Generative Modeling for Multi-task Visual Learning

Zhipeng Bao [1] **Martial Hebert** [1] **Yu-Xiong Wang** [2]

Abstract

Generative modeling has recently shown great promise in computer vision, but it has mostly focused on synthesizing visually realistic images. In this paper, motivated by multi-task learning of shareable feature representations, we consider a novel problem of learning a shared generative model that is useful across various visual perception tasks. Correspondingly, we 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 in different sample-size regimes.

1. Introduction

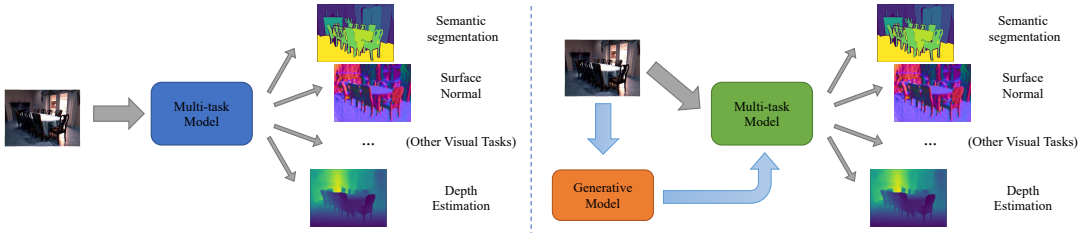
Seeing with the mind's eye – creating internal images of objects and scenes not actually present to the senses – is perhaps one of the hallmarks in human cognition (Pelaprat & Cole, 2011). For humans, this visual imagination integrates learning experience and facilitates learning by solving different problems (Egan, 1989; Pelaprat & Cole, 2011; Egan, 2014; Pearson, 2019). Inspired by such ability, there has been increasing interest in building generative models that can synthesize images (Goodfellow et al., 2014; Sohl-Dickstein et al., 2015; Van Den Oord et al., 2017; Kingma & Dhariwal, 2018; Wiles et al., 2020). Yet, most of the effort has focused on generating visually realistic images (Brock et al., 2019; Zhang et al., 2019), which are still far from being useful for machine perception tasks (Wu et al., 2017; Shmelkov et al., 2018; Borji, 2019). Even though recent work has started improving the "usefulness" of synthesized images, this line of investigation is often limited to a single specific task (Souly et al., 2017; Nguyen-Phuoc et al., 2018; Zhu et al., 2018; Sitzmann et al., 2019). Could we guide generative models to benefit *multiple* visual tasks?

While similar spirits of shareable feature representations have been widely studied as multi-task learning or meta-learning (Finn et al., 2017; Zamir et al., 2018), here we take a different perspective – *learning a shareable generative model across various tasks* (as illustrated in Figure 1). 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), 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 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 work as additional training data to build better task-specific models. These variations are especially critical when the data is limited.

This paper thus explores *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 normal prediction). In the single task scenario, existing work trains a separate generative model to synthesize paired pixel-level labeled data (Choi et al., 2019; Sandfort et al., 2019) and produce an augmented set. However, 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?

[1]Carnegie Mellon University [2]University of Illinois at Urbana-Champaign. Correspondence to: Zhipeng Bao <zbao@cs.cmu.edu>, Martial Hebert <hebert@cs.cmu.edu>, Yu-Xiong Wang <yxw@illinois.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.



(a) Traditional multi-task framework

(b) Multi-task oriented generative modeling (Ours)

Figure 1: **(a):** Traditional multi-task learning framework that learns a shared feature representation **vs. (b):** our proposed multi-task oriented generative modeling that additionally learns a shared generative model across various visual tasks.

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 to the discriminative multi-task network, we introduce a *refinement* network and a *self-supervision* network that satisfy 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 generative network synthesizes images conditioned on scene labels, leading to naturally paired image and scene-label data. The refinement network performs scene classification on the basis of the multi-task network predictions, which requires only scene labels. 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 (image-level) weakly-labeled synthesized images. We instantiate MGM with the representative encoder-decoder based multi-task network (Zamir et al., 2018), self-attention generative adversarial network (SAGAN) (Zhang et al., 2019), and contrastive learning based self-supervision network (Chen et al., 2020). 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 (Nathan Silberman & Fergus, 2012) and Taskonomy (Zamir et al., 2018) datasets. Consistent with the previous work (Standley et al., 2020; Sun et al., 2020), we focus on three representative tasks: semantic segmentation, depth estimation, and surface normal prediction. The evaluation shows that: **(1)** MGM consistently outperforms state-of-the-art multi-task approaches by large margins in different sample-size regimes. **(2)** With the increasing number of synthesized samples, the performance of MGM consistently improves and it also almost reaches the *performance upper-bound* that trains with weakly-annotated *real* images. **(3)** Our framework is scalable and can be extended to more visual tasks. The code of this work is available at `https://github.com/zpbao/multi-task-oriented_generative_modeling`.

Model	ST	ST_G
mLoss (↓)	0.111	0.148

Table 1: Pilot experiment for semantic segmentation on the Tiny-Taskonomy dataset. Directly using images synthesized by an off-the-shelf generative model (self-attention GAN) may hurt the performance on the downstream task. ST: single-task model trained on real images only; ST_G: the same model trained on both real and synthesized images.

2. Pilot Study

This pilot study provides an initial experimentation, which validates the importance and challenge of our proposed problem of task oriented generative modeling and further motivates the development of our method. Specifically, we show that *directly using images synthesized by an off-the-shelf generative model that is trained with the photo-realism objective is not helpful for downstream pixel-level perception tasks*. Such difficulty exists even for a single task, let alone for the more complicated multi-task scenario. Note that the goal here is *not* to motivate the specific components and design choices in our framework, which will be explained in Sec. 3.

Experimental Design: For ease of analysis, here we focus on a single task – semantic segmentation, and use the Tiny-Taskonomy dataset (Zamir et al., 2018). The dataset split and evaluation metric are the same as our main experiments (See Sec. 4 for details). We train a self-attention generative adversarial network (SAGAN) (Zhang et al., 2019) on Tiny-Taskonomy, and use it to generate the same number of synthesized images as the real images to augment the training set. Figure 6 (a) visualizes that the images synthesized by SAGAN are photo-realistic.

How to Generate Pixel-Level Annotations? One remaining question is how to generate pixel-level annotations for these images synthesized by SAGAN. While prior work

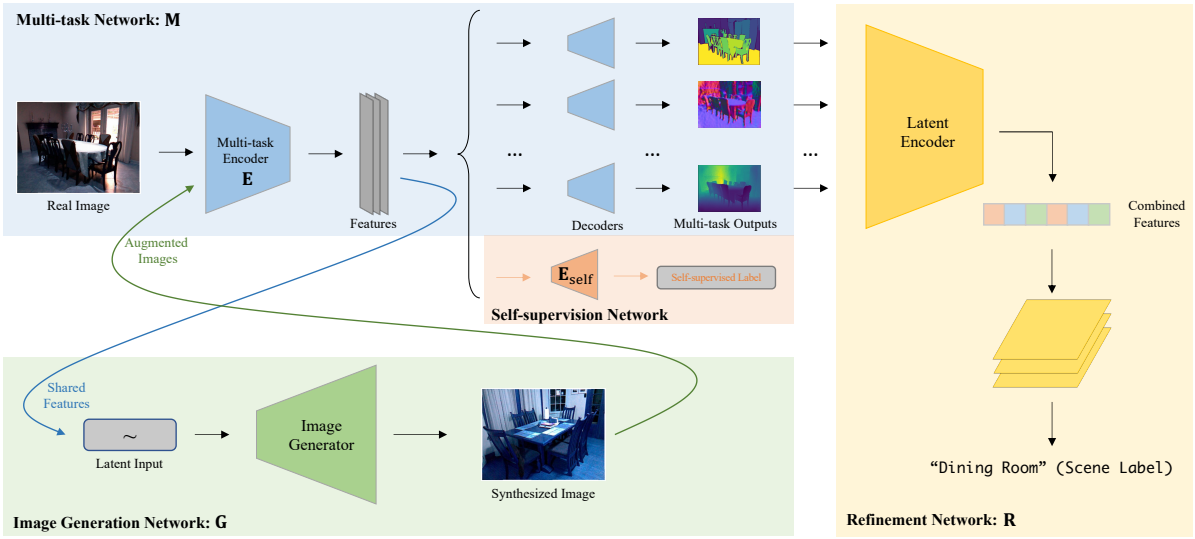


Figure 2: Architecture of our proposed multi-task oriented generative modeling (MGM) framework. There are four main components in the framework: Multi-task network to address the target multiple pixel-level prediction tasks; self-supervision network to facilitate representation learning using images without any annotation; refinement network to perform scene classification using weak annotation; image generation network to synthesize useful images that benefit multiple tasks.

has explored synthesizing both images and their pixel-level annotations for specific tasks (Choi et al., 2019; Sandfort et al., 2019), these annotations are still not reliable. For ease of analysis, in this study, we factor out the effect of annotations and assume that we have an *oracle annotator*. We use the annotator from Taskonomy (Zamir et al., 2018), which is a powerful fully-supervised semantic segmentation network. In fact, the ground-truth of semantic segmentation on Taskonomy is produced as the output of this network rather than labeled by humans. By doing so, we ensure that the annotations of the synthesized images are "accurate" and consistent with how the real images are labeled.

