DreamTeacher: Pretraining Image Backbones with Deep Generative Models

Daiqing Li [1*] Huan Ling[1,2,3*] Amlan Kar[1,2,3] David Acuna[1,2,3]
Seung Wook Kim[1,2,3] Karsten Kreis[1] Antonio Torralba[4] Sanja Fidler[1,2,3]

[1]NVIDIA [2]University of Toronto [3]Vector Institute [4]MIT

Project page: https://research.nvidia.com/labs/toronto-ai/DreamTeacher/

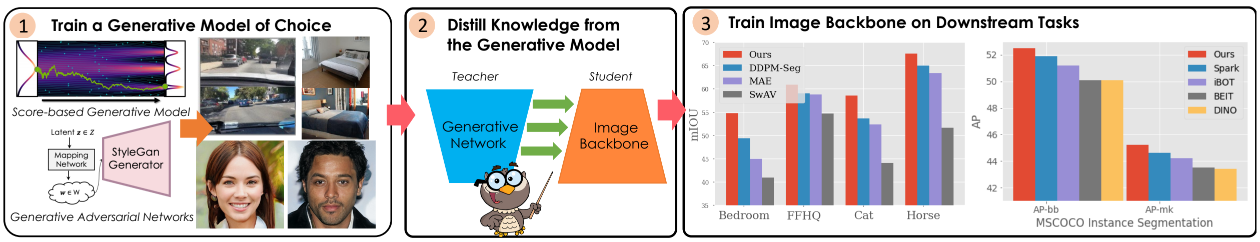

Figure 1. We propose DreamTeacher, a framework for distilling knowledge from a pre-trained generative network onto a target image backbone, as a generic pre-training mechanism that doesn't require labels. We investigate *feature distillation*, and optionally *label distillation* (when task-specific labels are available). Our DreamTeacher outperforms existing self-supervised methods on a variety of benchmarks.

Abstract

In this work, we introduce a self-supervised feature representation learning framework DreamTeacher that utilizes generative networks for pre-training downstream image backbones. We propose to distill knowledge from a trained generative model into standard image backbones that have been well engineered for specific perception tasks. We investigate two types of knowledge distillation: 1) distilling learned generative features onto target image backbones as an alternative to pretraining these backbones on large labeled datasets such as ImageNet, and 2) distilling labels *obtained from generative networks with task heads onto logits of target backbones. We perform extensive analyses on multiple generative models, dense prediction benchmarks, and several pre-training regimes. We empirically find that our DreamTeacher significantly outperforms existing self-supervised representation learning approaches across the board. Unsupervised ImageNet pre-training with DreamTeacher leads to significant improvements over ImageNet classification pre-training on downstream datasets, showcasing generative models, and diffusion generative models specifically, as a promising approach to representation learning on large, diverse datasets without requiring manual annotation.*

1. Introduction

Self-supervised representation learning is becoming an effective way of pre-training vision backbones [7,12,13,27,29]. The premise of this line of work is to leverage large unlabeled datasets as additional source of training data in order to boost performance of downstream networks, and to reduce the need for large labeled target datasets. Recent works have shown that self-supervised pre-training on ImageNet can now come close to supervised pre-training, even outperforming it on some downstream datasets and tasks such as pixelwise semantic and instance segmentation [13, 29, 63].

One of the dominant approaches to self-supervised representation learning are variants of contrastive learning, where the target backbone is trained to map transformed views of an image closer in latent space than images randomly drawn from the dataset [12]. Improvements to this paradigm include introducing spatial losses [63, 70, 71, 73], and improving training stability with fewer or no negative examples [13, 14, 27, 29].

Another line of work pursues reconstruction losses for supervision, where certain regions get masked from an input image, and backbones get trained to reconstruct them [21, 28, 64, 72], also known as Masked Image Modeling (MIM). This task is mostly treated as deterministic, ie supervising a single explanation for the masked region. This line of work typically investigates masking strategies, architecture design and training recipes to train better backbones. These methods have achieved state-of-the-art (SoTA) performance when applied to Vision Transformer-based backbones; however, recently sparse CNN-based image backbones [58] have been shown to be as performant.

In this paper, we argue for generative models as representation learners: for the simplicity of the objective – to

* Equal Contribution.

generate data, and intuitive representational power – generating high quality samples as an indication of learning semantically capable internal representations. Using generative networks as representation learners is not a novel concept. DatasetGAN and variants [4, 40, 82] proposed to add task-dependent heads on top of StyleGAN's or a diffusion model's features, and used these augmented networks as generators of labeled data, on which downstream networks are then trained. SemanticGAN [41] instead used StyleGAN with an additional task decoder as the task network itself – by encoding images into the latent space of the generative model and using the task head for producing perception output.

We introduce DreamTeacher, a representation learning framework that leverages generative models for pre-training downstream perception models via distillation. We investigate two types of distillation: 1) feature distillation, where we propose methods for distilling generative features to target backbones, as a general pre-training mechanism that does not require any labels. 2) label distillation: using task-heads on top of generative networks for distilling knowledge from a labeled dataset onto target backbones, in a semi-supervised regime. We focus our work on diffusion models [35, 54, 56] and GANs [26, 36, 37] as the choice of generative models. For target backbones, we focus on CNNs, for two major reasons. 1) CNN-based backbones have been shown to achieve SoTA representation learning performance for both contrastive and MIM approaches [44, 58, 62, 66], 2) SoTA generative models today (GANs and diffusion models) primarily still use CNNs internally. In preliminary experiments, we also explored vision transformer backbones, but found it challenging to distill features from CNN-based generative models into vision transformers. Generative models built with vision transformer architectures are nascent [2, 48], and hence we leave a thorough exploration of DreamTeacher with these architectures to future work.

