-supervision task works better for the semantic segmentation task; they are complementary to each other, and combining them generally achieves the best performance. In addition, the model could still gain some benefit even when we use some weak self-supervision tasks like reconstruction, which indicates the generability and robustness of our MGM model.

6.3.3 Number of Synthesized Images vs. Real images

From the previous results, we have found that the synthesized images could benefit the target multi-tasks in a way similar to weakly labeled real images. To further investigate the impact of the number of synthesized images, we vary it from 25% to 125% during multi-task training on NYUv2 in the 25% real data setting. Figure 6.2 summarizes the result. First, we can see that the performance gap between MGM_{/j} (without joint training) and MGM becomes larger for a higher ratio of weakly labeled data, which indicates the importance of our joint learning mechanism. *More importantly,*

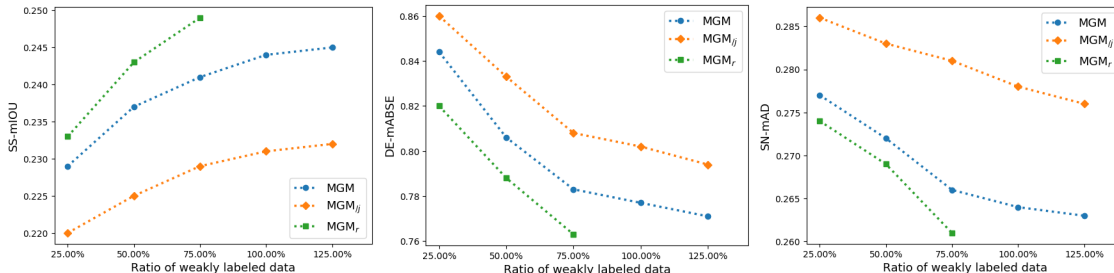


Figure 6.2: Performance change with different ratios of weakly labeled data. Joint learning significantly improves the performance. The performance of MGM keeps increasing with the number of the weakly labeled *synthesized* images, achieving results almost comparable to that of MGM_r trained with all the available weakly labeled *real* images.

Model	SS (\downarrow)	DE (\downarrow)	SN (\downarrow)	ET (\downarrow)	Re (\downarrow)	PC (\downarrow)
ST	0.120	1.768	0.157	0.228	0.703	0.462
MT	0.112	1.747	0.169	0.241	0.704	0.436
MGM	0.108	1.715	0.152	0.201	0.699	0.417

Table 6.4: Mean test losses for six tasks on Tiny-Taskonomy. Again, our MGM outperforms the baselines, indicating its flexibility, generability, and scalability.

while the real images are constrained in number due to the human collection effort, our generation network is able to synthesize *unlimited* amounts of images. This is demonstrated in the comparison between MGM_r (with real images) and MGM: the performance of our MGM keeps improving with respect to the number of synthesized images, achieving results almost comparable to that of MGM_r when MGM_r uses all the available weakly labeled real images.

6.4 Extension

Experiments with More Tasks: MGM is also flexible and scalable with different tasks. In addition to the three tasks addressed in the main experiments, here we add three extra tasks: Edge Texture (ET), Reshading (Re), and Principal Curvature (PC), leading to six tasks in total. We evaluate the performance of all the compared models on Tiny-taskonomy in the 50% data setting, and report the mean test loss for all the tasks. The result is reported in Table 6.4. Again, our proposed method still outperforms state-of-the-art baselines.

Chapter 7

Conclusions

In this thesis paper, we mainly focus on applying generative modeling to facilitate multi-task visual learning. We first propose a feedback-based bowtie network for the joint task of few-shot recognition and novel-view synthesis. This model consistently improves performance for both tasks, especially with extremely limited data. The proposed framework could be potentially extended to address more tasks, leading to a generative model useful and shareable across a wide range of tasks.

Motivated by the benefit of FBNet, we further target at introducing generative modeling for multi-task visual learning. The main challenge is that it is hard for generative models to synthesize both RGB images and pixel-level annotations in multi-task scenarios. We address this problem by proposing multi-task oriented generative modeling (MGM) framework equipped with the self-supervision network and the refinement network, which enable us to take advantage of synthesized images paired with image-level scene labels to facilitate multiple visual tasks. Experimental results indicate our MGM model consistently outperforms state-of-the-art multi-task approaches.