Comparisons: Single-Task (ST) model is our baseline which follows the architecture of the Taskonomy single-task network. ST is trained on real images only. $\mathbf{ST}_{\mathrm{G}}$ is the ST model trained on the augmented set. Table 1 reports the results of these two models, and $\mathbf{ST}_{\mathrm{G}}$ is worse than **ST**.

Do Photo-Realistic Images Synthesized off the Shelf Help Downstream Tasks? From Table 1, the answer is **NO**. Even though the images are synthesized to be photo-realistic (Figure 6 (a)) by one of the state-of-the-art generative models and are labeled by the oracle annotator, they still cannot benefit the downstream task. This is probably because these images are synthesized off the shelf *without* "knowing" the downstream task. *Our key insight* then is that we need to explicitly use the downstream task objective to guide the image synthesis process. Moreover, here we focused on a single task and assumed that we had the oracle annotations. However, an oracle annotator is difficult to obtain in practice, especially for multiple tasks. Also, existing work cannot synthesize paired images and pixel-level annotations for multiple tasks (Choi et al., 2019; Sandfort et al., 2019). To overcome these challenges, in what follows we demonstrate how to facilitate visual tasks with synthesized images that (i) are guided by the downstream task objective and (ii) only need image-level scene labels. Our approach is effective irrespective of a single task or multiple tasks.

3. Method

We propose multi-task oriented generative modeling (MGM) to leverage generative networks for multi-task visual learning, as summarized in Figure 2. 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 representative multi-task learning and image generation approaches. Finally, we discuss the detailed training strategy for the framework.

3.1. Problem Setting

Multi-task Discriminative Learning: Given a set of n visual tasks $\mathcal{T} = \{T_1, T_2, \cdots, T_n\}$, we aim to learn a discriminative multi-task model $\mathbf{M}$ that is able to address all of these tasks simultaneously: $\mathbf{M}(x) \rightarrow \widehat{\boldsymbol{y}} = (\widehat{y}^1, \widehat{y}^2, \cdots, \widehat{y}^n)$, where x is an input image and $\widehat{y}^i$ 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 $\mathcal{S}_{\mathrm{real}} = \{(x_j, y_j^1, y_j^2, \cdots, y_j^n, c_j)\}$.

Generative Learning: Meanwhile, we aim to learn a gen-

erative model **G** that produces a set of synthesized data but with only corresponding image-level scene labels (weak annotation): $\mathbf{G}(c, z) \rightarrow \widetilde{x}$, where z is a random input, and $\widetilde{x}$ is a synthesized image. The scene label of $\widetilde{x}$ is denoted as $\widetilde{c} = c$. We denote the set of synthesized images and their corresponding scene labels as $\widetilde{\mathcal{S}}_{\text{syn}} = \{(\widetilde{x}_k, \widetilde{c}_k)\}$.

Cooperation Between Discriminative and Generative Learning: Our objective is that the discriminative model **M** and the generative model **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 fully-labeled real data $\mathcal{S}_{\text{real}}$, and then the generative network **G** is trained to produce the synthesized set $\widetilde{\mathcal{S}}_{\text{syn}}$. Finally, **M** effectively learns from both $\mathcal{S}_{\text{real}}$ and $\widetilde{\mathcal{S}}_{\text{syn}}$. Note that, unlike most of the existing work on image generation (Brock et al., 2019; Zhang et al., 2019), *we do not focus on the visual realism of the synthesized images* $\widetilde{x}$. Instead, we hope **G** to capture the underlying image generation mechanism that benefits **M**.

3.2. Framework and Architecture

Figure 2 shows the architecture of our proposed MGM framework. It contains four components: the main discriminative multi-task network **M**, the image generation network **G**, the refinement network **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 **M** and **G** to better cooperate with each other. Notice that our MGM is a *model-agnostic* framework, and here we instantiate its components with representative models. In Sec. 4.3, we show that MGM works well with different choices of the model components.

Multi-task Network (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 (Zamir et al., 2018; Zhang et al., 2019; Sun et al., 2020). Considering the trade-off between model complexity and performance, we use a shared encoder **E** to extract features from input images, and individual decoders for each target task. We adopt a ResNet-18 (He et al., 2016) for the encoder and symmetric transposed decoders following Zamir et al. (2018). For each task, we have its own loss function to update the corresponding decoder and the shared encoder.

Image Generation Network (G): The generative model **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. **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 **G** with a self-attention generative adversarial network (SAGAN) (Zhang et al., 2019). We achieve conditional image generation by applying conditional batch normalization (CBN) layers (De Vries et al., 2017):

$$\text{CBN}\left(f_{i,c,h,w} \mid \gamma_c, \beta_c\right) = \gamma_c \frac{f_{i,c,w,h} - \text{E}\left[f_{\cdot,c,\cdot,\cdot}\right]}{\sqrt{\text{Var}\left[f_{\cdot,c,\cdot,\cdot}\right] + \epsilon}} + \beta_c, \quad (1)$$

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 also show the effectiveness of using DCGAN (Radford et al., 2015) in Sec. 4.3.

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 it only requires category labels, **R** can be effortlessly operationalized on the "weakly-annotated" synthesized images. Through refining the shared representation with the synthesized images, **R** also provides implicit guidance to the image generation network **G**, enforcing the semantic consistency of the synthesized images with **G**.

We use cross-entropy based scene classification loss to train the refinement network **R**. And we adopt two different strategies for real and synthesized images, respectively. For the fully-annotated real images $(x, \boldsymbol{y}, c)$, we use the classification loss to update **R** and then the encoder **E** in the multi-task network **M** with the decoders frozen. For the synthesized images $(\widetilde{x}, \widetilde{c})$, since their multi-task predictions produced by **M** might not be reliable, we apply an algorithm inspired by Expectation-Maximization (EM) (Dempster et al., 1977). At the Expectation step, we back-propagate the classification loss via **R** to estimate the *latent* multi-task ground-truth. At the Maximization step, we update the encoder **E** with **R** and the decoders frozen.

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

$$P(x, \boldsymbol{y}, c; \theta, \theta') = P(x) \left(\prod_{i=1}^{n} P\left(y^i \mid x; \theta\right) \right) P(c \mid \boldsymbol{y}; \theta'), \quad (2)$$

where x is an input image, $\boldsymbol{y}$ is the multi-task predictions, c is the scene label, θ is the parameters of the multi-task network, and θ' is the 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 both θ and θ'. In particular, θ' is updated for training the refinement network:

$$\theta'^{\star} = \underset{\theta'}{\text{argmax}}\, P(c \mid \boldsymbol{y}; \theta'). \quad (3)$$

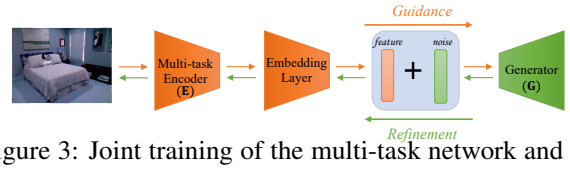


Figure 3: Joint training of the multi-task network and the image generation network. The multi-task network provides useful feature representation to guide the image generation process, while the generation network refines the shared representation through back-propagation.