We experimentally show that DreamTeacher outperforms existing self-supervised learning approaches on various benchmarks and settings. Most notably, when pre-trained on ImageNet without any labels, our method significantly outperforms methods that are pre-trained on ImageNet with full supervision, on several dense prediction benchmarks and tasks such as semantic segmentation on ADE20K [84], instance segmentation on MSCOCO [43] and on the autonomous driving dataset BDD100K [77]. On object-focused datasets with millions of unlabeled images [78, 82], our method, when trained solely on the target domain, significantly outperforms variants that are pre-trained on ImageNet with label supervision, and achieves new SoTA results. These results highlight generative models, especially diffusion-based generative models [20, 35, 56], as powerful representation learners that can effectively leverage diverse unlabeled datasets at scale.

2. Related Work

Discriminative Representation Learning. Early representation learning methods relied on handcrafted pretext tasks such as relative patch prediction [21], solving jigsaw puzzles [47], colorization [81], and relative rotation [25]. Instead, our pretext task is to predict features of a pretrained generative model, which in turn is trained with a simple and natural objective: maximize the log likelihood of the image data. The ability to synthesize and manipulate high quality samples is promising sign that generative networks learn both semantic and geometric knowledge internally [82].

Recent breakthroughs come from contrastive representation learning methods [12, 13, 27]. SimCLR [12] was the first to show competitive results in linear probing and transfer learning without using class labels, compared to supervised pre-training. Follow-up works MoCo [29], MoCoV2 [13] and BYOL [27] improve over the siamese network design with a memory bank and gradient stopping. However, these methods rely on heavy data augmentation [69] and heuristics to select the negative examples. This may not generalize well to datasets beyond well-curated object-centric datasets like ImageNet [18].

Another line of work [32, 63, 71, 73] aims to improve over the global contrastive objective and focuses on region-based features which are useful for dense prediction tasks. denseCL [63] extends MoCoV2 [13] to predict auxiliary dense features, PixPro [71] extends BYOL [27] to have pixel-wise consistency across two views, while DetCon [32] introduces masked pooling to focus on object-wise features. However, these methods require special designs for certain tasks [70, 71], or additional heuristics for complex scene datasets [32]. In our work, we focus on generative networks for representation learning specifically focused on various dense prediction tasks.

Generative Representation Learning. The ideas of leveraging generative models for learning representations for recognition tasks dates back to Hinton [33]. Recent works use advanced generative models and techniques to develop representation learning methods. BiGAN [22] proposed to jointly train an encoder with adversarial training objective. BigBiGAN [23] leveraged the advancement of BigGAN [5] and showed competitive linear probing results in ImageNet. Methods like iGPT [10] and VIM [79] pre-train large transformer networks with autoregressive generative pre-training objectives , achieving compelling linear probing results on ImagetNet, but they did not show results on dense prediction tasks. Furthermore, these methods train a single image backbone with both discriminative and generative objectives and thus cannot leverage the specific designs for each.

DatasetGAN [40, 82] was among the first to show that a pretrained GAN can significantly benefit perception tasks, especially in the low labeled data regime. Specifically, the authors added a task-specific head on top of StyleGAN and

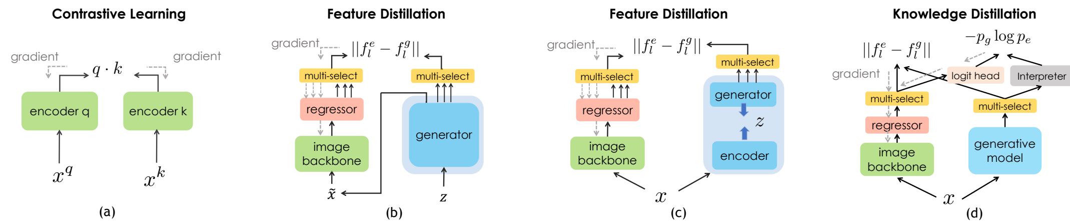

Figure 2. Different representation learning approaches: **(a)** a representative discriminative pretraining using a siamese-based network and contrastive loss, **(b)** our DreamTeacher generative pretraining framework when sampling examples from the generative model, **(c)** our DreamTeacher generative pretraining framework on encoded real data, **(d)** our mix distillation when a small number of labels are available (20-40 labeled data in our experiments). Multi-select means selecting features from different layers.

synthesized a labeled dataset for training downstream perception networks. SemanticGAN [41] proposed to model the joint distribution of images and labels. Inference was performed by first encoding the test images into the latent space of StyleGAN and then decoded the labels using the task-head. DDPM-seg [4] followed this line of work but used a denoising diffusion probabilistic model (DDPMs) instead of StyleGAN. Additionally, in GHFeat [74], a feature encoder is trained by feeding the output hierarchical feature into a fixed GAN generator for reconstruction. The authors demonstrated that the learned features can be used in both generative and discriminative tasks.

In our paper, we continue this line of work but focus on distilling knowledge from a pre-trained generative model, diffusion model specifically, to downstream image backbones as a general way of pre-training. We provide an extensive evaluation of generative networks in the context of representation learning on various benchmarks and tasks.

Knowledge Distillation. Hinton et al [34] were first to propose knowledge distillation as an effective means of improving performance – with the idea of distilling logits from a large teacher network into a smaller student network. FitNets [51] proposed to mimick the teacher's intermediate feature activations as additional hints for the student network. Follow-up works try to utilize different forms of knowledge from the teacher network: spatially [80], channel-wisely [53], and from multi-levels [11]. Usually, the teacher and student networks share a similar training objective, the network architecture, and require labels to train the teacher network. In our work, our generative model is treated as a teacher, and is trained without labels and the objective is not task-specific. Our student networks are image backbones of choice, which might not share a similar architecture as the teacher.

3. DreamTeacher Framework

We describe our DreamTeacher framework in the context of two scenarios: unsupervised representation learning where no labels are available during pre-training, and semi-supervised learning where a fraction of labels are available.

We utilize a trained generative model G and *distill* its learned representation into a target image backbone f. Our recipe for training f remains the same in both scenarios and choices of G and f. First, we create a *feature dataset* $D = \{x_i, \mathbf{f}_i^g\}_{i=1}^N$ of images x_i and corresponding features $\mathbf{f_i^g}$ extracted from the generative model. Next, we train f using the dataset D by distilling features $\mathbf{f}_i^g$ into the intermediate features of $f(x_i)$. We focus on convolutional backbones f, leaving exploration into transformers for future work. We drop subscript i for brevity from here on.