CHAPTER 7. CONCLUSIONS

Bibliography

- [1] Filippo Aleotti, Fabio Tosi, Matteo Poggi, and Stefano Mattoccia. Generative adversarial networks for unsupervised monocular depth prediction. In *ECCV Workshops*, 2018. [2.4](#)
- [2] Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera. In *ICCV*, 2019. [2.3](#)
- [3] Zhipeng Bao, Yu-Xiong Wang, and Martial Hebert. Bowtie networks: Generative modeling for joint few-shot recognition and novel-view synthesis. *ICLR*, 2021. [2.4](#)
- [4] Vasileios Belagiannis and Andrew Zisserman. Recurrent human pose estimation. In *International Conference on Automatic Face & Gesture Recognition (FG)*, 2017. [2.5](#)
- [5] Ali Borji. Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding*, 2019. [1.2](#)
- [6] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019. [1.2](#), [4.2.2](#), [5.1](#)
- [7] Dingding Cai, Ke Chen, Yanlin Qian, and Joni-Kristian Kämäräinen. Convolutional low-resolution fine-grained classification. *Pattern Recognition Letters*, 2019. [1.1](#), [3.1.2](#)
- [8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. [1.2](#), [2.6](#), [5.2.4](#), [5.2.4](#)
- [9] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *ICLR*, 2019. [3.1.1](#), [4.1](#), [4.3.1](#)
- [10] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *NeurIPS*, 2016. [3.1.2](#)
- [11] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich.

- Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *ICML*, 2018. 2.3
- [12] Zitian Chen, Yanwei Fu, Yu-Xiong Wang, Lin Ma, Wei Liu, and Martial Hebert. Image deformation meta-networks for one-shot learning. In *CVPR*, 2019. 2.1
- [13] Jaehoon Choi, Taekyung Kim, and Changick Kim. Self-ensembling with gan-based data augmentation for domain adaptation in semantic segmentation. In *ICCV*, 2019. 1.2, 2.4
- [14] Arun CS Kumar, Suchendra M Bhandarkar, and Mukta Prasad. Monocular depth prediction using generative adversarial networks. In *CVPRW*, 2018. 2.4
- [15] Harm De Vries, Florian Strub, J er mie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. Modulating early visual processing by language. In *NeurIPS*, 2017. 5.2.2
- [16] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 1977. 2.6, 5.2.3
- [17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 4.2.2
- [18] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *ICCV*, 2017. 2.3, 2.6
- [19] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *NeurIPS*, 2014. 2.6
- [20] DC Dowson and BV Landau. The fr chet distance between multivariate normal distributions. *Journal of multivariate analysis*, 1982. 4.2.2
- [21] Kieran Egan. Memory, imagination, and learning: Connected by the story. *Phi Delta Kappan*, 1989. 1
- [22] Kieran Egan. *Imagination in teaching and learning: The middle school years*. University of Chicago Press, 2014. 1
- [23] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015. 6.1
- [24] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *PAMI*, 2006. 1.1, 2.1
- [25] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 1.1, 1.2, 2.1, 4.1
- [26] Andrew Ford. *Modeling the environment: an introduction to system dynamics*

- models of environmental systems*. Island press, 1999. [2.5](#)
- [27] Maayan Frid-Adar, Idit Diamant, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification. *Neurocomputing*, 2018. [2.4](#)
- [28] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. [1.2](#)
- [29] Saurabh Gupta, Pablo Arbelaez, and Jitendra Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In *CVPR*, 2013. [6.1](#)
- [30] Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *ICCV*, 2017. [3.1.1](#)
- [31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. [1.1](#), [3.1.2](#), [4.1](#), [5.2.1](#)
- [32] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NeurIPS Workshops*, 2014. [3.1.2](#)
- [33] Zhiting Hu, Bowen Tan, Russ R Salakhutdinov, Tom M Mitchell, and Eric P Xing. Learning data manipulation for augmentation and weighting. In *NeurIPS*, 2019. [2.5](#)
- [34] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. [3.1.2](#)
- [35] Umar Iqbal, Pavlo Molchanov, and Jan Kautz. Weakly-supervised 3D human pose learning via multi-view images in the wild. *CVPR*, 2020. [2.2](#)
- [36] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, 2018. [2.3](#)
- [37] Iasonas Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *CVPR*, 2017. [2.3](#)
- [38] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 2015. [1.1](#), [2.1](#)
- [39] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *CVPR*, 2020. [4.4](#)
- [40] Wonhee Lee, Joonil Na, and Gunhee Kim. Multi-task self-supervised object detection via recycling of bounding box annotations. In *CVPR*, 2019. [2.6](#)