Algorithm 1 Training procedure of MGM

Initialization:
$e_{\max}$: Maximum number of epochs for the training;
M: Multi-task network, **G**: Image generation network;
E: Multi-task encoder, **R**: Refinement network;
$\mathbf{E}_{\text{self}}$: Self-supervision network encoder;
N: minibatch size;
for $e = 1$ **to** $e_{\max}$ **do**
  Split $\mathcal{S}_{\text{real}}$ into minibatches with size N: $\mathcal{S}_{\text{mini}}$
  for $(x, \boldsymbol{y}, c) \in \mathcal{S}_{\text{mini}}$ **do**
    $\hat{\boldsymbol{y}} = \mathbf{M}(x)$
    $\mathcal{L}_{\text{multi}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) \rightarrow$ update **M**;
    $\hat{c} = \mathbf{R}(\hat{\boldsymbol{y}})$
    $\mathcal{L}_{\text{CE}}(c, \hat{c}) \rightarrow$ update **R**, **E**;
    Sample $2N$ augmented images x_{aug}
    $\mathcal{L}_{\text{NT-Xent}}(x_{\text{aug}}) \rightarrow$ update **E**, $\mathbf{E}_{\text{self}}$;
    Use $\mathcal{L}_{\text{GAN}}$ to train **G**;
    $(\widetilde{x}, \widetilde{c}) = \mathbf{G}(x, c)$, $\widetilde{c} = c$
    $\mathcal{L}_{\text{CE}}(\widetilde{c}, \mathbf{R}(\mathbf{M}(\widetilde{x}))) \rightarrow$ update **E** ;
    Sample $2N$ augmented synthesized images $\widetilde{x}_{\text{aug}}$
    $\mathcal{L}_{\text{NT-Xent}}(\widetilde{x}_{\text{aug}}) \rightarrow$ update **E**.
  end for
end for

For data samples in $\widetilde{\mathcal{S}}_{\text{syn}}$, we only update θ in an EM-like manner with θ' frozen. At the Expectation step, we estimate the latent multi-task ground-truth by:

$$\boldsymbol{y}^{\dagger} = \underset{\boldsymbol{y}}{\operatorname{argmax}}\, P\left(\boldsymbol{y} \mid \widetilde{x}; \theta\right) P(\widetilde{c} \mid \boldsymbol{y}; \theta'). \tag{4}$$

Then at the Maximization step, we back-propagate the error between $\boldsymbol{y}^{\dagger}$ and $\widehat{\boldsymbol{y}}$ (the multi-task predictions) to update θ (more precisely, the multi-task encoder with the decoders frozen):

$$\theta^{\star} = \underset{\theta}{\operatorname{argmax}}\, P\left(\boldsymbol{y}^{\dagger} \mid \widetilde{x}; \theta\right). \tag{5}$$

Self-supervision Network: The self-supervision network, operationalized on both real and synthesized images, facilitates representation learning of the encoder **E** by performing self-supervised learning tasks on images without any annotation. We modify SimCLR (Chen et al., 2020), one of the state-of-the-art approaches, as our self-supervision network.

This network contains an additional embedding network $\mathbf{E}_{\text{self}}$, working on the output of the multi-task encoder **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 (See Sec. A in the appendix), 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-scaled cross-entropy (*NT-Xent*) loss (Chen et al., 2020):

$$\ell_{i,j} = -\log \frac{\exp\left(\operatorname{dis}\left(\mu_i, \mu_j\right)/\tau\right)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp\left(\operatorname{dis}\left(\mu_i, \mu_k\right)/\tau\right)}, \tag{6}$$

where $\ell_{i,j}$ is the *NT-Xent* loss for a positive pair of examples (μ_i, μ_j) in the latent space. $\mathbb{1}_{[k \neq i]} \in 0, 1$ is an indicator function, evaluating to 1 if $k \neq i$, and τ is a temperature hyper-parameter. $\operatorname{dis}\left(\mu_i, \mu_j\right)$ is a distance function, and we use cosine distance following Chen et al. (2020). This loss is further back-propagated to refine the multi-task encoder **E**. Notice that other types of self-supervised tasks are applicable as well. To demonstrate this, in Sec. 4.3 we also report the result with another task – image reconstruction.

3.3. Interaction Among Networks

Cooperation Through Joint Training: We propose a simple but effective joint training algorithm shown in Figure 3. The image generation network **G** takes the feature representation of the multi-task encoder **E**, which is transformed via an additional embedding layer and 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 **G** will be back-propagated to **E** to further refine the shared representation. This process can be also viewed as introducing image generation as an additional task in the multi-task learning framework.

Training Procedure: We describe the procedure in Algorithm 1 and further explain it in the appendix.

4. Experiments

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.

4.1. Datasets and Compared Methods

Datasets: Following the work of Sun et al. (2020) and Standley et al. (2020), 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

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

Table 2: Main results (mean ± std) on the NYUv2 and Tiny-Taskonomy datasets. SS: semantic segmentation; DE: depth estimation; SN: surface normal prediction. ↑: higher is better; ↓: lower is better. We use different metrics on the two datasets, following the 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).

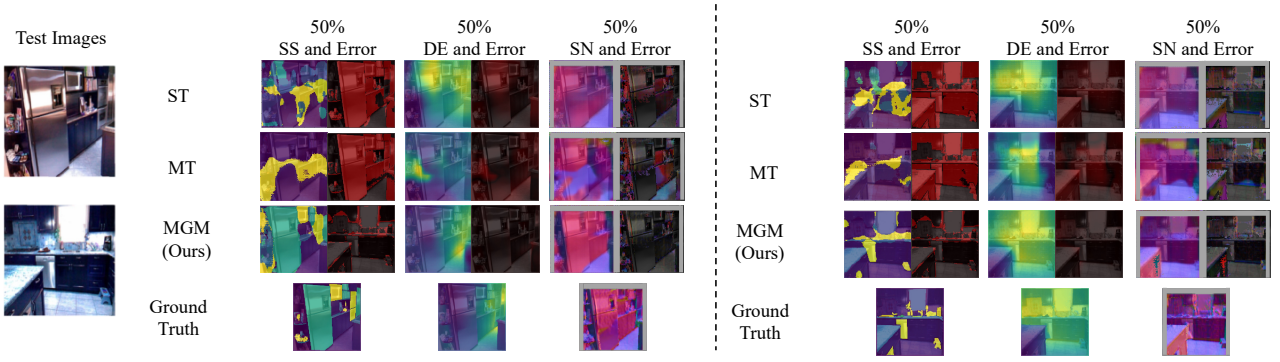


Figure 4: Visualization and error comparison of the multi-task prediction outputs in the 50% data setting on NYUv2. The prediction results of MGM are quite close to the ground-truth, significantly outperforming the baselines. The grey color in the SN ground-truth and results denotes that there is no annotation provided in the boundaries.

datasets: **NYUv2** (Nathan Silberman & Fergus, 2012; Eigen & Fergus, 2015) and **Tiny-Taskonomy** (Zamir et al., 2018). See Sec. D in the appendix for more details.

Compared Methods: We mainly focus on comparing with two widely-used discriminative baselines: **Single-Task (ST)** model follows the architecture of the Taskonomy single-task network (Zamir et al., 2018), and addresses each task individually; **Multi-Task (MT)** model refers to the sub-network for the three tasks of interest in Standley et al. (2020). 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 is produced by the generation network through joint training. In addition, to further validate the effectiveness of our **MGM** model, we consider its variant model $\mathbf{MGM}_r$ that is trained with both fully- and weakly-labeled *real* data. $\mathbf{MGM}_r$ 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 (Zamir et al., 2018).

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 $\mathbf{MGM}_r$, 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: We follow the standard metrics on these two datasets for comparison with prior work. For NYUv2, following the metrics in Eigen & Fergus (2015); Sun et al. (2020), 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 prediction task. For Tiny-Taskonomy, we follow the evaluation metrics of previous work (Zamir et al., 2018; Standley et al., 2020; Sun et al., 2020) and report the averaged loss values on the test set.

Model	$\text{MGM}_{/\text{G}}$	$\text{MGM}_{/\text{j}}$	MGM
SS-mIOU (↑)	0.243	0.243	**0.251**
DE-mABSE (↓)	0.799	0.763	**0.734**
SN-mAD (↓)	0.287	0.281	**0.273**

Table 3: Comparison of our MGM model with its variants on NYUv2. $\text{MGM}_{/\text{G}}$: *without* synthesizing images; $\text{MGM}_{/\text{j}}$: *without* joint learning. Our MGM outperforms single-task and multi-task baselines (Table 2) *even without synthesized data*, showing its effectiveness as a general multi-task learning framework. The model performance further improves with joint learning.

