Generative Semantic Segmentation

Jiaqi Chen[1] Jiachen Lu[1] Xiatian Zhu[2] Li Zhang[1*]
[1]Fudan University [2]University of Surrey
https://github.com/fudan-zvg/GSS

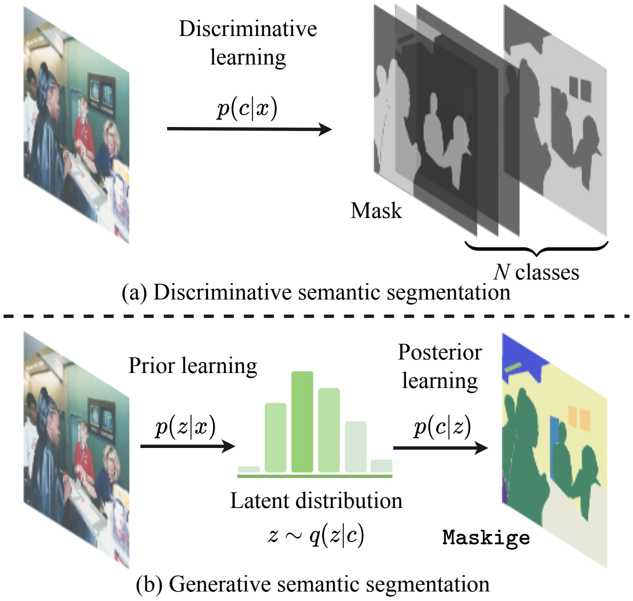
Abstract

*We present **Generative Semantic Segmentation** (GSS), a generative learning approach for semantic segmentation. Uniquely, we cast semantic segmentation as an **image-conditioned mask generation problem**. This is achieved by replacing the conventional per-pixel discriminative learning with a latent prior learning process. Specifically, we model the variational posterior distribution of latent variables given the segmentation mask. To that end, the segmentation mask is expressed with a special type of image (dubbed as* maskige*). This posterior distribution allows to generate segmentation masks unconditionally. To achieve semantic segmentation on a given image, we further introduce a conditioning network. It is optimized by minimizing the divergence between the posterior distribution of maskige (i.e. segmentation masks) and the latent prior distribution of input training images. Extensive experiments on standard benchmarks show that our GSS can perform competitively to prior art alternatives in the standard semantic segmentation setting, whilst achieving a new state of the art in the more challenging cross-domain setting.*

1. Introduction

The objective of semantic segmentation is to predict a label for every single pixel of an input image [32]. Conditioning on each pixel's observation, existing segmentation methods [4,9,50,56] naturally adopt the *discriminative learning* paradigm, along with dedicated efforts on integrating task prior knowledge (*e.g.*, spatial correlation) [9, 23, 46, 56]. For example, existing methods [4, 50, 56] typically use a linear projection to optimize the log-likelihood classification for each pixel. Despite the claim of subverting per-pixel classification, the bipartite matching-based semantic segmentation [8, 9] still cannot avoid the per-pixel max log-likelihood.

In this paper, we introduce a new approach, ***Generative Semantic Segmentation*** (GSS), that formulates semantic segmentation as *an image-conditioned mask generation problem*. This conceptually differs from the conventional formulation of discriminative per-pixel classification learning, based on the log-likelihood of a conditional probability (*i.e.* the classification probability of image pixels). Taking the manner of image generation instead [24, 44], we generate the *whole* segmentation masks with *an auxiliary latent variable distribution* introduced. This formulation is not only simple and more task-agnostic, but also facilitates the exploitation of off-the-shelf big generative models (*e.g.* DALL·E [39] trained by 3 billion iterations on a 300 million open-image dataset, far beyond both the data scale and training cost of semantic segmentation).



(a) Discriminative semantic segmentation

(b) Generative semantic segmentation

Figure 1. Schematic comparison between (**a**) conventional discriminative learning and (**b**) our generative learning based model for semantic segmentation. Our GSS introduces a latent variable z and, given the segmentation mask c, it learns the posterior distribution of z subject to the reconstruction constraint. Then, we train a conditioning network to model the prior of z by aligning with the corresponding posterior distribution. This formulation can thus generate the segmentation mask for an input image.

[*] Li Zhang (lizhangfd@fudan.edu.cn) is the corresponding author with School of Data Science, Fudan University.

However, achieving segmentation segmentation in a generic generation framework (*e.g.* the Transformer architecture [15]) is non-trivial due to drastically different data format. To address this obstacle, we propose a notion of `maskige` that expresses the segmentation mask in the RGB image form. This enables the use of a pretrained latent posterior distribution (*e.g.* VQVAE [39]) of existing generative models. Our model takes a two-stage optimization: **(i)** Learning the posterior distribution of the latent variables conditioned on the semantic segmentation masks so that the latent variables can simulate the target segmentation masks; To achieve this, we introduce an fixed pre-trained VQVAE [39] and a couple of lightweight transformation modules, which can be trained with minimal cost, or they can be manually set up without requiring any additional training. In either case, the process is efficient and does not add significant overhead to the overall optimization. **(ii)** Minimizing the distance between the posterior distribution and the prior distribution of the latent variables given input training images and their masks, enabling to condition the generation of semantic masks on the input images. This can be realized by a generic encoder-decoder style architecture (*e.g.* a Transformer).

We summarize the *contributions* as follows. **(i)** We propose a ***Generative Semantic Segmentation*** approach that reformulates semantic segmentation as an image-conditioned mask generation problem. This represents a *conceptual shift* from conventional discriminative learning based paradigm. **(ii)** We realize a GSS model in an established conditional image generation framework, with minimal need for task-specific architecture and loss function modifications while fully leveraging the knowledge of off-the-shelf generative models. **(iii)** Extensive experiments on several semantic segmentation benchmarks show that our GSS is competitive with prior art models in the standard setting, whilst achieving a new state of the art in the more challenging and practical cross-domain setting (*e.g.* MSeg [26]).

2. Related work

Semantic segmentation Since the inception of FCN [32], semantic segmentation have flourished by various deep neural networks with ability to classify each pixel. The follow-up efforts then shift to improve the limited receptive field of these models. For example, PSPNet [55] and DeepLabV2 [3] aggregate multi-scale context between convolution layers. Sequentially, Nonlocal [47], CCNet [21], and DGMN [54] integrate the attention mechanism in the convolution structure. Later on, Transformer-based methods (*e.g.* SETR [56] and Segformer [50]) are proposed following the introduction of Vision Transformers. More recently, MaskFormer [9] and Mask2Former [8] realize semantic segmentation with bipartite matching. Commonly, all the methods adopt the discriminative pixel-wise classification learning paradigm. This is in contrast to our generative semantic segmentation.