In Sec. 3.1, we describe the design of our unsupervised distillation process. We tackle the semi supervised regime in Sec. 3.2, where labels are available on a fraction of the pre-training dataset.

3.1. Unsupervised Representation Learning

For unsupervised representation learning given a feature dataset D, we attach feature regressors at different hierarchical levels of the backbone f to regress the corresponding generative features $\mathbf{f}_i^g$ from an image x_i. We first discuss creating a feature dataset, followed by the design of feature regressors and end by introducing our distillation objective.

Creating a feature dataset D. Generative models give us two distinct ways of creating our desired feature dataset D. One could sample images from the generative model G and record intermediate features from the generative process. In principle, this could synthesize datasets of infinite size, but may suffer from issues such as mode dropping, where the generative model may not have learned some parts of the distribution sufficiently well. We refer to such a dataset as a *synthesized dataset*. Instead, one could encode real images, labeled or unlabeled, into the latent space of the generative model G, using an encoding process. We refer to such a dataset as an *encoded dataset*.

A *synthesized dataset* D is created by sampling images $\tilde{x} \sim G(z)$, where z is sampled from the generative model G's prior distribution. We record hierarchical intermediate features from $G(z)$ as $\mathbf{f}^g = \{f_l^g\}_{l=1}^L$ where l denotes the hierarchy level of the features from a total of L levels. We employ this approach when using GANs [5, 9, 36] as G, due to their sampling speed, and inability to encode real images by design. Note that we are not concerned with bad samples, i.e. images with artifacts, as our main goal is to train the image backbone f to map images into features, regardless of image quality. This process is visualized in Fig. 2 (b). Also

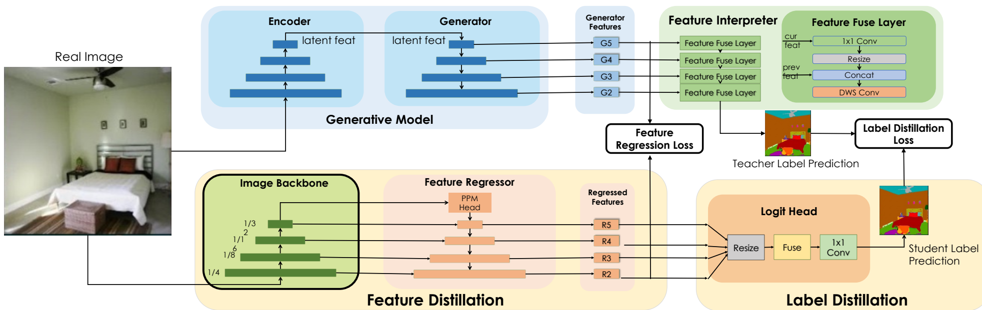

Figure 3. **DreamTeacher architecture**: Feature regression module (FR) maps and fuses multi-scale features of a (CNN) image backbone. We supervise FR with features from the generator's decoding network. We optionally add a *feature interpreter* [82] to the generator to train a task head with supervised labels – used to supervise the image backbone with label distillation loss.

see (a) for a side-by-side comparison of a representative discriminative pretraining paradigm.

Encoded dataset is created by encoding a real image x into the latent space of the generative model using an encoding process to get a latent variable $\tilde{z}$. Then, we similarly run the generative process and record hierarchical intermediate features from $G(\tilde{z})$ to obtain our dataset D. This process is visualized in Fig. 2(c). Also see Fig. 7 for encoded ImageNet images and their feature activation maps. For generative models that come with an encoder network by design, such as VAEs [38, 61], we can simply re-use it. For diffusion based generative models (DM) [20, 35, 56], which is the class of generative models we focus our investigation on, we use the forward diffusion process to encode a real image. Specifically, we run forward diffusion for T steps, followed by a single denoising step to extract hierarchical features f_l^g from intermediate layers of the denoising network, typically a U-Net [52]. See Fig. 7 for visualization of feature activation maps at different diffusion steps. The choice of T and the encoding process in diffusion models (stochastic [35] or deterministic [55]) can strongly affect properties of the trained model f. We systematically ablate these choices through experiments, and find that distilling stochastically encoded features, which we view as data augmentation in feature space, increases robustness of the downstream backbone f.

Both *synthesized* and *encoded* feature datasets can either be pre-computed *offline*, or created *online* while training f. In practice, we use *online sampling* for synthesized datasets, and *online encoding* for encoded datasets to allow fast in-memory access and efficient materialization and removal of samples and corresponding high dimensional features. This allows us to scale to pre-training with datasets and features $\mathbf{f}^g$ of any size without additional pre-processing and storage costs. Online encoding is also the natural choice when using stochastic encoding techniques in diffusion models, since an offline dataset could only store one or a few samples from all possible stochastic encodings of a real image.

Feature Regressor. In order to distill generative representations $\mathbf{f}^g$ into a general backbone f, we design a feature regressor module that maps and aligns the image backbone's features with the generative features. Inspired by the design of the Feature Pyramid Network (FPN) [42], our feature regressor takes multi-level features from the backbone f and uses a top-down architecture with lateral skip connections to fuse the backbone features and outputs multi-scale features. We apply a Pyramid Pooling Module (PPM) from PSPNet [83] similar to [68], on the last layer of the image backbones before the FPN branch to enhance feature mixing. Fig. 3 (bottom) visually depicts this architecture.

Feature Distillation. Denote intermediate features from encoder f at different levels as $\{f_2^e, f_3^e, f_4^e, f_5^e\}$, and the corresponding feature regressor outputs as $\{f_2^r, f_3^r, f_4^r, f_5^r\}$. We use a 1×1 convolution to match the number of channels in f_l^r and f_l^g, if they are different. Our feature regression loss is simple and is inspired by FitNet [51], which proposed distilling knowledge from a teacher onto a student network by mimicking intermediate feature activations:

$$\mathcal{L}_{MSE} = \frac{1}{L}\sum_{l}^{L} \|f_l^r - \mathbb{W}(f_l^g)\|_2^2 \tag{1}$$

Here, $\mathbb{W}$ is a non-learnable whitening operator implemented as LayerNorm [1], which normalizes differing feature magnitudes across layers. Layer number $l = \{2, 3, 4, 5\}$ corresponds to features at 2^l stride relative to the input resolution.