Implementation Details: See Sec. A in the appendix for the training details and the hyper-parameter sensitivity.

4.2. Main Results

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 2. We have the following key observations that support the effectiveness of our approach. (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 training with weakly-labeled synthesized data through the self-supervision network and the refinement network, the results of our model in the 50% data setting are sometimes even better than those of baselines in the 100% data setting. (3) More interestingly, 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. (4) The performance gap between the two models is especially minimal in the 25% labeled data setting, suggesting that our MGM model is, in particular, beneficial when the data is limited. This is further validated with additional experiments in even lower-data regimes in Sec. B in the appendix.

Qualitative Results: We also visualize the prediction results on the three tasks for ST, MT, and MGM in the 50% data setting in Figure 4 as well as Sec. E in the appendix. While obvious defects can be found for all the baselines, the results of our MGM are quite close to the ground-truth.

4.3. Analysis and Ablation Study

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

How Does Generative Modeling Benefit Multiple Tasks? We further consider two variants of our MGM model: $\textbf{MGM}_{/\text{G}}$ 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. $\textbf{MGM}_{/\text{j}}$ is trained with images synthesized by a pre-trained SAGAN *without* the joint training mechanism. Table 3 shows the results on NYUv2. The full results are shown in Sec. C.6 in the appendix.

Model	SS-mIOU (↑)	DE-mABSE (↓)	SN-mAD (↓)
$\text{MGM}_{/\text{self}}$	0.239	0.776	0.279
$\text{MGM}_{/\text{refine}}$	**0.254**	0.808	0.290
$\text{MGM}_{\text{recon}}$	0.241	0.768	0.285
$\text{MGM}_{\text{DCGAN}}$	0.245	0.750	0.285
MGM	0.251	**0.734**	**0.273**

Table 4: Ablation study on NYUv2. (1) $\text{MGM}_{/\text{self}}$: *without* the self-supervision network; (2) $\text{MGM}_{/\text{refine}}$: *without* the classification refinement network; (3) $\text{MGM}_{\text{recon}}$: *with* a simple reconstruction task as self-supervision; (4) $\text{MGM}_{\text{DCGAN}}$: *with* a naive generative model, DCGAN. The refinement network is more crucial to the depth estimation and surface normal prediction tasks, while the self-supervision network is more crucial to the semantic segmentation task. Their combination achieves the best performance. Both (3) and (4) consistently outperform single-task (ST) and multi-task (MT) baselines, indicating the robustness and flexibility of MGM.

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

Impact of Self-supervision and Refinement Networks: Two important components of the proposed framework are the self-supervision and refinement networks. We evaluate their impact individually in Table 4. $\text{MGM}_{/\text{self}}$ is the model trained *without* the self-supervision network; $\text{MGM}_{/\text{refine}}$ is the model *without* the refinement network. We could see that the refinement network is more crucial to the depth estimation and surface normal prediction tasks, while the self-supervision network is more crucial to the semantic segmentation task. They are complementary to each other, and combining them generally achieves the best performance.

MGM Is Model-Agnostic: MGM is flexible with different choices of model components. Here we show its flexibility for the image generation and self-supervision networks. $\text{MGM}_{\text{recon}}$ replaces the SimCLR based self-supervision method with a weaker reconstruction task; $\text{MGM}_{\text{DCGAN}}$ replaces SAGAN with a weaker generative network DCGAN (Radford et al., 2015). The results are shown in Table 4. Combining with Table 2, both variants consistently improve the performance on all the tasks even with weaker components, indicating the generalizability and robustness of our MGM framework. In addition, we find that MGM outperforms $\text{MGM}_{\text{DCGAN}}$, suggesting that a more powerful

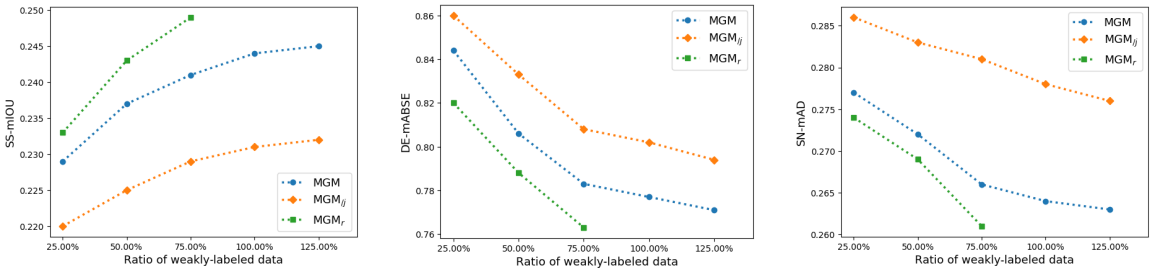


Figure 5: Performance change with respect to different ratios of weakly-labeled data on NYUv2. Joint learning significantly improves the performance. The performance of MGM keeps increasing with more weakly-labeled *synthesized* images, achieving results almost comparable to that of MGM$_r$ trained with all the available weakly-labeled *real* images.

image generation network leads to better performance.

Number of Synthesized Images vs. Real images: From the previous results, we have found that the synthesized images could benefit the target multiple 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 5 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*, 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.

Visual Realism vs. Downstream Task Usefulness of Synthesized Images: The evaluation so far has focused on the multi-task learning performance, without consideration of the visual realism of synthesized images which is a conventional way to evaluate generative models. Such a protocol is consistent with our main objective of introducing generative models to facilitate multi-task learning. Here we further investigate the visual quality of synthesized images both qualitatively and quantitatively to better understand the *difference between visual realism and usefulness to downstream tasks*. To this end, Figure 6 visualizes the images synthesized by SAGAN and MGM on Tiny-Taskonomy. We observe that **(1)** the conventional SAGAN, trained with the photo-realism objective and without the guidance of downstream tasks, produces visually appealing images; **(2)** the visual quality of images synthesized by MGM becomes degraded, where SAGAN is jointly trained with the multi-task learning objective and under the guidance of downstream tasks. Interestingly, a similar phenomenon has been observed in Souly et al. (2017), where a generative model is used to facilitate the semantic segmentation task.

(a) SAGAN **(b) MGM**

Figure 6: Comparison of images synthesized by SAGAN and MGM on Tiny-Taskonomy. **(a):** The images synthesized by *off-the-shelf* SAGAN are photo-realistic. **(b):** By contrast, after jointly training with the discriminative network under our MGM framework, the synthesized images are not visually realistic, but they are *helpful to improve the downstream task performance*.

We hypothesize that this is because those synthesized images that are useful for improving downstream tasks might not be necessarily photo-realistic. While the images synthesized by MGM are not visually realistic, they may contain some crucial discriminative information that can be leveraged for addressing downstream tasks – for example, the synthesized images may contain some unseen patterns from the real images, which increases the diversity of the training data. In addition, the difference of synthesized images between SAGAN and MGM can also partially explain the result in the pilot study (Sec. 2) – the images synthesized off the shelf are quite different from the desired images for multi-task learning, and thus they are not effective in facilitating downstream tasks.

Furthermore, we investigate the *training behavior* of MGM, measured by the change of the Fréchet inception distance (FID) of images synthesized by the generation network and the averaged prediction loss values on downstream tasks by the discriminative network. As shown in Figure 7, when we start jointly training the two networks, FID (visual quality) drops but the performance on downstream tasks continually improves, which is consistent with our model design.

Additional ablation studies on the impact of hyper-parameters, training strategies, and higher-resolution images as well as the generalization capability of MGM are provided in Sec. C in the appendix.

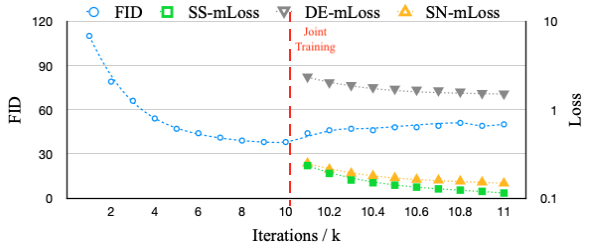


Figure 7: Training curve of MGM on Tiny-Taskonomy. When we start jointly training the generation and discriminative networks, FID (visual quality) of synthesized images drops, but the performance on downstream tasks continually improves. This indicates that the generation network in MGM is optimizing downstream task usefulness of synthesized images at the expense of their degraded visual quality.