- [41] Aoxue Li, Tiange Luo, Tao Xiang, Weiran Huang, and Liwei Wang. Few-shot learning with global class representations. In *ICCV*, 2019. [2.1](#)
- [42] Yujia Li, Kevin Swersky, and Rich Zemel. Generative moment matching networks. In *ICML*, 2015. [2.1](#)
- [43] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *ICCV*, 2019. [4.5](#)
- [44] Qiuhua Liu, Xuejun Liao, and Lawrence Carin. Semi-supervised multitask learning. In *NeurIPS*, 2008. [2.6](#)
- [45] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, 2016. [3.1.2](#)
- [46] Pauline Luc, Camille Couprie, Soumith Chintala, and Jakob Verbeek. Semantic segmentation using adversarial networks. *arXiv preprint arXiv:1611.08408*, 2016. [2.4](#)
- [47] Canjie Luo, Yuanzhi Zhu, Lianwen Jin, and Yongpan Wang. Learn to augment: Joint data augmentation and network optimization for text recognition. In *CVPR*, 2020. [2.5](#)
- [48] Mateusz Michalkiewicz, Sarah Parisot, Stavros Tsogkas, Mahsa Baktashmotlagh, Anders Eriksson, and Eugene Belilovsky. Few-shot single-view 3-D object reconstruction with compositional priors. In *ECCV*, 2020. [2.3](#)
- [49] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *CVPR*, 2016. [2.3](#)
- [50] Siva Karthik Mustikovela, Varun Jampani, Shalini De Mello, Sifei Liu, Umar Iqbal, Carsten Rother, and Jan Kautz. Self-supervised viewpoint learning from image collections. In *CVPR*, 2020. [2.4](#), [2.5](#)
- [51] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgb-d images. In *ECCV*, 2012. [1.2](#), [6.1](#)
- [52] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. HoloGAN: Unsupervised learning of 3D representations from natural images. In *ICCV*, 2019. [1.1](#), [2.2](#), [3.1.2](#), [3.1.2](#), [3.1.2](#), [4.1](#)
- [53] Thu Nguyen-Phuoc, Christian Richardt, Long Mai, Yong-Liang Yang, and Niloy Mitra. BlockGAN: Learning 3D object-aware scene representations from unlabelled images. *NeurIPS*, 2020. [2.2](#), [4.5](#)
- [54] Thu H Nguyen-Phuoc, Chuan Li, Stephen Balaban, and Yongliang Yang. RenderNet: A deep convolutional network for differentiable rendering from 3D shapes. In *NeurIPS*, 2018. [1.1](#), [1.2](#), [2.2](#)
- [55] Atsuhiko Noguchi and Tatsuya Harada. RGBD-GAN: Unsupervised 3D repre-