Additionally, we explore the activation-based Attention Transfer (AT) [80] objective. AT distills a one dimensional "attention map" per spatial feature, using an operator defined as $F_{sum}^p(A) = \sum_i^C |A_i|^p$ to sum the power p of the absolute values of the feature activation A across channel dimension C, which improves convergence speed over regressing high dimensional features directly. Specifically,

$$\mathcal{L}_{AT} = \frac{1}{L}\sum_{l}^{L}\sum_{j\in I} \left\| \frac{Q_{l,j}^r}{\|Q_{l,j}^r\|_2} - \frac{Q_{l,j}^g}{\|Q_{l,j}^g\|_2} \right\|_p \tag{2}$$

where $Q_{l,j}^r = vec(F_{sum}^p(f_{l,j}^r))$, $Q_{l,j}^g = vec(F_{sum}^p(f_{l,j}^g))$ are respectively the j-th pair in layer l of the regressor's and generative model's features in vectorized form. We follow [80] to use $p = 2$ in our experiments.

Our combined feature regression loss is:

$$\mathcal{L}_{feat} = \mathcal{L}_{MSE} + \lambda_{AT}\mathcal{L}_{AT} \tag{3}$$

where λ_{AT} controls the weighting of the loss $\mathcal{L}_{AT}$. We choose $\lambda_{AT} = 10.0$ in our experiments, to make the two losses in the same scale. We empirically ablate choices of the loss function and feature regressor designs.

3.2. Label-Guided Representation Learning

In the semi-supervised setting, where a fraction of downstream task labels are available for pre-training, we train a task-dependent branch, called a *feature interpreter*, on top of the frozen generative network G in a supervised manner, following DatasetGAN [82]. While DatasetGAN synthesized a labeled dataset for training downstream task networks, we instead use soft label distillation for both *encoded* and *synthesized* datasets, i.e. we include predicted soft labels in our feature dataset D. This is visualized in Fig.2(d). We first describe the architecture of the *feature interpreter* followed by our distillation objective for soft labels.

Feature Interpreter. We utilize a similar design to BigDatasetGAN [40], which improves the interpreter design over DatasetGAN with better memory efficiency and prediction accuracy. Specifically, the interpreter takes multi-level features f_l^g from the generator as inputs which are fed into a series of *Feature Fusion Layers* (see Fig 3) to lower the feature dimension and fuse with the next-level features, to finally output per-pixel logits. We follow BigDatasetGAN's interpreter design and only replace the convolutional fused block with depth-wise separable convolutions [17], Group Norm [67], and Swish activation [49].

We explore training the interpreter branch with segmentation labels, and use a combination of the cross-entropy and Dice [57] objectives for training:

$$\mathcal{L}_{interpreter} = \mathcal{H}(I_\theta(f_l^g), y) + \lambda_d \mathcal{D}(I_\theta(f_l^g), y), \quad (4)$$

where I_θ are the weights of the feature interpreter, y are the task labels. $\mathcal{H}(\cdot,\cdot)$ denotes pixel-wise cross-entropy loss, and $\mathcal{D}(\cdot,\cdot)$ is Dice Loss. λ_d is a hyperparameter to weigh the dice loss. We use $\lambda_d = 3.0$ in all our experiments following [57].

Label Distillation. We follow [34] for label distillation. Specifically, we use:

$$\mathcal{L}_{ld} = \mathcal{H}(P_\tau^g, P_\tau^r), \quad (5)$$

where P_τ^g and P_τ^r are the logits from the feature interpreter and the logit-head of the target image backbone f, respectively. $\mathcal{H}$ is the cross-entropy, with temperate τ, controlling the sharpness of the output distribution. We use $\tau = 4.0$ in all our experiments following [34].

We use the label distillation objective in conjunction with our feature distillation objective:

$$\mathcal{L}_{mix} = \mathcal{L}_{feat} + \lambda_{ld}\mathcal{L}_{ld} \quad (6)$$

where λ_{ld} is a hyperparameter controlling the weighting between the losses, which we use $\lambda_{ld} = 1.0$ in our experiment.

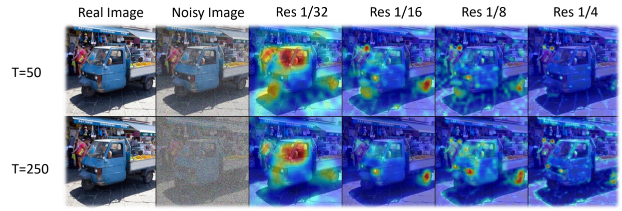

Figure 4. **ADM feature visualization (ImageNet)**. We visualize ADM feature activation maps at different resolution blocks (columns) at different diffusion time steps T (rows). At lower resolution blocks, features activate on objects like humans and cars. For higher resolution block, features focus on smaller parts like wheels and headlights. With increasing T, feature activations become smoother.

We pre-train the image backbone f using the mixed distillation losses over all images in our pre-training dataset, either labeled or unlabeled. Annotated labels are only used for training the feature interpreter, and we only use soft labels from the feature interpreter for pre-training f with distillation.

4. Experiments

In this section, we first experimentally evaluate the performance of DreamTeacher for both: self-supervised representation learning and semi-supervised learning (Subsec. 4.1). We then additionally investigate the performance of our model for *in-domain-pretraining* (Subsec. 4.2). In the *in-domain* setting, the same target dataset is used for both pretraining and finetuning, and the backbones are initialized from scratch. Finally, we ablate different generative models and design choices of DreamTeacher (Subsec. 4.3).