Model	SS (↓)	DE (↓)	SN (↓)	ET (↓)	Re (↓)	PC (↓)
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 5: Mean test losses for six tasks on Tiny-Taskonomy. Again, our MGM outperforms the baselines, indicating its flexibility, generalizability, and scalability.

4.4. 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 5. Again, our proposed method still outperforms state-of-the-art baselines.

5. Related Work

Multi-task Learning and Task Relationship: Multi-task learning aims to leverage information from related tasks to benefit each individual task (Doersch & Zisserman, 2017). Most recent work can be grouped into two types of strategies (Ruder, 2017): hard parameter sharing (Doersch & Zisserman, 2017; Kokkinos, 2017; Pentina & Lampert, 2017) and soft parameter sharing (Misra et al., 2016; Chen et al., 2018; Sener & Koltun, 2018). These strategies have achieved good performance when the tasks are similar. In addition, relationships among different tasks have been studied to improve their cooperation. For example, *Taskonomy* exploits the relationships among various visual tasks to benefit transfer or multi-task learning (Zamir et al., 2018). Task cooperation and competition are considered (Standley et al., 2020), in a way of assigning tasks to a few neural networks to balance all of them. Some follow-up work also explores task relationships among different types of tasks (Armeni et al., 2019; Pal & Balasubramanian, 2019; Sun et al., 2020; Zamir et al., 2020; Wallace et al., 2021; Yeo et al., 2021), mainly in the paradigm of discriminative learning. In comparison, our work is the *first* that introduces generative modeling to multi-task visual learning.

Generative Modeling for Visual Learning: While the initial goal of generative models is to synthesize realistic images, some recent work has explored their potential to synthesize "usefull" images for downstream visual tasks (Shorten & Khoshgoftaar, 2019), including classification (Frid-Adar et al., 2018; Zhan et al., 2018; Zhu et al., 2018), semantic segmentation (Luc et al., 2016; Souly et al., 2017), and depth estimation (Aleotti et al., 2018; Pilzer et al., 2018). This is often achieved by generating images and corresponding annotations off the shelf and using them as data augmentation for a target visual task (Wang et al., 2018; Choi et al., 2019; Sandfort et al., 2019; Bao et al., 2021; Gui et al., 2021). Another strategy to leverage generative models is through well-designed error feedback or adversarial training (Luc et al., 2016; CS Kumar et al., 2018; Mustikovela et al., 2020). Different from prior work, MGM is applicable to *various visual tasks jointly* and different generative networks.

Learning with Less Labeling: Recent work takes advantage of weakly-labeled or unlabeled data by assigning some self-created labels, *e.g.*, via colorization, rotation, or reconstruction (Dosovitskiy et al., 2014; Noroozi & Favaro, 2016; Pathak et al., 2016; Noroozi et al., 2017; Chen et al., 2020). Similar self-supervised techniques have been proved useful for multi-task learning (Liu et al., 2008; Doersch & Zisserman, 2017; Ren & Jae Lee, 2018; Lee et al., 2019). Among these techniques, a notable one is the Expectation-Maximization (EM) algorithm (Dempster et al., 1977; Papandreou et al., 2015), which leverages the information of weakly-labeled or unlabeled data by iteratively estimating and refining their labels. We adopt a similar spirit and introduce the refinement network for the MGM framework.

6. Conclusion

This work introduces multi-task oriented generative modeling (MGM) that improves the usefulness of synthesized images to downstream tasks, instead of optimizing their photo-realism as is normally the case. A main challenge is that current generative models cannot synthesize both RGB images and pixel-level annotations in multi-task scenarios. We address this problem by equipping the MGM framework with the self-supervision and refinement networks, which enable us to take advantage of synthesized images paired with image-level scene labels to facilitate multiple visual tasks. Experimental results demonstrate that MGM consistently outperforms state-of-the-art multi-task approaches.

Acknowledgement: This work was supported in part by ONR MURI N000014-16-1-2007 and AFRL Grant FA23861714660. YXW was supported in part by NSF Grant 2106825, the Jump ARCHES endowment through the Health Care Engineering Systems Center, and the New Frontiers Initiative.

References

Aleotti, F., Tosi, F., Poggi, M., and Mattoccia, S. Generative adversarial networks for unsupervised monocular depth prediction. In *ECCV Workshops*, 2018.

Armeni, I., He, Z.-Y., Gwak, J., Zamir, A. R., Fischer, M., Malik, J., and Savarese, S. 3D scene graph: A structure for unified semantics, 3D space, and camera. In *ICCV*, 2019.

Bao, Z., Wang, Y.-X., and Hebert, M. Bowtie networks: Generative modeling for joint few-shot recognition and novel-view synthesis. *ICLR*, 2021.

Borji, A. Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding*, 2019.

Brock, A., Donahue, J., and Simonyan, K. Large scale gan training for high fidelity natural image synthesis. In *ICLR*, 2019.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

Chen, Z., Badrinarayanan, V., Lee, C.-Y., and Rabinovich, A. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *ICML*, 2018.

Choi, J., Kim, T., and Kim, C. Self-ensembling with gan-based data augmentation for domain adaptation in semantic segmentation. In *ICCV*, 2019.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

CS Kumar, A., Bhandarkar, S. M., and Prasad, M. Monocular depth prediction using generative adversarial networks. In *CVPR Workshops*, 2018.

De Vries, H., Strub, F., Mary, J., Larochelle, H., Pietquin, O., and Courville, A. C. Modulating early visual processing by language. In *NeurIPS*, 2017.

Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 1977.

Doersch, C. and Zisserman, A. Multi-task self-supervised visual learning. In *ICCV*, 2017.

Dosovitskiy, A., Springenberg, J. T., Riedmiller, M., and Brox, T. Discriminative unsupervised feature learning with convolutional neural networks. In *NeurIPS*, 2014.

Egan, K. Memory, imagination, and learning: Connected by the story. *Phi Delta Kappan*, 1989.

Egan, K. *Imagination in teaching and learning: The middle school years*. University of Chicago Press, 2014.

Eigen, D. and Fergus, R. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., and Greenspan, H. Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification. *Neurocomputing*, 2018.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *NeurIPS*, 2014.

Gui, L., Bardes, A., Salakhutdinov, R., Hauptmann, A., Hebert, M., and Wang, Y.-X. Learning to hallucinate examples from extrinsic and intrinsic supervision. In *ICCV*, 2021.

Gupta, S., Arbelaez, P., and Malik, J. Perceptual organization and recognition of indoor scenes from rgb-d images. In *CVPR*, 2013.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. *NeurIPS*, 2018.

Kokkinos, I. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *CVPR*, 2017.

Lee, W., Na, J., and Kim, G. Multi-task self-supervised object detection via recycling of bounding box annotations. In *CVPR*, 2019.

Liu, Q., Liao, X., and Carin, L. Semi-supervised multitask learning. In *NeurIPS*, 2008.

Luc, P., Couprie, C., Chintala, S., and Verbeek, J. Semantic segmentation using adversarial networks. *arXiv preprint arXiv:1611.08408*, 2016.

Misra, I., Shrivastava, A., Gupta, A., and Hebert, M. Cross-stitch networks for multi-task learning. In *CVPR*, 2016.

Mustikovela, S. K., Jampani, V., Mello, S. D., Liu, S., Iqbal, U., Rother, C., and Kautz, J. Self-supervised viewpoint learning from image collections. In *CVPR*, 2020.

Nathan Silberman, Derek Hoiem, P. K. and Fergus, R. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

Nguyen-Phuoc, T. H., Li, C., Balaban, S., and Yang, Y. Rendernet: A deep convolutional network for differentiable rendering from 3D shapes. In *NeurIPS*, 2018.

Noroozi, M. and Favaro, P. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016.

Noroozi, M., Pirsiavash, H., and Favaro, P. Representation learning by learning to count. In *ICCV*, 2017.

Pal, A. and Balasubramanian, V. N. Zero-shot task transfer. In *CVPR*, 2019.

Papandreou, G., Chen, L.-C., Murphy, K. P., and Yuille, A. L. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *ICCV*, 2015.

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., and Efros, A. A. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.

Pearson, J. The human imagination: the cognitive neuroscience of visual mental imagery. *Nature Reviews Neuroscience*, 2019.

Pelaprat, E. and Cole, M. "minding the gap": Imagination, creativity and human cognition. *Integrative Psychological and Behavioral Science*, 2011.

Pentina, A. and Lampert, C. H. Multi-task learning with labeled and unlabeled tasks. In *ICML*, 2017.