Image generation In parallel, generative models [15, 39] also excel. They are often optimized in a two-stage training process: (1) Learning data representation in the first stage and (2) building a probabilistic model of the encoding in the second stage. For learning data representation, VAE [24] reformulates the autoencoder by variational inference. GAN [19] plays a zero-sum game. VQVAE [44] extends the image representation learning to discrete spaces, making it possible for language-image cross-model generation. [27] replaces element-wise errors of VAE with feature-wise errors to capture data distribution. For probabilistic model learning, some works [14, 40, 49] use flow for joint probability learning. Leveraging the Transformers to model the composition between condition and images, Esser et al. [15] demonstrate the significance of data representation (*i.e.* the first stage result) for the challenging high-resolution image synthesis, obtained at high computational cost. This result is inspiring to this work in the sense that the diverse and rich knowledge about data representation achieved in the first stage could be transferable across more tasks such as semantic segmentation.

Generative models for visual perception Image-to-image translation made one of the earliest attempts in generative segmentation, with far less success in performance [22]. Some good results were achieved in limited scenarios such as face parts segmentation and Chest X-ray segmentation [28]. Replacing the discriminative classifier with a generative Gaussian Mixture model, GMMSeg [29] is claimed as generative segmentation, but the most is still of discriminative modeling. The promising performance of Pix2Seq [7] on several vision tasks leads to the prevalence of sequence-to-sequence task-agnostic vision frameworks. For example, Unified-I/O [33] supports a variety of vision tasks within a single model by seqentializing each task to sentences. Pix2Seq-D [6] deploys a hierarchical VAE (*i.e.* diffusion model) to generate panoptic segmentation masks. This method is inefficient due to the need for iterative denoising. UViM [25] realizes its generative panoptic segmentation by introducing latent variable conditioned on input images. It is also computationally heavy due to the need for model training from scratch. To address these issues, we introduce a notion of `maskige` for expressing segmentation masks in the form of RGB images, enabling the adopt of off-the-shelf data representation models (*e.g.* VGVAE) already pretrained on vast diverse imagery. This finally allows for generative segmentation model training as efficiently as conventional discriminative counterparts.

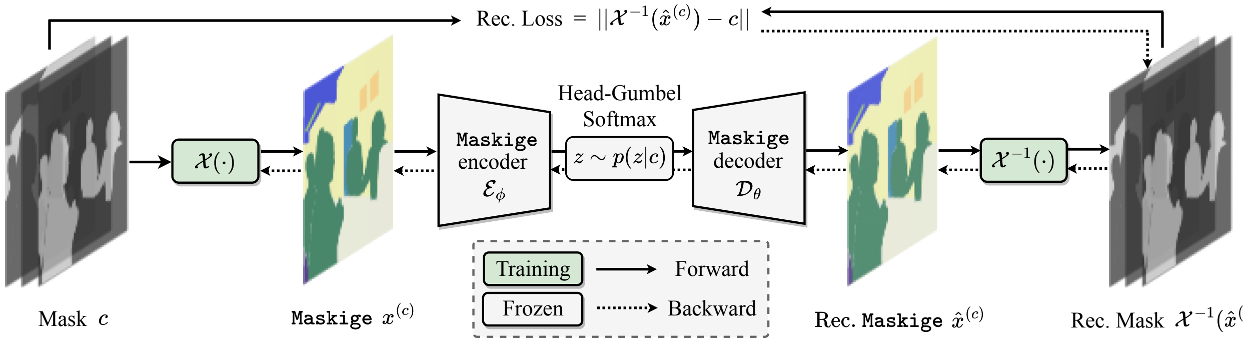

Rec. Loss = $||\mathcal{X}^{-1}(\hat{x}^{(c)}) - c||$
Head-Gumbel Softmax
$\mathcal{X}(\cdot)$
Maskige encoder $\mathcal{E}_\phi$
$z \sim p(z|c)$
Maskige decoder $\mathcal{D}_\theta$
$\mathcal{X}^{-1}(\cdot)$
Training
Forward
Frozen
Backward
Mask c
Maskige $x^{(c)}$
Rec. Maskige $\hat{x}^{(c)}$
Rec. Mask $\mathcal{X}^{-1}(\hat{x}^{(c)})$

Figure 2. **An illustration of our efficient latent posterior learning.** Instead of training a maskige encoder-decoder, we utilize a well pretrained VQVAE [44] (gray blocks) and optimize the transformations $\mathcal{X}$ and $\mathcal{X}^{-1}$ (green blocks). To optimize $\mathcal{X}$ and $\mathcal{X}^{-1}$ with gradient descent, we employ the Gumbel softmax relaxation technique [34]. "Rec.": Reconstructed.

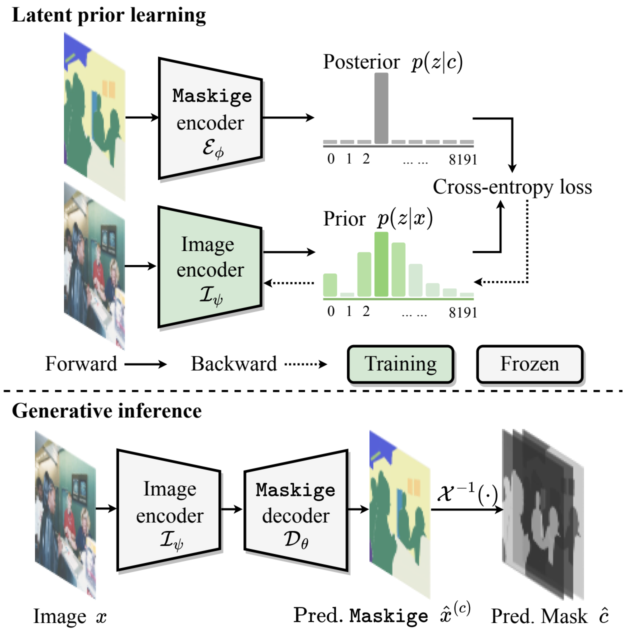

Latent prior learning
Maskige encoder $\mathcal{E}_\phi$
Posterior $p(z|c)$
0 1 2 8191
Cross-entropy loss
Image encoder $\mathcal{I}_\psi$
Prior $p(z|x)$
0 1 2 8191
Forward
Backward
Training
Frozen
Generative inference
Image encoder $\mathcal{I}_\psi$
Maskige decoder $\mathcal{D}_\theta$
$\mathcal{X}^{-1}(\cdot)$
Image x
Pred. Maskige $\hat{x}^{(c)}$
Pred. Mask $\hat{c}$