We investigate several generative models: for GANs, we use unconditional BigGAN [5], ICGAN [9], StyleGAN2 [37] and for diffusion-based model, ADM [20], and Stable Diffusion (SD) Models [50]. We use four datasets for pre-training, both for training the generative models, as well as knowledge distillation to downstream backbones. We use *BDD100K* [77], *ImageNet-1k(IN1k-1M)*, *LSUN* [78] and *FFHQ* [36], which contain 100k, 1.28 million, 10 million, and 100k images, respectively. We focus on convolutional networks as target image backbones.

4.1. ImageNet Evaluation and Transfer

Imagenet Pretraining. We first validate the effectiveness of DreamTeacher for ImageNet pretraining. In this setting, we follow the recent SoTA method, SparK [58], and evaluate two convolutional architectures as downstream backbones, ConvNext-B [44] and ResNet50 [31]. Following common practice in the literature [28, 58], we pre-train image backbones unsupervised on ImageNet-1k. For a comparison with transformer-based self-supervised methods, we follow SparK's methodology [58] and pre-train a modern CNN-based backbone ConvNeXt [44] with a similar number of parameters. Additionally, to ensure a fair comparison with CNN-based self-supervised methods, we pre-train and evaluate a classical backbone, ResNet-50.

Pre-training Method	PT task	Arch.	Eff. epoch	Cls Acc.	Det. AP^{bb}	AP^{bb}_{75}	Seg. AP^{mk}	AP^{mk}_{75}
Vision Transformer Backbone								
Supervised [28]	-	ViT-B	300	82.3	49.8	53.8	43.2	46.5
MoCov3 [15]	CL	ViT-B	1600	83.2	-	-	-	-
DINO [8]	CL	ViT-B	1600	82.8	50.1	54.3	43.4	47.0
BEiT [3]	MIM	ViT-B	800	83.2	50.1	54.6	43.5	47.1
MAE [28]	MIM	ViT-B	1600	83.6	-	-	-	-
iBOT [85]	MIM + CL	ViT-B	1600	84.0	51.2	55.5	44.2	47.7
Convolutional Backbone								
Supervised [44]	-	ConvX-B	300	83.8	51.2	55.5	44.3	47.9
SparK [58]	MIM	ConvX-B	1600	**84.8**	51.9	56.5	44.6	48.4
DT-feat.distil. w/ ADM [20]	GEN	ConvX-B	*600	83.9	**52.5**	**57.4**	**45.2**	**49.0**

Table 1. **Comparing DreamTeacher with SoTA self-supervised methods on ImageNet and instance segmentation on COCO.** All the baselines including ADM are pre-trained on ImageNet-1k. For ImageNet classification, we adopt SparK's fine-tuning setting with resolution 224. For COCO, we follow iBOT to fine-tune Cascade Mask R-CNN [6] for 12 (1×) epochs. Average precisions of detection box (AP^{bb}) and segmentation mask (AP^{mk}) on val2017 are reported. For a fair comparison, both our method and baselines follow iBOT fine-tuning schedule and setting. Our DT pre-training task is highlighted as generative(GEN) comparing to contrastive(CL) and masking(MIM) based objectives. *Our effective epochs includes 400 epochs generative model training and 200 epochs feature distillation training.

Pre-training (ResNet-50)	PT task	Eff. epoch	Cls. (Acc.)	1× Schedule AP^{bb}	AP^{mk}	2× Schedule AP^{bb}	AP^{mk}
Supervised	-	-	79.8	38.9	35.4	41.3	37.3
SimSiam [14]	CL	800	79.1	-	-	-	-
MoCo [29]	CL	800	-	38.5	35.1	40.8	36.9
MoCov2 [13]	CL	1600	79.8	40.4	36.4	41.7	37.6
SimCLR [12]	CL	4000	80.0	-	-	-	-
InfoMin [59]	CL	800	-	40.6	36.7	42.5	38.4
BYOL [27]	CL	1600	80.0	40.4	37.2	42.3	38.3
SwAV [7]	CL	1200	80.1	-	-	42.3	38.2
SparK [60]	MIM	1600	**80.6**	41.6	37.7	43.4	39.4
DT-feat.distil. w/ ADM [20]	GEN	*600	80.2	**44.1**	**40.1**	**45.1**	**40.8**

Table 2. **ResNet-50 results on ImageNet and COCO instance segmentation.** For ImageNet classification, we follow SparK's fine-tuning setting with resolution 224. Top-1 accuracy (Acc) on ImageNet val set is reported. For COCO, Mask R-CNN [30] ResNet50-FPN is equally fine-tuned for 12 or 24 epochs (1× or 2×), following the same setup as SparK. *Our effective epochs includes 400 epochs generative model training and 200 epochs feature distillation training.

Pre-training (ResNet-50)	ADE20k mIoU	BDD100k AP^{bb}	AP^{mk}
Supervised	40.9	26.1	20.2
SimCLR [12]	39.9	24.5	20.6
SparK [60]	40.5	25.7	22.4
SimSiam [14]	40.6	26.3	22.7
MoCov2 [13]	40.9	26.9	22.9
denseCL [63]	41.1	27.1	23.4
SwAV [7]	41.2	25.6	22.2
BYOL [27]	41.6	26.2	22.6
PixPro [71]	41.6	27.2	23.1
DT-feat.distil. w/ ADM [20]	**42.5**	**28.3**	**24.8**

Table 3. **Transfer learning: ADE20k and BDD100k.** All methods are pre-trained on ImageNet-1k and fine-tuned on downstream tasks. For ADE20k, we follow [44] to use UperNet [68] and fine-tune for 160k iterations, reported number is mean IoU at single scale. For BDD100k, we follow official setup [77] to use Mask R-CNN ResNet50-FPN fine-tune for 36 (3×) epochs.

Implementation. We use pre-trained unconditional ADM with resolution 256 from the official release. We only use horizontal flip augmentation and train using LAMB [76] optimizer with a batch size of 2048. We adopt a cosine-annealing learning rate with peak value = 0.0002 × *batchsize* / 256. See appendix for other hyperparameters.