Pilzer, A., Xu, D., Puscas, M., Ricci, E., and Sebe, N. Unsupervised adversarial depth estimation using cycled generative networks. In *3DV*, 2018.

Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Ren, Z. and Jae Lee, Y. Cross-domain self-supervised multi-task feature learning using synthetic imagery. In *CVPR*, 2018.

Ruder, S. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

Sandfort, V., Yan, K., Pickhardt, P. J., and Summers, R. M. Data augmentation using generative adversarial networks (cyclegan) to improve generalizability in ct segmentation tasks. *Scientific reports*, 2019.

Sener, O. and Koltun, V. Multi-task learning as multi-objective optimization. In *NeurIPS*, 2018.

Shmelkov, K., Schmid, C., and Alahari, K. How good is my gan? In *ECCV*, 2018.

Shorten, C. and Khoshgoftaar, T. M. A survey on image data augmentation for deep learning. *Journal of Big Data*, 2019.

Sitzmann, V., Thies, J., Heide, F., Nießner, M., Wetzstein, G., and Zollhofer, M. Deepvoxels: Learning persistent 3D feature embeddings. In *CVPR*, 2019.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015.

Souly, N., Spampinato, C., and Shah, M. Semi supervised semantic segmentation using generative adversarial network. In *ICCV*, 2017.

Standley, T., Zamir, A. R., Chen, D., Guibas, L., Malik, J., and Savarese, S. Which tasks should be learned together in multi-task learning? In *ICML*, 2020.

Sun, X., Panda, R., and Feris, R. Adashare: Learning what to share for efficient deep multi-task learning. In *NeurIPS*, 2020.

Van Den Oord, A., Vinyals, O., et al. Neural discrete representation learning. *NeurIPS*, 2017.

Wallace, B., Wu, Z., and Hariharan, B. Can we characterize tasks without labels or features? In *CVPR*, 2021.

Wang, Y.-X., Girshick, R., Hebert, M., and Hariharan, B. Low-shot learning from imaginary data. In *CVPR*, 2018.

Wiles, O., Gkioxari, G., Szeliski, R., and Johnson, J. Synsin: End-to-end view synthesis from a single image. In *CVPR*, 2020.

Wu, Y., Burda, Y., Salakhutdinov, R., and Grosse, R. On the quantitative analysis of decoder-based generative models. In *ICLR*, 2017.

Yeo, T., Kar, O. F., and Zamir, A. Robustness via cross-domain ensembles. In *ICCV*, 2021.

Zamir, A. R., Sax, A., Shen, W., Guibas, L. J., Malik, J., and Savarese, S. Taskonomy: Disentangling task transfer learning. In *CVPR*, 2018.

Zamir, A. R., Sax, A., Cheerla, N., Suri, R., Cao, Z., Malik, J., and Guibas, L. J. Robust learning through cross-task consistency. In *CVPR*, 2020.

Zhan, F., Lu, S., and Xue, C. Verisimilar image synthesis for accurate detection and recognition of texts in scenes. In *ECCV*, 2018.

Zhang, H., Goodfellow, I., Metaxas, D., and Odena, A. Self-attention generative adversarial networks. In *ICML*, 2019.

Zhu, X., Liu, Y., Li, J., Wan, T., and Qin, Z. Emotion classification with data augmentation using generative adversarial networks. In *PAKDD*, 2018.

Method	SS-mIOU (↑)	DE-mABSE (↓)	SN-mAD (↓)
ST (Zamir et al., 2018)	0.199	0.908	0.312
MT (Zamir et al., 2018)	0.207	0.874	0.296
TaskGrouping (Standley et al., 2020)	0.215	0.853	0.292
Cross Stitch (Ren & Jae Lee, 2018)	0.205	0.917	0.296
AdaShare (Sun et al., 2020)	0.211	0.875	0.289
MGM	**0.229**	**0.844**	**0.277**

Table 6: Comparison with state-of-the-art multi-task models in the 25% data setting on the NYUv2 dataset. Notably, with a simple shared encoder architecture, our MGM model outperforms other state-of-the-art multi-task networks with more sophisticated architectures, which indicates the benefit of introducing generative modeling for multi-task learning. In addition, our MGM is a model-agnostic framework and could be incorporated with these different multi-task models for further improvement.

Method	SS-mIOU(↑)
ST	0.57
MGM	**0.64**

Table 7: Results on the CityScape Subset. MGM still outperforms the baseline ST model, indicating the robustness and generalizability of the model with multi-hot object labels.

We summarize the content of the appendix as follows. Sec. A includes additional details of model architecture of our proposed multi-task oriented generative modeling (MGM) framework. Sec. B provides additional experimental evaluations including the comparison with other state-of-the-art multi-task models, experiments with other datasets, and investigations in the few-shot regime. Section C provides additional ablation studies including the impact of parameters, different training strategies, training with higher resolution images, and the generalizibility of the shared feature representation. Sec. D describes implementation details of MGM and also the dataset settings. Finally, Sec. E shows more prediction visualizations.

A. Additional Details of Model Architecture

Multi-task Network: The multi-task network contains a shared encoder network and separate decoder networks for target tasks. We use a ResNet-18 (He et al., 2016) as the encoder network, and its architecture follows the standard Pytorch implementation[1]. We only change the size of the features of each layer group from $[64, 128, 256, 512]$ to $[48, 96, 192, 360]$, so as to better address our GPU memory constraints. The size of the final feature representation is $(360, 8, 8)$. For the decoder network, we use the same architecture as in Taskonomy (Zamir et al., 2018).

Self-supervision Network: We adopt SimCLR (Chen et al., 2020) as our self-supervision network. SimCLR is one of the state-of-the-art self-supervised learning approaches based on instance-level discrimination tasks. Following (Chen et al., 2020), we randomly apply 5 types of transformations on a source image to obtain an augmented image. These transformations are as follows: (1) random resizing and cropping followed by resizing back to the original size; (2) random horizontal flipping with probability of 0.5; (3) random color jittering with probability of 0.5; (4) random transformation of RGB images to gray-scale images with probability of 0.2; (5) random Gaussian blur with probability of 0.5. The shape of the transformed latent feature, which is used to perform contrastive learning, is $(128,)$.

Refinement Network: The refinement network takes as input the prediction results of the multi-task network. For each individual prediction, we apply a ResNet-10 (He et al., 2016) as the refinement encoder to extract the features. The feature dimension of each layer group of the refinement encoder is the same as the multi-task encoder network. Then we concatenate all the features together and apply a fully-connected layer with the hidden size of 128 to obtain the final scene class prediction.

Image Generation Network: We have instantiated the image generation network with two widely used generative networks: self-attention GAN (SAGAN)[2] (Zhang et al., 2019) in the main paper and deep convolutional GAN (DCGAN)[3] (Radford et al., 2015) in Sec. 4.3 of this document. For DCGAN, we change the original batch-normalization layers to conditional batch normalization layers (De Vries et al., 2017) to allow conditional image generation. For the additional embedding layer used for joint learning, we use a simple global averaged pooling layer followed by a dense layer.

[1] https://github.com/pytorch/vision/blob/master/torchvision/models/resnet.py
[2] https://github.com/voletiv/self-attention-GAN-pytorch
[3] https://github.com/Natsu6767/DCGAN-PyTorch

Method	SS-mIOU (↑)	DE-mABSE (↓)	SN-mAD (↓)
ST	0.162	1.004	0.337
MT	0.185	0.930	0.311
MGM	**0.197**	**0.911**	**0.291**

Table 8: Comparison in the few-shot regime – in the 10% data setting on the NYUv2 dataset where around 3 images for each scene is used as the training set. Again, MGM significantly outperforms the compared models, showing the benefit of generative models in the *extremely low-data regime*.

Model	SS (↓)	DE (↓)	SN (↓)
ST	0.137	1.836	0.161
MT	0.156	1.807	0.162
MGM	**0.125**	**1.670**	**0.153**

Table 9: Comparison with extreme low data in Taskonomy. In this data setting, MGM significantly outperforms both ST and MT, indicating that MGM is robust and especially helpful in low-data regime.

B. Additional Experimental Evaluations

B.1. Comparison with Other Multi-task Models

In the main paper, for a fair comparison we focused on comparing our MGM model with internal models (*e.g.*, the multi-task model upon which MGM builds). To have a more comprehensive understanding of the performance of MGM, we also compare our method with some state-of-the-art multi-task models. We focus on the 25% data setting for the NYUv2 dataset, where collaboration between different tasks and the utilization of data is vitally important.