Figure 3. **Illustration of latent prior learning (*top*) and generative inference pipeline (*bottom*).** During training, for latent prior learning, we optimize the image encoder $\mathcal{I}_\psi$ while freezing the maskige encoder $\mathcal{E}_\phi$. The objective is to minimize the divergence (*e.g.* cross entropy loss) between the prior distribution and the posterior distribution of latent tokens. During generative inference, we use the prior $z \sim p(z|x)$ inferred by $\mathcal{I}_\psi$ to generate the maskige with maskige decoder $\mathcal{D}_\theta$. "Pred.": Predicted.

3. Methodology

3.1. GSS formulation

Traditionally, semantic segmentation is formulated as a discriminative learning problem as

$$\max_{\pi} \log p_\pi(c|x) \tag{1}$$

where $x \in \mathbb{R}^{H\times W\times 3}$ is an input image, $c \in \{0,1\}^{H\times W\times K}$ is a *segmentation mask* in K semantic categories, and p_π is a discriminative pixel classifier. Focusing on learning the classification boundary of input pixels, this approach enjoys high data and training efficiency [37].

In this work, we introduce ***Generative Semantic Segmentation*** (GSS) by introducing a discrete L-dimension latent distribution $q_\phi(z|c)$ (with $z \in \mathbb{Z}^L$) to the above log-likelihood as:

$$\log p(c|x) \geq \mathbb{E}_{q_\phi(z|c)}\left[\log \frac{p(z,c|x)}{q_\phi(z|c)}\right],$$

which is known as the Evidence Lower Bound (ELBO) [24] (details are given in the supplementary material). Expanding the ELBO gives us

$$\mathbb{E}_{q_\phi(z|c)}\left[\log p_\theta(c|z)\right] - D_{KL}\Big(q_\phi(z|c), p_\psi(z|x)\Big), \tag{2}$$

where we have three components in our formulation:

- p_ψ: An **image encoder** (denoted as $\mathcal{I}_\psi$) that models *the prior distribution* of latent tokens z conditioned on the input image x.
- q_ϕ: A function that encodes the semantic segmentation mask c into discrete latent tokens z, which includes a **maskige encoder** (denoted as $\mathcal{E}_\phi$, implemented by a VQVAE encoder [44]) and a linear projection (denoted as $\mathcal{X}$, which will be detailed in Section 3.3).
- p_θ: A function that decodes the semantic segmentation mask c from the discrete latent tokens z, which includes a **maskige decoder** (denoted $\mathcal{D}_\theta$, implemented by a VQVAE decoder [44]) and $\mathcal{X}^{-1}$ (the inverse process of $\mathcal{X}$).

Architecture The architecture of GSS comprises three components: $\mathcal{I}_\psi$, $\mathcal{E}_\phi$ and $\mathcal{D}_\theta$. $\mathcal{E}\phi$ and $\mathcal{D}_\theta$ are implemented as VQVAE encoder and decoder [44], respectively. Meanwhile, $\mathcal{I}_\psi$ is composed of an image backbone R(*e.g.* esNet [20] or Swin Transformer [31]) and a *Multi-Level Aggregation* (MLA). As an essential part, MLA is constructed using D shifted window Transformer layers [31] and a linear projection layer. The resulting output is a discrete code $z \in \mathbb{Z}^{H/d\times W/d}$, where d denotes the downsample ratio.

Optimization Compared to the log-likelihood in discriminative models, optimizing the ELBO of a general model is more challenging [37]. End-to-end training cannot reach a global optimization. For Eq. (2), often we name the first term $\mathbb{E}_{q_\phi(z|c)}[\log p_\theta(c|z)]$ as a reconstruction term and the second KL-divergence as the prior term. In the next section we will introduce the optimization of this ELBO.

3.2. ELBO optimization for semantic segmentation

The ELBO optimization process for semantic segmentation involves two main steps, as described in [39]. The first step is ***latent posterior learning***, also known as reconstruction (see Figure 2). Here, the ELBO is optimized with respect to θ and ϕ through training a VQVAE [44] to reconstruct the desired segmentation masks. The second step is ***latent prior learning*** (see Figure 3). Once θ and ϕ are fixed, an image encoder ψ is optimized to learn the prior distribution of latent tokens given an input image.

Typically, the first stage of ELBO optimization is both most important and most expensive (much more than many discriminative learning counterparts) [15]. To address this challenge, we propose an efficient latent posterior learning process.

3.3. Stage I: Efficient latent posterior learning

During the first stage (shown in Figure 2), the initial prior $p_\psi(z|x)$ is set as the uniform distribution. Conventionally, the first stage latent posterior training is conducted by

$$\min_{\theta,\phi} \mathbb{E}_{q_\phi(z|c)} \|p_\theta(c|z) - c\|. \tag{3}$$

To optimize Eq. (3) more efficiently, we introduce a random variable transformation $\mathcal{X} : \mathbb{R}^K \to \mathbb{R}^3$, its pseudo-inverse function $\mathcal{X}^{-1} : \mathbb{R}^3 \to \mathbb{R}^K$, and the `maskige` decoder $\mathcal{D}_\theta$. Applying expectation with transformed random variable, we have

$$\begin{aligned} &\min_{\hat\phi,\hat\theta} \mathbb{E}_{q_{\hat\phi}(\hat z|\mathcal{X}(c))} \|\mathcal{D}_{\hat\theta}(\hat z) - \mathcal{X}(c)\| \\ &+ \min_{\mathcal{X}^{-1}} \mathbb{E}_{q_{\hat\phi}(\hat z|\mathcal{X}(c))} \|\mathcal{X}^{-1}(\mathcal{D}_{\hat\theta}(\hat z)) - c\|. \end{aligned} \tag{4}$$