- sentation learning from natural image datasets via rgb-d image synthesis. In *ICLR*, 2020. 4.5
- [56] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016. 2.6
- [57] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *ICCV*, 2017. 2.6
- [58] Arghya Pal and Vineeth N Balasubramanian. Zero-shot task transfer. In *CVPR*, 2019. 2.3
- [59] George Papandreou, Liang-Chieh Chen, Kevin P Murphy, and Alan L Yuille. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *ICCV*, 2015. 2.6
- [60] Eunbyung Park, Jimei Yang, Ersin Yumer, Duygu Ceylan, and Alexander C Berg. Transformation-grounded image generation network for novel 3d view synthesis. In *CVPR*, 2017. 1.1, 2.2
- [61] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016. 2.6
- [62] Joel Pearson. The human imagination: the cognitive neuroscience of visual mental imagery. *Nature Reviews Neuroscience*, 2019. 1
- [63] Etienne Pelaprat and Michael Cole. “minding the gap”: Imagination, creativity and human cognition. *Integrative Psychological and Behavioral Science*, 2011. 1
- [64] Xi Peng, Zhiqiang Tang, Fei Yang, Rogerio S Feris, and Dimitris Metaxas. Jointly optimize data augmentation and network training: Adversarial data augmentation in human pose estimation. In *CVPR*, 2018. 2.5
- [65] Anastasia Pentina and Christoph H Lampert. Multi-task learning with labeled and unlabeled tasks. In *ICML*, 2017. 2.3
- [66] Andrea Pilzer, Dan Xu, Mihai Puscas, Elisa Ricci, and Nicu Sebe. Unsupervised adversarial depth estimation using cycled generative networks. In *3DV*, 2018. 2.4
- [67] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016. 3.1.2
- [68] Sai Rajeswar, Fahim Mannan, Florian Golemo, Jérôme Parent-Lévesque, David Vazquez, Derek Nowrouzezahrai, and Aaron Courville. Pix2Shape: Towards unsupervised learning of 3D scenes from images using a view-based representation. *IJCV*, 2020. 2.2
- [69] Krishna Regmi and Ali Borji. Cross-view image synthesis using conditional gans. In *CVPR*, 2018. 1.1

- [70] Zhongzheng Ren and Yong Jae Lee. Cross-domain self-supervised multi-task feature learning using synthetic imagery. In *CVPR*, 2018. [2.6](#)
- [71] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017. [2.3](#), [4.2.3](#)
- [72] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NeurIPS*, 2016. [4.2.2](#)
- [73] Veit Sandfort, Ke Yan, Perry J Pickhardt, and Ronald M Summers. Data augmentation using generative adversarial networks (cyclegan) to improve generalizability in ct segmentation tasks. *Scientific reports*, 2019. [1.2](#), [2.4](#)
- [74] Shibani Santurkar, Andrew Ilyas, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Image synthesis with a single (robust) classifier. In *NeurIPS*, 2019. [2.3](#)
- [75] Eli Schwartz, Leonid Karlinsky, Joseph Shtok, Sivan Harary, Mattias Marder, Abhishek Kumar, Rogerio Feris, Raja Giryes, and Alex Bronstein. Delta-encoder: an effective sample synthesis method for few-shot object recognition. In *NeurIPS*, 2018. [2.1](#)
- [76] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *NeurIPS*, 2018. [2.3](#)
- [77] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. How good is my gan? In *ECCV*, 2018. [1.2](#)
- [78] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 2019. [2.4](#)
- [79] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3D feature embeddings. In *CVPR*, 2019. [1.1](#), [1.2](#), [2.2](#)
- [80] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017. [1.1](#), [2.1](#), [3.1.2](#), [3.1.2](#), [3.1.2](#), [4.1](#)
- [81] Nasim Souly, Concetto Spampinato, and Mubarak Shah. Semi supervised semantic segmentation using generative adversarial network. In *ICCV*, 2017. [1.2](#), [2.4](#)
- [82] Trevor Standley, Amir R Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *ICML*, 2020. [1.2](#), [2.3](#), [6.1](#)
- [83] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *CVPR*, 2019. [2.1](#)
- [84] Shao-Hua Sun, Minyoung Huh, Yuan-Hong Liao, Ning Zhang, and Joseph J Lim. Multi-view to novel view: Synthesizing novel views with self-learned