Transferring to Downstream Tasks. We assess the quality of learned representations obtained using DreamTeacher by fine-tuning the pre-trained backbone with additional heads per task (see Appendix for implementations). We test downstream transfer performance for ImageNet classification and COCO [43] instance segmentation, which are representative global and spatial image understanding tasks commonly used in literature. Prior self-supervised learning methods have excelled at ImageNet classification, and have recently shown improvement over supervised ImageNet pre-training for spatial understanding tasks such as object detection and segmentation that are much more cost-intensive to label. Additionally, we also include linear probing experiments on ImageNet for both classification and semantic segmentation tasks in the Appendix (Table 12).

Discussion. Comparing to self-supervised methods based on vision-transformer, DreamTeacher outperforms existing approaches in both detection and segmentation, and performs on par in the classification setting (Table 1). Specifically, DreamTeacher achieves 52.5 AP^{bb} and 45.2 AP^{mk} on the COCO instance segmentation task outperforming the SoTA transformer-based method iBOT by $+1.3$ and $+1.0$. DreamTeacher also outperforms the recently proposed sparse-convolution based MIM method SparK [58], in the tasks of detection and segmentation by $+0.6$ and $+0.6$, respectively. We notice that our method does not outperform this baseline on the task of image classification. This may likely be due to our approach of distilling spatial features

Pre-training (ResNet-50)	PT task	Eff. epoch	BDD100k Ins. AP^{bb}	AP^{mk}
Supervised [77]	-	-	26.1	20.2
BYOL [27]	CL	5000	23.9	20.0
SparK [58]	MIM	2500	24.4	20.6
DT-feat.distil. w/ StyleGAN2 [37]	GEN	*900	25.1	21.4
DT-feat.distil. w/ ADM [20]	GEN	*900	**26.7**	**22.9**

Table 4. **In-domain pre-training on BDD100k.** We follow the recommendation of [24] to pre-train contrastive and masking based self-supervised method with long schedule for small dataset like BDD100k with 70k train images. We finetune on BDD100k instance segmentation task using Mask R-CNN ResNet50-FPN for 36(3×) epochs.

method	backbone	params	pre-data	Bedroom-28	FFHQ-34	Cat-15	Horse-21
classific. sup.	RN101	43M	IN1k-1M	34.4	53.6	38.8	51.1
classific. sup.	ConvNX-B	89M	IN21k-14M	41.0	59.2	47.3	56.0
SwAV [7]	RN50-w2	94M	task domains	41.0	54.7	44.1	51.7
MAE [28]	ViT-L	305M	task domains	45.0	58.8	52.4	63.4
DatasetGAN [82]	RN101	43M	task domains	31.3	57.0	36.5	45.4
DatasetDDPM [4]	RN101	43M	task domains	47.9	56.0	47.6	60.8
DDPM-seg [4]	UNet	554M	task domains	49.4	59.1	53.7	65.0
DT-mix.distil. w/ ADM [20]	RN101	43M	task domains	49.9	59.4	56.7	65.9
DT-mix.distil. w/ ADM [20]	ConvNX-B	89M	task domains	**54.8**	**61.2**	**58.6**	**67.6**

Table 5. **Label-efficient semantic segmentation benchmark.** We compare our DreamTeacher (DT) with various representation learning baselines. Our *DT-mix.distil.* with ResNet 101 backbone (only 43M parameters) beats all baselines, some with 10x the number of parameters. We also show our method with ConvNX-B achieves the new SoTA without using any extra data, i.e. IN1k-1M or IN21k-14M.

from the generative model, which might contain more semantically localized information for generation (visualized in Fig. 7), which empirically seems to favor dense prediction tasks. It is also worth noting that our method is $\sim 2.5\times$ more efficient than SparK w.r.t. effective training epochs [58] on ImageNet (600 vs 1600). This number includes training steps of the generative model, ADM.

In Table 2 we show results for Resnet-50 using SparK's setting and parameters. Specifically, we evaluate ImageNet classification performance with full fine-tuning and COCO instance segmentation with two schedules (1× and 2×). Similar to the previous experiment, we achieve comparable performance as baselines for ImageNet classification. For COCO instance segmentation, we notably outperform all contrastive methods and the masking-based approach SparK (+2.5 AP^{bb} for 1× and +1.7 AP^{bb} for 2× schedule). In Table 3, we further evaluate transfer learning on the ADE20k semantic segmentation task and BDD100k instance segmentation task. We include SoTA contrastive methods for dense prediction tasks, denseCL [63] and PixPro [71]. Our approach using generation as pre-training task outperforms both global and dense contrastive pre-training tasks as well as the masked image modelling task.

4.2. In-domain Pre-training

For in-domain pre-training, we first pre-train the backbone with various self-supervised training approaches. Pre-training efficacy is evaluated by fine-tuning the backbone on different tasks with label supervision, on the same dataset. Note that both baselines and DreamTeacher use randomly initialized downstream backbones. We evaluate unsupervised pre-training using the BDD-100k benchmark and semi-supervised pre-training using multiple datasets from the label efficiency benchmark used by [4, 40, 82].

BDD100k Benchmark. We pre-train all self-supervised learning methods, including DreamTeacher on 70k unlabeled images from BDD100k. We then evaluate all methods on BDD100k, which contains 10k images annotated with semantic, instance and panoptic labels. We follow the official dataset split, using 7k labels for supervised training. Results are reported on the validation set (1k images). We use a Resnet-50 [31] backbone for all methods.

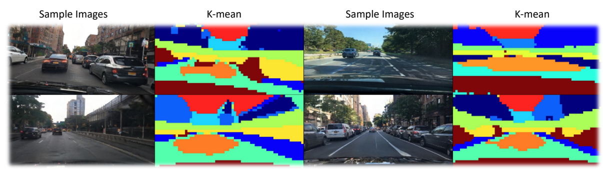

Figure 5. **K-means clustering of StyleGAN2's features trained on BDD100K.** We run kmeans clustering ($k = 10$) on 10k sampled features, and show unsupervised segmentation maps on sampled images. Notice that the clusters are consistent across images (car, sky, tree etc), indicating a semantic meaning of the generative features.