Please refer to supplementary material for more details. Then we find that $\mathcal{X}(c) = x^{(c)} \in \mathbb{R}^{H\times W\times 3}$ can be regarded as a kind of RGB image, where each category is represented by a specific color. For convenience, we term it `maskige`. Therefore, the optimization can be rewritten as

$$\begin{aligned} &\min_{\hat\phi,\hat\theta} \mathbb{E}_{q_{\hat\phi}(\hat z|x^c)} \|\mathcal{D}_{\hat\theta}(\hat z) - x^{(c)}\| \\ &+ \min_{\mathcal{X}^{-1}} \mathbb{E}_{q_{\hat\phi}(\hat z|\mathcal{X}(c))} \|\mathcal{X}^{-1}(\hat x^{(c)}) - c\|, \end{aligned} \tag{5}$$

where $\hat x^{(c)} = \mathcal{D}_{\hat\theta}(\hat z)$. Now, the first term of Eq. (5) can be regarded as an image reconstruction task (see Figure 2). In practice, this has been already well optimized by [15, 39] using million-scale datasets. This allows us to directly utilize the off-the-shelf pretrained models. As such, our optimization problem can be simplified as:

$$\min_{\mathcal{X}^{-1}} \mathbb{E}_{q_{\hat\phi}(\hat z|\mathcal{X}(c))} \|\mathcal{X}^{-1}(\hat x^{(c)}) - c\|. \tag{6}$$

Note that the parameters (0.9K∼466.7K in our designs) of $\mathcal{X}$ and $\mathcal{X}^{-1}$ are far less than θ, ϕ (totally 29.1M parameters with the VQVAE from [39]), thus more efficient and cheaper to train. Concretely, we only optimize the small $\mathcal{X}$ while freezing θ, ϕ. Following [6, 25], we use cross-entropy loss instead of MSE loss in Eq. (6) for a better minimization between segmentation masks.

Linear maskige designs The optimization problem for $\mathcal{X}$ and $\mathcal{X}^{-1}$ is non-convex, making their joint optimization challenging. To overcome this issue, we optimize $\mathcal{X}$ and $\mathcal{X}^{-1}$ separately. For simplicity, we model both $\mathcal{X}$ and $\mathcal{X}^{-1}$ as linear functions (*linear assumption*). Specifically, we set $x^{(c)} = c\beta$ where $\beta \in \mathbb{R}^{K\times 3}$, and $\hat c = \hat x^{(c)}\beta^\dagger$ where $\beta^\dagger \in \mathbb{R}^{3\times K}$. Under the linear assumption, we can transfer Eq. (6) into a least squares problem with an explicit solution $\beta^\dagger = \beta^\top(\beta\beta^\top)^{-1}$, so that $\mathcal{X}^{-1}$ is free of training.

To enable zero-cost training of $\mathcal{X}$, we can also manually set the value of β properly. We suggest a ***maximal distance assumption*** for selecting the value of β to encourage the encoding of K categories to be as widely dispersed as possible in the three-dimensional Euclidean space $\mathbb{R}^3$. More details are provided in the supplementary material.

Non-linear maskige designs For more generic design, non-linear models (*e.g.* CNNs or Transformers) can be also used to express $\mathcal{X}^{-1}$. (*non-linear assumption*)

Concrete maskige designs We implement four optimization settings for $\mathcal{X}$ and/or $\mathcal{X}^{-1}$ with varying training budgets. We define the naming convention of "GSS-[F/T] [F/T] (-*O*)" in the following rules. ***(i)*** *Basic settings* "-[F/T] [F/T]" on whether $\mathcal{X}$ and/or $\mathcal{X}^{-1}$ require training: "F" stands for *F*ree of training, and "T" for *T*raining required. ***(ii)*** *Optional settings* "-O" (*e.g.* "R" or "W") which will be explained later on. All GSS variants are described below.

- **GSS-FF** (training free): Modeling both $\mathcal{X}$ and $\mathcal{X}^{-1}$ using linear functions, with β initialized under *maximal distance assumption*, and $\beta^\dagger$ optimized using least squares. For comparison, we will experiment with **GSS-FF-R**, where β is **R**andomly initialized.
- **GSS-FT** (training required): Modeling $\mathcal{X}$ using a linear function but modeling $\mathcal{X}^{-1}$ using a non-linear function (*e.g.* a three-layer convolutional neural network). We initialize β with *maximal distance assumption* and optimize $\mathcal{X}^{-1}$ with gradient descent. A stronger design **GSS-FT-W** utilizes a single-layer Shifted **W**indow Transformer block [31] as the non-linear $\mathcal{X}^{-1}$. Notably, we train $\mathcal{X}$ and $\mathcal{X}^{-1}$ separately, as described in Section 4.1.

- **GSS-TF** (training required): Modeling both $\mathcal{X}$ and $\mathcal{X}^{-1}$ using linear functions. We train β with gradient descent and optimize $\beta^{\dagger}$ with least squares according to β.
- **GSS-TT** (training required): Modeling $\mathcal{X}$ using a linear function but modeling $\mathcal{X}^{-1}$ using a non-linear function (*e.g.* a three-layer CNN). We jointly train both functions using gradient descent.

To perform end-to-end optimization of the $\mathcal{X}$ function using gradient descent for both GSS-TF&TT, a hard Gumbel-softmax relaxation technique [34] is used. This involves computing the `argmax` operation during the forward step, while broadcasting the gradients during the backward step. Our *linear* designs (*i.e.* GSS-FF&FF-R) is training free with zero cost. Our non-linear assumption based designs (*e.g.* GSS-FT&FT-W) has high performance potential at acceptable training cost (see Section 4.2).

3.4. Stage II: Latent prior learning

We show in Figure 3 (***Top***) the latent prior learning. In this stage, we learn the prior joint distribution between mask latent representation z and images x, with ϕ, θ both fixed.

Objective The optimization target of this stage is the second term of Eq. (2):

$$\min_{\psi} D_{KL}\Big(q_{\phi}(z|c), p_{\psi}(z|x)\Big),$$

where z is in a discrete space of codebook-sized (*e.g.* 8192 in [39]) integers. The objective is to minimize the distance between the discrete distribution of z predicted by latent prior encoder p_{ψ} and the z given by VQVAE. Since the entropy of q_{ϕ} is fixed (*i.e.*the ground truth), we can use the cross-entropy function to measure their alignment.