We include six models in this experiments. **ST** is the single-task model, where the encoder and the decoder are adopted from Zamir et al. (2018). **MT** is the multi-task model that uses a shared encoder as ST and separate decoders. **ST** and **MT** are the baselines compared in the main paper. **TaskGrouping** uses the optimal network for the three tasks concluded from Standley et al. (2020). Another two well-performing multi-task models are **Cross-stitch** (Ren & Jae Lee, 2018)[4] and **AdaShare** (Sun et al., 2020). **MGM** is our proposed model. Table 6 summarizes the results. *Notably, with a simple shared encoder architecture, our MGM model outperforms other state-of-the-art multi-task networks with more sophisticated architectures, which indicates the benefit of introducing generative modeling for multi-task learning*. In addition, our MGM is a *model-agnostic framework* and could be incorporated with these different multi-task models for further improvement.

B.2. Experimental Evaluation on CityScape Subset

In this section, we demonstrate that MGM can work with datasets when no image-level labels are available. In an alternative may, the proposed MGM frame model can work with object labels as well since the generative network and refinement network can naturally work with multi-hot labels — the refinement network can work with a multi-label classifier, and the generative network can be a multi-label-conditional GAN.

We conducted semantic segmentation on a subset of CityScape (Cordts et al., 2016) dataset, the Zurich street scene. We focused on the semantic segmentation task, which is a representative task on CityScape. We use the 30 standard multi-hot CityScape semantic object labels for the generative model and also the refinement networks. We use 80% of the data for training and 20 % for testing and compare the performance of **ST** and **MGM** for this experiment. We generate the same amount of data with random multi-hot labels using MGM. The results are shown in Table 7. MGM still outperforms the baseline ST model, indicating the robustness and generalizability of the model with multi-hot object labels.

B.3. Experiments with Few-Shot Setting

Since the learned generative model facilitates flow of knowledge across tasks and provides meaningful variations in existing images, it is especially beneficial in the low-data regime. So we designed a 10% data setting for NYUv2 dataset, where around 3 images for each scene is used as the training set. We also compare our MGM model with ST and MT. From Table 8, we can see that MGM outperforms the other compared models significantly, indicating the gain of generative models in the *extremely low-data regime*.

B.4. Experiments with Extreme Low data at Taskonomy

We noticed that for Tiny-Taskonomy dataset, 25% data setting is still far from low-data regime. To further explore the effectiveness with our model with low-data, we further conduct an experiment with a subset of Tiny-Taskonomy dataset. We randomly select 3 nodes (allensville, benevolence, and coffeen) from Tiny-Taskonomy dataset to build a dataset with

[4]We modify the network architecture following Sun et al. (2020) to make it work for the three tasks.

Model	SS-mIOU (↑)	DE-mABSE (↓)	SN-mAD (↓)
ST	0.230	0.837	0.309
MT	0.237	0.819	0.291
ST_l	0.232	0.841	0.304
MT_l	0.236	0.804	0.288

Table 10: Impact of parameters. ST_l and MT_l: baselines with a larger number of parameters (with deeper backbones). Simply increasing the number of parameters cannot significantly boost performance.

Model	SS-mIOU (↑)	DE-mABSE (↓)	SN-mAD (↓)
ST	0.230	0.837	0.309
MGM-SS	0.244	-	-
MGM-DE	-	0.752	-
MGM-SN	-	-	0.277
MGM-Combine	0.249	0.747	0.277
MGM	**0.251**	**0.734**	**0.273**

Table 11: Ablation with MGM for single tasks and a stronger baseline with the learned information from the three individual tasks but without jointly training. The experiments are conducted on NYUv2 50% data setting. MGM-SS, MGM-DE, MGM-SN: variantal MGM model for single tasks. MGM-Combine: MGM variant trained with augmented images generated by the above three models. The proposed MGM framework can consistently benefit each single tasks and MGM-combine cannot reach the performance of MGM, indicating the importance of joint training mechanism.

17,404 images—around 5% data setting compared with the full Tiny-Taskonomy dataset. We then conduct experiments with ST, MT and MGM for this subset. All the other experimental settings keep the same as the main paper. Table 9 shows the comparable results. Combining the results in Table 2, we can find that MGM consistently outperforms ST and MT and is robust and especially helpful in low-data regime.

C. Additional Ablation Study

C.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: $\mathbf{ST}_l$ and $\mathbf{MT}_l$ use ResNet-34 as the encoder network and the corresponding decoder networks. These two networks have a similar amount of parameters as MGM. The result in Table 10 show that simply increasing the number of parameters cannot significantly boost performance.

C.2. Ablation with Single Tasks

MGM is a general framework that can be applied to both single tasks and multiple tasks. In the main submission, we mainly focused on the more challenging multi-task scenario. In this subsection, we conduct experiments with single tasks. Here we add three baselines applying MGM to the single tasks (SS, DE, SN) named **MGM-SS**, **MGM-DE** and **MGM-SN** in the NYUv2 50% data setting. We further provide an additional baseline by using the equivalently sampled data from the above three model as the augmented data, but not jointly train the generative network, named **MGM-Combine**. The results of these models are shown in Table 11. We have the following observations: (1) The proposed MGM framework can consistently benefit each single tasks though without leveraging shared features from multiple tasks. (2) Compared with the full MGM model, the performance drops when only using single tasks to jointly train with the generative model. (3) When using the individually optimized generative models, the performance is slight better than $\text{MGM}_{/\text{j}}$ but still could not reach MGM, indicating the importance of our joint training mechanism.

C.3. Training Strategies for Refinement Network

In the main paper, we proposed an Expectation-Maximum (EM) like algorithm to coordinate the training between the refinement network and the main network. Here we compare our **EM-Like Training (EML)** with two alternative ways of the training procedure: **Plain End-to-End Training (PEoE)** backwards the refinement loss to update the entire network directly; **Loosely Separate Training (LSeT)** trains the refinement network and the main network separately, and *only* backwards the error to the encoder network when dealing with weakly labeled images. Table 12 shows the comparison

Method	SS-mIOU (↑)	DE-mABSE (↓)	SN-mAD (↓)
MT	0.237	0.815	0.291
PEoE	0.211	0.896	0.301
LSeT	0.247	0.768	0.277
EML	**0.251**	**0.734**	**0.273**

Table 12: Results on the NYUv2 dataset in the 50% data setting *with different training strategies for the refinement network*. 'MT': multi-task learning baseline; 'PEoE': plain end-to-end training; 'LSeT': loosely separate training; 'EML': EM-like training (proposed in the main paper). Our EML significantly outperforms alternative strategies to train the refinement network.

Model	SS-mIOU (↑)	DE-mABSE (↓)	SN-mAD (↓)
ST	0.239	0.849	0.282
MT	0.244	0.834	0.313
MGM	**0.257**	**0.819**	**0.275**

Table 13: Experiments with 256 image resolution on the NYUv2 dataset. Our MGM still consistently outperforms the compared baselines, showing the great robustness and flexibility of the proposed framework.

results on the NYUv2 dataset in the 50% data setting. From this table, we could find: (1) A *naïve* end-to-end training strategy is not able to facilitate the cooperation between different networks and thus hurts the overall performance; (2) Loosely separate training enables the communication between the refinement network and the main network *when needed* and thus outperforms the baseline; (3) Our proposed EM-like training strategy further improves over the loosely separate training and achieves the best performance.

C.4. Experiments on Higher Image Resolution

In principle, the proposed framework is agnostic to the specific types of multi-task networks and image generation networks, thus flexible with image resolutions. The practical constraint lies in that it is still challenging and resource-consuming for modern generative models to synthesize very high-resolution images (Brock et al., 2019; Zhang et al., 2019), although the deep multi-task models normally work better with high-resolution images. In the main experiments, consistent with exiting image synthesis work (Zhang et al., 2019), we focused on the resolution of 128×128. Here we further made an attempt to run our experiments with a higher resolution, 256×256 on NYUv2. The results of all the compared models in the 50% data setting are shown in Table 13. We could find that MGM still consistently outperforms the baselines, indicating the great robustness and flexibility of our proposed framework.