Feature Visualization. We visualize the knowledge learned by different generative models in Fig. 5. Specifically, we show scenes sampled from StyleGAN2 trained on BDD100k. We perform k-means clustering ($k = 10$) of StyleGAN's features and visualize clusters with different colors. Notice that the clusters roughly correspond to major semantic classes.

Results. In Table 4, we compare DreamTeacher with the representative contrastive method BYOL and recently proposed MIM-based method SparK on BDD100k instance segmentation task. As investigated in [24], contrastive and masking-based self-supervised methods require a longer pre-training schedule to converge on a small in-domain dataset. We pre-trained backbones using DreamTeacher feature distillation with StylegGAN2 and ADM, and the effective epochs comprise 300 training epochs of the generative model and 600 training epochs for feature distillation. Our methods outperform contrastive and masking-based techniques significantly for in-domain pre-training with better training efficiency. Notably, our method with ADM outperforms the ImageNet supervised pre-trained backbone, showing promising results without relying on large-scale curated datasets like ImageNet. See appendix for qualitative results and semantic segmentation and panoptic segmentation results.

Label-efficient Benchmarks. We now evaluate in-domain pre-training in our semi-supervised setting. We follow the setup in DDPM-seg [4] and train on "bedroom", "cat" and "horse" categories from LSUN [78], and human faces from FFHQ [36] (at 256x256 resolution). We evaluate semantic segmentation, where the datasets have 28, 15, 21, 34 semantic classes, respectively. Datasets contain only 40, 30, 30 and 20 labeled images. We pre-train all backbones from scratch, i.e. without ImageNet pre-trained initialization. We use UPerNet [68] for semantic segmentation. Note that some baselines utilize different settings. DatasetGAN [82] and DatasetDDPM [4] both train a small task-specific head on top of a pre-trained generative model, and generate a large labeled dataset for training a downstream network. On the other hand, DDPM-seg directly leverages the diffusion-based generative model with a task head as the segmentation network.

Results are reported in Table 5 . We highlight several key observations below:

- Given the same backbone, ResNet101, DreamTeacher trained with our mixed distillation (Eq. 6) outperforms DatasetDDPM across all datasets. We outperform DatasetDDPM by 3.4% on FFHQ-34, and 9.1% on Cat-15.
- Using both a 10x and a 6x smaller backbone (ResNet-101 and a ConvNX-B [44], respectively), we outperform DDPM-Seg on all classes. On Bedroom-28 and Cat-15, we improve over the baseline by more than 5%.
- Given the same backbone, our method significantly outperforms pre-training with ImageNet classification labels. With ConvNX-B [44], our proposed approach is better than ImageNet pre-training by more than 10% on Bedroom-28, Cat-15, and Horse-21. These results may indicate that if the in-domain datasets are sufficiently large relative to the complexity of the task, in-domain pre-training is more effective than pre-training on large generic datasets like ImageNet. Note that this is true for both semi-supervised (these results) and unsupervised pre-training (Table 4).

4.3. Ablation Studies

We first ablate DreamTeacher with different generative models in Table 6. Result shows ADM trained on IN1k-1M has the highest downstream performance. We also exploited off-the-shelf Stable Diffusion trained on LAION-400M and pretrain backbone on IN1k-1M. However, it performs slightly worse than ADM trained on IN1k-1M. In Table 7 we ablate our proposed distillation losses. Mixing feature- and label- distillation achieves the best performance except for the FFHQ-34 dataset. We demonstrate our design choices of the decoder used in pre-training in Table 8, loss functions in Table 9, encoding modes (deterministic and stochastic) in Table 10 and diffusion steps in Table 11. Ablation studies pre-train backbone for 100 epochs and report performance on BDD100k instance segmentation. These results confirm our choices.

Limitations: Our framework relies on generative models for representation learning, and training a generative model on large-scale datasets at high resolution is costly, especially with diffusion-based models. Further, our feature distillation method only considers features at the same spatial resolution and we limit our scope to CNN-based image backbones. Distilling features into vision transformers is for future work.

Pre-training (ResNet-50)	Gen. Data	Pre. Data	ADE20k mIoU	COCO AP^{bb}	COCO AP^{mk}
DT-feat.distil. w/ BigGAN [5]	IN1k-1M	IN1k-1M	40.8	40.7	36.9
DT-feat.distil. w/ ICGAN [9]	IN1k-1M	IN1k-1M	41.2	40.0	36.5
DT-feat.distil. w/ SD1.4 [50]	LAION-400M	IN1k-1M	41.4	43.3	39.4
DT-feat.distil. w/ ADM [20]	IN1k-1M	IN1k-1M	42.5	44.1	40.1

Table 6. **Ablation study with different generative models using DreamTeacher.** We use off-the-shelf SD with version 1.4 pre-trained on LAION-400M without finetuning, and it performs slightly worse than DT with ADM, which is trained on ImageNet-1k.

Loss	Bedroom-28	FFHQ-34	Cat-15	Horse-21
feat distil.	53.1	61.1	58.2	64.7
label distil.	54.6	61.3	58.4	64.4
mix distil.	54.8	61.2	58.6	67.6

Table 7. **Ablating feature/label distillation.** We pretrain ConvNeXt-B to convergence. Feature distillation (FD) does not leverage labels in pre-training, yet performs competitively.

Decoder	*Box mAP*	*Mask mAP*
FPN	23.6	20.3
FPN+Atten. layer	23.9	20.7
PaFPN	24.0	20.8
FPN+PPM	25.1	21.4

Table 8. **Ablating feat. regressors** We pretrain ResNet50 with FT. We compare FPN with an attention layer, and add a bottom-up branch to fuse FPN features (PaFPN).

Decoder	*Box mAP*	*Mask mAP*
Finnet(MSE)	23.6	20.3
AT	22.3	19.3
MSE+AT	25.0	21.6

Table 9. **Ablating distillation losses.** We pretrain ResNet50 with MSE or AT loss using feature distill. Combining losses achieves best results.

Encoding	*Box mAP*	*Mask mAP*
Determin.	23.4	20.8
Stochastic	24.3	21.1

Table 10. **Ablating DDPM encoding.** We use DDIM [55] sampling for deterministic encoding. In both cases, backbone is pretrained for 100 epochs.