Unlabeled area auxiliary Due to high labeling cost and challenge, it is often the case that a fraction of areas per image are unlabeled (*i.e.* unknown/missing labels). Modeling per-pixel conditional probability $p(c|x)$ in existing discriminative models, this issue can be simply tackled by ignoring all unlabeled pixels during training.

In contrast, generative models (*e.g.* UViM [25] and our GSS) are trained at the latent token level, without flexible access to individual pixels. As a result, unlabeled pixels bring about extra challenges, as they can be of objects/stuff of any categories heterogeneously. Without proper handling, a generative model may learn to classify difficult pixels as the unlabelled and hurting the final performance (see Figure 4).

To address this problem, we exploit a pseudo labeling strategy. The idea is to predict a label for each unlabeled pixel. Specifically, we further introduce an auxiliary head $p_{\xi}(\bar{c}|z)$ during latent prior learning (*i.e.* state II) to label all unlabeled areas. Formally, we form an enhanced ground-truth mask by $\tilde{c} = M_u \cdot \bar{c} + (1 - M_u) \cdot c$ where M_u masks out labeled pixels, $\bar{c}$ denotes the pseudo labels, and $\tilde{c}$ denotes the labels after composition. The training objective of this stage cane be then revised as:

$$\min_{\psi} D_{KL}\left(q_{\phi}(z|c), p_{\psi}(z|x)\right) + p_{\xi}(\bar{c}|z). \qquad (7)$$

3.5. Generative inference

As illustrated in Figure 3 ***(bottom)***, we first take the latent tokens z that are predicted by the image encoder $\mathcal{I}_{\psi}$, and feed them into the `maskige` decoder $\mathcal{D}_{\theta}$ to generate the predicted `maskige` $\hat{x}^{(c)}$. Next, we apply the inverse transformation $\mathcal{X}^{-1}$ (Section 3.3) to the predicted `maskige` to obtain the final segmentation mask $\hat{c}$.

4. Experiment

4.1. Experimental setup

Cityscapes [11] provides pixel-level annotations for 19 object categories in urban scene images at a high resolution of 2048×1024. It contains 5000 finely annotated images, split into 2975, 500 and 1525 images for training, validation and testing respectively.

ADE20K [57] is a challenging benchmark for scene parsing with 150 fine-grained semantic categories. It has 20210, 2000 and 3352 images for training, validation and testing.

MSeg [26] is a composite dataset that unifies multiple semantic segmentation datasets from different domains. In particular, the taxonomy and pixel-level annotations are aligned by relabeling more than 220,000 object masks in over 80,000 images. We follow the standard setting: the train split [2, 3, 11, 30, 38, 42, 45, 51] for training a unified semantic segmentation model, the test split (unseen to model training) [1, 12, 16, 18, 36, 53] for cross-domain validation.

Evaluation metrics The mean Intersection over Union (mIoU) and pixel-level accuracy (mAcc) are reported for all categories, following the standard evaluation protocol [11].

Implementation details We operate all experiments on *mmsegmentation* [10] with 8 NVIDIA A6000 cores. *(i) Data augmentation*: Images are resized to 1024×2048 on Cityscapes, 512×2048 on ADE20K and MSeg, and random cropped (768×768 on cityscapes and 512×512 on ADE20K and MSeg) and random horizontal flipped during training. **No test time augmentation** is applied. *(ii) Training schedule for latent prior learning*: The batch size is 16 on Cityscapes and MSeg and 32 on ADE20K. The total number of iterations is 80,000 on Cityscapes and 160,000 on ADE20K and MSeg.

4.2. Ablation studies

Latent posterior learning We evaluate the variants of latent posterior learning as described in Section 3.3. We observe from Table 1 that: **(i)** GSS-FF comes with no extra training

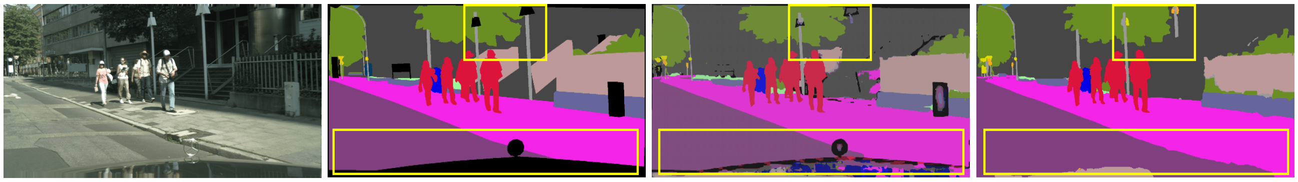


Image | Ground Truth | without auxiliary | with auxiliary

Figure 4. Qualitative results of *unlabeled area auxiliary* on Cityscapes [11] dataset.

GSS variants	mIoU	Training time
GSS-FF-R	62.83	**0**
GSS-FF	84.31	**0**
GSS-FT	86.10	$\leq$20
GSS-TF	84.37	$\leq$5
GSS-TT	36.11	$\leq$5
GSS-FT-W	**87.73**	$\leq$350

Table 1. **Ablation on the variants of latent posterior learning** on the `val` set of ADE20K. Metrics: The `maskige` reconstruction performance in mIoU, as well as the training time in `GPU hours` (*i.e.*, effective single-core hours).

Design	`Maskige`?	Cityscapes	ADE20K	Train time
VQGAN [15]	✗	82.16	81.89	$\leq$500
VQGAN [15]	✓	75.09	42.70	$\leq$100
UViM [25]	✗	89.14	78.98	$\leq$2,000
DALL·E [39]	✓	**95.17**	**87.73**	$\leq$350

Table 2. **Ablation on `maskige` reconstruction by different VQVAE designs** on Cityscapes semantic `val` split and ADE20k `val` split. In case of *no `maskige`*, we directly reconstruct the segmentation mask with K (the number of classes) channels. Unit for *training time* is `GPU hour` (*i.e.* the effective single-core hour).

d	Unlabel	MLA	mIoU	mAcc
1/8			40.64	52.55
1/8	✓		43.72	56.08
1/4	✓		43.98	56.11
1/4	✓	✓	**46.29**	**57.84**

Table 3. **Ablation on latent prior learning** on the `val` split of ADE20K. "Unlabel" denotes unlabeled area auxiliary, and "MLA" denotes Multi-Level Aggregation. "d" is the downsample ratio of discrete mask representation size between input image size.