C.5. Generalization of the Shared Feature Representation

Intuitively, our MGM achieves state-of-the-art performance by effectively learning a shared feature representation. We further show the generalization capability of this representation by designing the following experiment: for the multi-task model and our MGM model, we first learn the shared feature space with the SS and DE tasks, and we then use that learned feature space to train a new decoder for the SN task. We report the results on NYUv2 in Table 14. Our MGM outperforms the multi-task model in all the three data settings, which means that MGM indeed learns a better and robust shared feature space.

C.6. Full Impact with Generative Networks and Joint Training

In Table 15, we show the full results for $\text{MGM}_{/\text{G}}$ and $\text{MGM}_{/\text{j}}$ on the two datasets. 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.

D. Implementation Details

Data Processing: For the NYUv2 (Nathan Silberman & Fergus, 2012) dataset, following Sun et al. (2020) we resize and normalize the RGB images to $(-1, 1)$, standardize the normal ground-truth to $(0, 1)$, and do not normalize the depth. For the Taskonomy dataset (Zamir et al., 2018), we follow the standard data normalization in Zamir et al. (2018).

Model	mAD-100% (↓)	mAD-50% (↓)	mAD-25% (↓)
MT	0.291	0.310	0.323
MGM	**0.280**	**0.298**	**0.305**

Table 14: Results for the SN task with pre-trained feature representations by the SS and DE tasks. MGM consistently outperforms multi-task (MT), indicating that MGM learns a more effective and generalizable feature representation.

	Data Setting	100% Data Setting			50% Data Setting			25% Data Setting		
	Models	$\text{MGM}_{/\text{G}}$	$\text{MGM}_{/\text{j}}$	MGM	$\text{MGM}_{/\text{G}}$	$\text{MGM}_{/\text{j}}$	MGM	$\text{MGM}_{/\text{G}}$	$\text{MGM}_{/\text{j}}$	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 Task-onomy	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 15: Full Comparison of our MGM model with its variants. $\text{MGM}_{/\text{G}}$: *without* generating synthesized images; $\text{MGM}_{/\text{j}}$: *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.

Additional Implementation Details: We use Adam (Kingma & Ba, 2014) optimizer for all the models. The learning rates are set to 0.001 for the multi-task, self-supervision, and refinement networks, 0.0001 for the SAGAN generator, and 0.0004 for the SAGAN discriminator. The batch size is set to 32. We use a cross-entropy loss for semantic segmentation and the scene classification task of the refinement network, and an l_1 loss for surface normal and depth estimation.

Due to the different converge time for the different modules, we use a three-stage strategy to perform joint training: (1) We first train the multi-task network *separately* with fully labeled real data; (2) We then freeze the multi-task network, and train the image generation network and the embedding network *separately*; (3) We do joint training with *the whole* network using both fully labeled real data and weakly labeled sythesized data. During the pre-training process of SAGAN, we set the batch size to 128 to train a better model following Zhang et al. (2019). Then, for the joint training, we use a batch size of 32 for all the sub-networks. Additionally, for the same minibatch of data, we update the image generation network 2 times iteratively during stage (3).

Training Procedure: We summarized the training procedure in Algorithm 1 in the main paper. 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 $\hat{\boldsymbol{y}} = \mathbf{M}(x)$, and then use the task-specific losses between $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$ to update the multi-task network $\mathbf{M}$.
2. We predict the scene labels by $\hat{c} = \mathbf{R}(\hat{\boldsymbol{y}})$, 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 $(\widetilde{x}, \widetilde{c})$, and use these data to update $\mathbf{E}$ by performing both the EM-like algorithm described in Sec. 3.2 (main paper) with $\mathbf{R}$ and the self-supervised learning as in step 3.

Dataset Setting We evaluate all the models on two widely-benchmarked datasets: **NYUv2** (Nathan Silberman & Fergus, 2012; Eigen & Fergus, 2015) containing 1,449 images with 40 types of objects (Gupta et al., 2013); **Tiny-Taskonomy** which is the standard tiny split of the Taskonomy dataset (Zamir et al., 2018).

For Tiny-Taskonomy dataset, 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.

E. More Visualizations

In Figure 4 of the main paper, we visualized the prediction results. Here we provide more visualizations of the multi-task predictions for MGM and the compared baselines in Figure 8. Notice that, for the surface normal predictions, the ground-truth of NYU-V2 dataset has padded boundaries. For a better visualization, we mask the boundaries in the main paper but include them in Figure 8. Our MGM model significantly outperforms both ST and MT baselines.

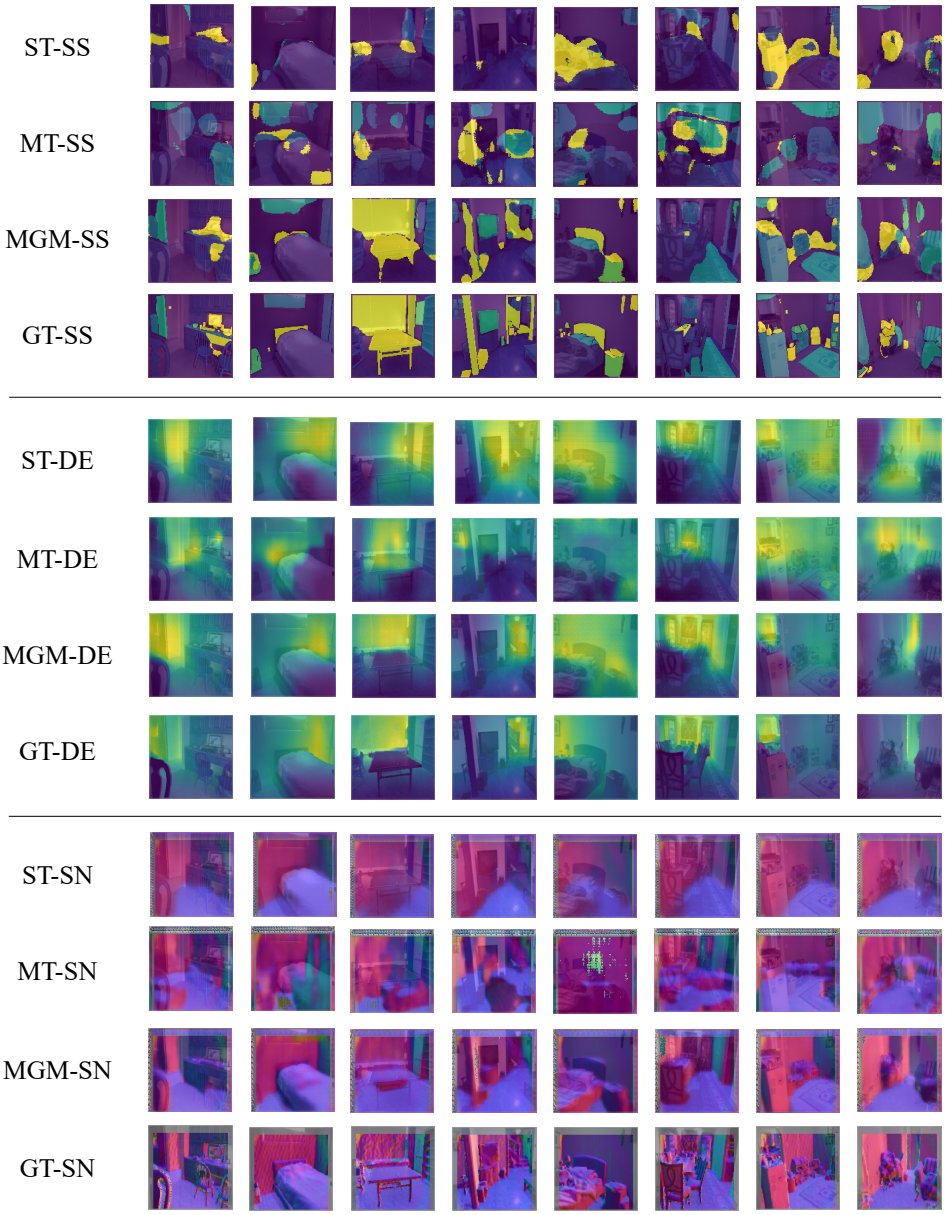


Figure 8: More visualizations of the multi-task predictions for MGM and the compared baselines. SS: semantic segmentation task; DE: depth estimation task; SN: surface normal prediction task; ST: single-task model; MT: multi-task model; MGM: multi-task oriented generative modeling (our proposed model); GT: ground-truth. The prediction results of our MGM model are much closer to the ground-truth and significantly outperform the state-of-the-art results.