- confidence. In *ECCV*, 2018. [2.3](#)
- [85] Ximeng Sun, Rameswar Panda, and Rogerio Feris. Adashare: Learning what to share for efficient deep multi-task learning. *arXiv preprint arXiv:1911.12423*, 2019. [1.2](#), [2.3](#), [5.2.1](#), [6.1](#)
- [86] Sebastian Thrun. Is learning the n-th thing any easier than learning the first? In *NeurIPS*, 1996. [2.1](#)
- [87] Satoshi Tsutsui, Yanwei Fu, and David Crandall. Meta-reinforced synthetic data for one-shot fine-grained visual recognition. In *NeurIPS*, 2019. [2.1](#)
- [88] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *CVPR*, 2020. [2.2](#)
- [89] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NeurIPS*, 2016. [1.1](#), [2.1](#)
- [90] Yu-Xiong Wang and Martial Hebert. Learning to learn: Model regression networks for easy small sample learning. In *ECCV*, 2016. [1.1](#), [2.1](#)
- [91] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *CVPR*, 2018. [2.1](#), [2.3](#), [2.4](#), [4.1](#)
- [92] Zhangyang Wang, Shiyu Chang, Yingzhen Yang, Ding Liu, and Thomas S Huang. Studying very low resolution recognition using deep networks. In *CVPR*, 2016. [1.1](#), [3.1.2](#)
- [93] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. [4.1](#)
- [94] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. SynSin: End-to-end view synthesis from a single image. In *CVPR*, 2020. [2.2](#), [4.1](#)
- [95] Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. Supermasks in superposition. In *NeurIPS*, 2020. [2.2](#)
- [96] Yuhuai Wu, Yuri Burda, Ruslan Salakhutdinov, and Roger Grosse. On the quantitative analysis of decoder-based generative models. *arXiv preprint arXiv:1611.04273*, 2016. [1.2](#)
- [97] Yongqin Xian, Saurabh Sharma, Bernt Schiele, and Zeynep Akata. F-VAEGAN-D2: A feature generating framework for any-shot learning. In *CVPR*, 2019. [2.3](#)
- [98] Yang Xiao and Renaud Marlet. Few-shot object detection and viewpoint estimation for objects in the wild. In *ECCV*, 2020. [2.3](#)
- [99] Wei Xiong, Yutong He, Yixuan Zhang, Wenhan Luo, Lin Ma, and Jiebo Luo.

- Fine-grained image-to-image transformation towards visual recognition. In *CVPR*, 2020. [2.3](#)
- [100] Linjie Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. A large-scale car dataset for fine-grained categorization and verification. In *CVPR*, 2015. [4.1](#)
- [101] Yibo Yang, Zhisheng Zhong, Tiancheng Shen, and Zhouchen Lin. Convolutional neural networks with alternately updated clique. In *CVPR*, 2018. [2.5](#)
- [102] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *CVPR*, 2018. [2.2](#)
- [103] Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. In *CVPR*, 2020. [2.2](#), [4.1](#)
- [104] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *CoRL*, 2020. [1.1](#)
- [105] Amir R Zamir, Te-Lin Wu, Lin Sun, William B Shen, Bertram E Shi, Jitendra Malik, and Silvio Savarese. Feedback networks. In *CVPR*, 2017. [2.5](#)
- [106] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *CVPR*, 2018. [1.1](#), [1.2](#), [2.3](#), [5.2.1](#), [6.1](#)
- [107] Amir R Zamir, Alexander Sax, Nikhil Cheerla, Rohan Suri, Zhangjie Cao, Jitendra Malik, and Leonidas J Guibas. Robust learning through cross-task consistency. In *CVPR*, 2020. [2.3](#)
- [108] Fangneng Zhan, Shijian Lu, and Chuhui Xue. Verisimilar image synthesis for accurate detection and recognition of texts in scenes. In *ECCV*, 2018. [2.4](#)
- [109] Feifei Zhang, Tianzhu Zhang, Qirong Mao, and Changsheng Xu. A unified deep model for joint facial expression recognition, face synthesis, and face alignment. *TIP*, 2020. [2.5](#)
- [110] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *ICML*, 2019. [1.2](#), [1.2](#), [5.1](#), [5.2.1](#), [5.2.2](#)
- [111] Hongguang Zhang, Jing Zhang, and Piotr Koniusz. Few-shot learning via saliency-guided hallucination of samples. In *CVPR*, 2019. [2.1](#)
- [112] Ruixiang Zhang, Tong Che, Zoubin Ghahramani, Yoshua Bengio, and Yangqiu Song. MetaGAN: An adversarial approach to few-shot learning. In *NeurIPS*, 2018. [2.1](#)
- [113] Xinyue Zhu, Yifan Liu, Jiahong Li, Tao Wan, and Zengchang Qin. Emotion classification with data augmentation using generative adversarial networks. In

PAKDD, 2018. [1.2](#), [2.4](#)