cost. Whilst UViM consumes nearly 2K GPU hours for training a VQVAE with similar performance achieved [25]. **(ii)** The randomly initialized β (*i.e.* GSS-FF-R) leads to considerable degradation on the least square optimization. **(iii)** With our *maximal distance assumption*, the regularized β of GSS-FF brings clear improvement, suggesting the significance of initialization and its efficacy of our strategy. **(iv)** With three-layer conv network with activation function for $\mathcal{X}^{-1}$, GSS-FT achieves good reconstruction. Further equipping with a two-layer Shifted Window Transformer block [31] for $\mathcal{X}^{-1}$ (*i.e.* GSS-FT-W) leads to the best result at a cost of extra 329.5 GPU hours. This is due to more accurate trans-

Method	Pretrain	Backbone	Iteration	mIoU
- Discriminative modeling:				
FCN [32]	1K	ResNet-101	80k	77.02
PSPNet [55]	1K	ResNet-101	80k	79.77
DeepLab-v3+ [5]	1K	ResNet-101	80k	80.65
NonLocal [47]	1K	ResNet-101	80k	79.40
CCNet [21]	1K	ResNet-101	80k	79.45
Maskformer [9]	1K	ResNet-101	90k	78.50
Mask2former [8]	1K	ResNet-101	90k	80.10
SETR [56]	22K	ViT-Large	80k	78.10
UperNet [48]	22K	Swin-Large	80k	82.89
Maskformer [9]	22K	Swin-Large	90k	78.50
Mask2former [8]	22K	Swin-Large	90k	**83.30**
SegFormer [50]	1K	MiT-B5	160k	82.25
- Generative modeling:				
UViM† [25]	22K	Swin-Large	160k	70.77
GSS-FF (Ours)	1K	ResNet-101	80k	77.76
GSS-FT-W (Ours)	1K	ResNet-101	80k	78.46
GSS-FF (Ours)	22K	Swin-Large	80k	78.90
GSS-FT-W (Ours)	22K	Swin-Large	80k	**80.05**

Table 4. **Performance comparison on the Cityscapes `val` split:** UViM† [25] is reproduced by us on PyTorch. "1K" means pretrained on ImageNet 1K [13] while "22K" means pretrained on ImageNet 22K [13].

Method	Pretrain	Backbone	Iteration	mIoU
- Discriminative modeling:				
FCN [32]	1K	ResNet-101	160k	41.40
CCNet [21]	1K	ResNet-101	160k	43.71
DANet [17]	1K	ResNet-101	160k	44.17
UperNet [48]	1K	ResNet-101	160k	43.82
Deeplab V3+ [5]	1K	ResNet-101	160k	45.47
Maskformer [9]	1K	ResNet-101	160k	45.50
Mask2former [8]	1K	ResNet-101	160k	47.80
OCRNet [52]	1K	HRNet-W48	160k	43.25
SegFormer [50]	1K	MiT-B5	160k	**50.08**
SETR [56]	22K	ViT-Large	160k	48.28
- Generative modeling:				
UViM† [25]	22k	Swin-Large	160k	43.71
GSS-FF (Ours)	22K	Swin-Large	160k	46.29
GSS-FT-W (Ours)	22K	Swin-Large	160k	**48.54**

Table 5. **Performance comparison with previous art methods on the ADE20K `val` split**. UViM† [25] is reproduced by ourselves.

Method	Backbone	Iteration	VOC [16]	Context [36]	CamVid [1]	WildDash [53]	KITTI [18]	ScanNet [12]	*h. mean*
- *Discriminative modeling:*									
CCSA [35]	HRNet-W48	500k	48.9	-	52.4	36.0	-	27.0	39.7
MGDA [41]	HRNet-W48	500k	69.4	-	57.5	39.9	-	33.5	46.1
MSeg [26]	HRNet-W48	500k	70.7	42.7	**83.3**	62.0	67.0	48.2	59.2
MSeg† [26]	HRNet-W48	160k	63.8	39.6	73.9	60.9	65.1	43.5	54.9
MSeg† [26]	Swin-Large	160k	78.7	47.5	75.1	**66.1**	**68.1**	49.0	61.7
- *Generative modeling:*									
GSS-FF (Ours)	HRNet-W48	160k	64.1	37.1	72.3	59.3	62.0	40.6	52.6
GSS-FT-W (Ours)	HRNet-W48	160k	65.2	38.8	75.2	62.5	66.2	43.1	55.2
GSS-FF (Ours)	Swin-Large	160k	78.7	45.8	74.2	61.8	65.4	46.9	59.5
GSS-FT-W (Ours)	Swin-Large	160k	**79.5**	**47.7**	75.9	65.3	68.0	**49.7**	**61.9**

Table 6. **Cross-domain semantic segmentation performance on MSeg dataset `test` split.** "*h. mean*" is the harmonic mean [26]. MSeg† [26] is reproduced by us on MMSegmentation [10]

Sharing $\mathcal{I}_\psi$	Sharing `maskiage`	GSS-FF	GSS-FT-W
		78.9	**80.5**
	✓	78.0	79.5
✓	✓	76.6	78.4

Table 7. **Transferring the `maskiage` and image encoder $\mathcal{I}_\psi$ from MSeg to Cityscapes (`val` split).** Metric: mIoU.

lation from predicted `maskige` to segmentation mask. **(v)** Interestingly, with automatic β optimization, GSS-TF brings no benefit over GSS-FF. **(vi)** Further, joint optimization of both $\mathcal{X}$ and $\mathcal{X}^{-1}$ (*i.e.* GSS-TT) fails to achieve the best performance. **(vii)** In conclusion, GSS-FF is most efficient with reasonable accuracy, whilst GSS-FT-W is strongest with good efficiency.

VQVAE design We examine the effect of VQVAE in the context of `maskige`. We compare three designs: **(1)** *UViM-style* [25]: Using images as auxiliary input to reconstruct a segmentation mask in form of K-channels (K is the class number) in a ViT architecture. In this no `maskige` case, the size of segmentation mask may vary across different datasets, leading to a need for dataset-specific training. This scheme is thus more expensive in compute. **(2)** *VQGAN-style* [15]: Using a CNN model for reconstructing natural images (`maskige` needed for segmentation mask reconstruction) or K-channel segmentation masks (no `maskige` case) separately, both optimized in generative adversarial training manner with a smaller codebook. **(3)** *DALL·E-style* [39]: The one we adopt, as discussed earlier. We observe from Table 2 that: **(i)** Due to the need for dataset specific training, UViM-style is indeed more costly than the others. This issue can be well mitigated by our `maskige` with the first stage training cost compressed dramatically, as evidenced by DALL·E-style and VQGAN-style. Further, the inferiority of UViM over DALL·E suggests that our `maskige` is a favored strategy than feeding image as auxiliary input. **(ii)** In conclusion, using our `maskige` and DALL·E pretrained VQVAE yields the best performance in terms of both accuracy and efficiency.

Latent prior learning We ablate the second training stage for learning latent joint prior. The baseline is GSS-FF without the unlabeled area auxiliary and Multi-Level Aggregation (MLA, including a 2-layer Swin block [31]), under 1/8 downsample ratio. We observe from Table 3 that: **(i)** Our unlabeled area auxiliary boosts the accuracy by 3.1%, suggesting the importance of complete labeling which however is extremely costly in semantic segmentation. **(ii)** Increasing the discrete mask representation resolution is slightly useful. **(iii)** The MLA plays another important role, *e.g.* giving a gain of 2.3%.

4.3. Single-domain semantic segmentation

We compare our GSS with prior art discriminative methods and the latest generative model (UViM [25], a replicated version for semantic segmentation task). We report the results in Table 4 for Cityscapes [11] and Table 5 for ADE20K [57]. ***(i)*** *In comparison to discriminative methods*: Our GSS yields competitive performance with either Transformers (Swin) or CNNs (*e.g.* ResNet-101). For example, under the same setting, GSS matches the result of Maskformer [9]. Also, GSS-FT-W is competitive to the Transformer-based SETR [56] on both datasets. ***(ii)*** *In comparison to generative methods*: GSS-FF surpasses UViM [25] by a large margin whilst enjoying higher training efficiency. Specifically, UViM takes 1,900 TPU-v3 hours for the first training stage and 900 TPU-v3 hours for the second stage. While the first stage takes only 329.5 GPU hours with GSS-FT-W, and zero time with GSS-FF. The second stage of GSS-FF requires approximately 680 GPU hours. This achievement is due to our `maskige` mechanism for enabling the use of pretrained data representation and a series of novel designs for joint probability distribution modeling.

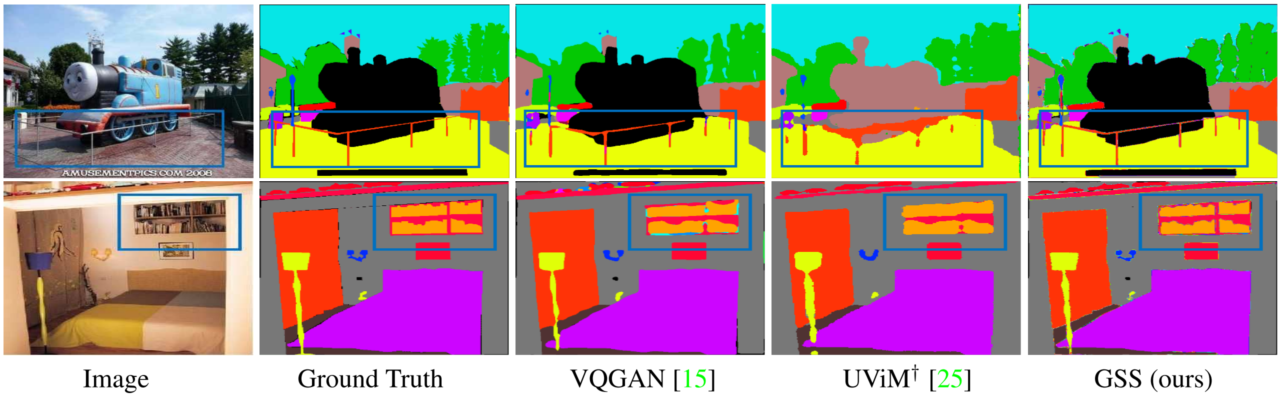


Figure 5. Qualitative results of maskige reconstruction on ADE20K [57] dataset. Note that the black areas in the Ground Truth correspond to unlabeled regions, and thus *no impact* on mIoU measurement.

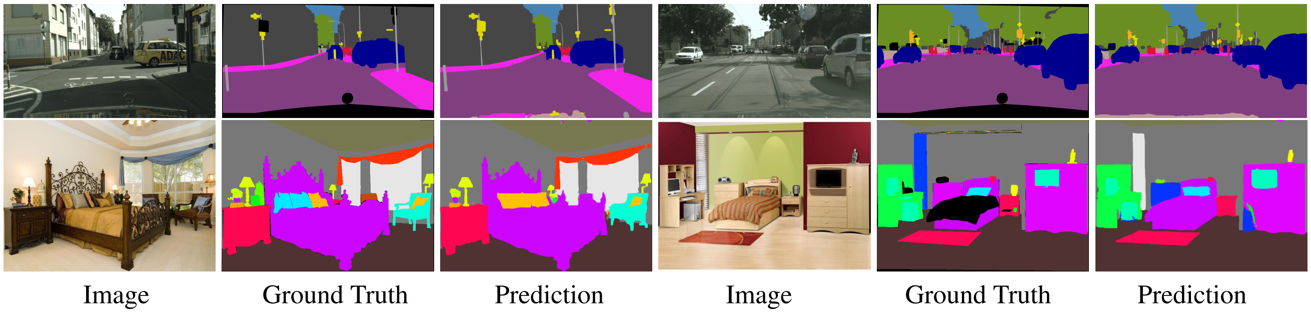


Figure 6. Qualitative results of semantic segmentation on Cityscapes [11] and ADE20K [57] datasets.

4.4. Cross-domain semantic segmentation

We evaluate cross-domain zero-shot benchmark [26]. We compare the proposed GSS with MSeg [26], a domain generalization algorithm (CCSA) [35] and a multi-task learning algorithm (MGDA) [41]. We test both HRNet-W48 [43] and Swin-Large [31] as backbone. As shown in Table 6, our GSS is superior to all competitors using either backbone. This suggests that generative learning could achieve more domain-generic representation than conventional discriminative learning counterparts.

Domain generic `maskige` Being independent to the visual appearance of images, `maskige` is intrinsically domain generic. To evaluate this, we transfer the `maskige` from MSeg to Cityscapes. As shown in Table 7, GSS can still achieve 79.5 mIoU (1% drop). In comparison, image representation transfer would double the performance decrease.

4.5. Qualitative evaluation

We evaluate the first-stage reconstruction quality of our GSS, UViM [25] and VQGAN [15]. As shown in Figure 5, GSS produces almost error-free reconstruction with clear and precise edges, and UViM fails to recognize some small objects while yielding distorted segmentation. VQGAN [15] achieves better classification accuracy but produces more ambiguous edge segmentation. As shown in Figure 6, GSS produces fine edge segmentation for interior furniture divisions on ADE20K [57] and accurately segments distant pedestrians and slender poles on Cityscapes [11].

5. Conclusion

In this paper, we have presented a *Generative Semantic Segmentation* (GSS) approach. Casting semantic segmentation as an image-conditioned mask generation problem, our formulation is drastically distinctive to conventional discriminative learning based alternatives. This is established on a novel notion of maskige and an efficient optimization algorithm in two stages: (i) Learning the posterior distribution of the latent variables for segmentation mask reconstruction, and (ii) minimizing the distance between posterior distribution and the prior distribution of latent variables for enabling input images to be conditioned on. Extensive experiments on standard benchmarks demonstrate that our GSS achieves competitive performance in comparison to prior art discriminative counterparts, whilst establishing new state of the art in the more challenging cross-domain evaluation setting.

Acknowledgments This work was supported in part by National Natural Science Foundation of China (Grant No. 62106050), Lingang Laboratory (Grant No. LG-QS-202202-07) and Natural Science Foundation of Shanghai (Grant No. 22ZR1407500).

References

[1] Gabriel J Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 2009. 5, 7

[2] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *CVPR*, 2018. 5

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint*, 2014. 2, 5

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint*, 2017. 1

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 6

[6] Ting Chen, Lala Li, Saurabh Saxena, Geoffrey Hinton, and David J Fleet. A generalist framework for panoptic segmentation of images and videos. *arXiv preprint*, 2022. 2, 4

[7] Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. In *ICLR*, 2021. 2

[8] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. 1, 2, 6

[9] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *NeurIPS*, 2021. 1, 2, 6, 7

[10] MMSegmentation Contributors. Openmmlab semantic segmentation toolbox and benchmark. `https://github.com/open-mmlab/mmsegmentation`, 2020. 5, 7

[11] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 5, 6, 7, 8

[12] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 5, 7

[13] 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. 6

[14] Patrick Esser, Robin Rombach, and Bjorn Ommer. A disentangling invertible interpretation network for explaining latent representations. In *CVPR*, 2020. 2

[15] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021. 2, 4, 6, 7, 8

[16] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 5, 7

[17] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, 2019. 6

[18] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 2013. 5, 7

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 2020. 2

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[21] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *ICCV*, 2019. 2, 6

[22] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 2

[23] Ge-Peng Ji, Keren Fu, Zhe Wu, Deng-Ping Fan, Jianbing Shen, and Ling Shao. Full-duplex strategy for video object segmentation. In *ICCV*, 2021. 1

[24] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint*, 2013. 1, 2, 3

[25] Alexander Kolesnikov, André Susano Pinto, Lucas Beyer, Xiaohua Zhai, Jeremiah Harmsen, and Neil Houlsby. Uvim: A unified modeling approach for vision with learned guiding codes. *arXiv preprint*, 2022. 2, 4, 5, 6, 7, 8

[26] John Lambert, Zhuang Liu, Ozan Sener, James Hays, and Vladlen Koltun. Mseg: A composite dataset for multi-domain semantic segmentation. In *CVPR*, 2020. 2, 5, 7, 8

[27] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *ICML*, 2016. 2

[28] Daiqing Li, Junlin Yang, Karsten Kreis, Antonio Torralba, and Sanja Fidler. Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization. In *CVPR*, 2021. 2

[29] Chen Liang, Wenguan Wang, Jiaxu Miao, and Yi Yang. Gmmseg: Gaussian mixture based generative semantic segmentation models. In *NeurIPS*, 2022. 2

[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 5

[31] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 3, 4, 6, 7, 8

[32] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1, 2, 6

[33] Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. *arXiv preprint*, 2022. 2

[34] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint*, 2016. 3, 5

[35] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *ICCV*, 2017. 7, 8

[36] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, 2014. 5, 7

[37] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012. 3, 4

[38] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, 2017. 5

[39] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, 2021. 1, 2, 4, 5, 6, 7

[40] Robin Rombach, Patrick Esser, and Björn Ommer. Making sense of cnns: Interpreting deep representations and their invariances with inns. In *ECCV*, 2020. 2

[41] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *NeurIPS*, 2018. 7, 8

[42] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*, 2015. 5

[43] Ke Sun, Yang Zhao, Borui Jiang, Tianheng Cheng, Bin Xiao, Dong Liu, Yadong Mu, Xinggang Wang, Wenyu Liu, and Jingdong Wang. High-resolution representations for labeling pixels and regions. *arXiv preprint*, 2019. 8

[44] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *NeurIPS*, 2017. 1, 2, 3, 4

[45] Girish Varma, Anbumani Subramanian, Anoop Namboodiri, Manmohan Chandraker, and CV Jawahar. Idd: A dataset for exploring problems of autonomous navigation in unconstrained environments. In *WACV*, 2019. 5

[46] Qiang Wan, Zilong Huang, Jiachen Lu, YU Gang, and Li Zhang. Seaformer: Squeeze-enhanced axial transformer for mobile semantic segmentation. In *ICLR*. 1

[47] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 2, 6

[48] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018. 6

[49] Zhisheng Xiao, Qing Yan, and Yali Amit. Generative latent flow. *arXiv preprint*, 2019. 2

[50] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, 2021. 1, 2, 6

[51] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, 2020. 5

[52] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *ECCV*, 2020. 6

[53] Oliver Zendel, Katrin Honauer, Markus Murschitz, Daniel Steininger, and Gustavo Fernandez Dominguez. Wilddash-creating hazard-aware benchmarks. In *ECCV*, 2018. 5, 7

[54] Li Zhang, Dan Xu, Anurag Arnab, and Philip HS Torr. Dynamic graph message passing networks. In *CVPR*, 2020. 2

[55] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 2, 6

[56] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, 2021. 1, 2, 6, 7

[57] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *IJCV*, 2019. 5, 7, 8