Steps	*Box mAP*	*Mask mAP*
T=50	23.8	20.4
T=150	23.9	20.6
T=250	24.4	21.1
T=350	23.4	20.1

Table 11. **Ablating # of diffusion steps.** We pretrain ResNet50 with feature distillation using different # of diffusion steps. Performance varies with T.

5. Conclusion

We proposed DreamTeacher, a framework for distilling knowledge from generative models onto target image backbones. We investigated several different settings, generative models, target backbones, and benchmarks. Experiments show that generative networks that leverage large unlabeled datasets with generative objectives learn semantically meaningful features that can be successfully distilled on target image backbones. We empirically show our generative-based pre-training method outperforms existing contrastive based and MIM based self-supervised learning approaches in several challenging benchmarks including COCO, ADE20K and BDD100K. We hope our exploration and discovery can inspire future works to study generative pre-training and leveraging geneartive models for vision tasks.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

[2] Fan Bao, Chongxuan Li, Yue Cao, and Jun Zhu. All are worth words: a vit backbone for score-based diffusion models. *arXiv preprint arXiv:2209.12152*, 2022. 2

[3] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 6

[4] Dmitry Baranchuk, Ivan Rubachev, Andrey Voynov, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models, 2021. 2, 3, 7, 8

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 2, 3, 5, 8, 13

[6] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: high quality object detection and instance segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 43(5):1483–1498, 2019. 6

[7] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *NeurIPS*, 33:9912–9924, 2020. 1, 6, 7, 12

[8] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 6

[9] Arantxa Casanova, Marlene Careil, Jakob Verbeek, Michal Drozdzal, and Adriana Romero Soriano. Instance-conditioned gan. *Advances in Neural Information Processing Systems*, 34:27517–27529, 2021. 3, 5, 8, 13

[10] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. pages 1691–1703. PMLR, 2020. 2

[11] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review. In *CVPR*, pages 5008–5017, 2021. 3

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. pages 1597–1607. PMLR, 2020. 1, 2, 6, 12

[13] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 1, 2, 6, 12

[14] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, pages 15750–15758, 2021. 1, 6, 12

[15] Xinlei Chen*, Saining Xie*, and Kaiming He. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021. 6

[16] Xiaokang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang. Semi-supervised semantic segmentation with cross pseudo supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2613–2622, 2021. 15

[17] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017. 5, 13

[18] Elijah Cole, Xuan Yang, Kimberly Wilber, Oisin Mac Aodha, and Serge Belongie. When does contrastive visual representation learning work? In *CVPR*, 2022. 2

[19] 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*, pages 3213–3223, 2016. 12

[20] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 2, 4, 5, 6, 7, 8, 12, 13, 14, 16

[21] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015. 1, 2

[22] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016. 2, 12

[23] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. *NeurIPS*, 32, 2019. 2, 12

[24] Alaaeldin El-Nouby, Gautier Izacard, Hugo Touvron, Ivan Laptev, Hervé Jegou, and Edouard Grave. Are large-scale datasets necessary for self-supervised pre-training? *arXiv preprint arXiv:2112.10740*, 2021. 7, 12, 13

[25] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 2

[26] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 2

[27] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *NeurIPS*, 33:21271–21284, 2020. 1, 2, 6, 7, 12, 13

[28] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 16000–16009, 2022. 1, 5, 6, 7

[29] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020. 1, 2, 6

[30] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 6, 15

[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5, 7

[32] Olivier J Hénaff, Skanda Koppula, Jean-Baptiste Alayrac, Aaron Van den Oord, Oriol Vinyals, and João Carreira. Efficient visual pretraining with contrastive detection. In *ICCV*, pages 10086–10096, 2021. 2, 15

[33] Geoffrey Hinton. To recognize shapes, first learn to generate images. *Progress in brain research*, 165:535–47, 02 2007. 2

[34] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015. 3, 5

[35] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 2, 4

[36] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 2, 3, 5, 7

[37] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 2, 5, 7, 13, 14

[38] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 4

[39] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *CVPR*, pages 6399–6408, 2019. 13, 15

[40] Daiqing Li, Huan Ling, Seung Wook Kim, Karsten Kreis, Sanja Fidler, and Antonio Torralba. Bigdatasetgan: Synthesizing imagenet with pixel-wise annotations. In *CVPR*, pages 21330–21340, 2022. 2, 5, 7, 12

[41] 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*, pages 8300–8311, 2021. 2, 3

[42] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 4, 13

[43] 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*, pages 740–755. Springer, 2014. 2, 6

[44] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, pages 11976–11986, 2022. 2, 5, 6, 8

[45] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 12

[46] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 14

[47] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, pages 69–84. Springer, 2016. 2

[48] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022. 2

[49] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017. 5, 13

[50] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 5, 8

[51] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014. 3, 4

[52] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 4

[53] Changyong Shu, Yifan Liu, Jianfei Gao, Zheng Yan, and Chunhua Shen. Channel-wise knowledge distillation for dense prediction. In *ICCV*, pages 5311–5320, 2021. 3

[54] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2015. 2

[55] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 4, 8

[56] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 2, 4

[57] Carole H Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pages 240–248. Springer, 2017. 5

[58] Keyu Tian, Yi Jiang, Qishuai Diao, Chen Lin, Liwei Wang, and Zehuan Yuan. Designing bert for convolutional networks: Sparse and hierarchical masked modeling. *arXiv:2301.03580*, 2023. 1, 2, 5, 6, 7, 13, 14

[59] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *arXiv preprint arXiv:2005.10243*, 2020. 6

[60] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM Transactions on Graphics (TOG)*, 40(4):1–14, 2021. 6, 12

[61] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in Neural Information Processing Systems*, 33:19667–19679, 2020. 4

[62] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. *arXiv preprint arXiv:2211.05778*, 2022. 2

[63] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *CVPR*, pages 3024–3033, 2021. 1, 2, 6, 7, 12

[64] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *CVPR*, pages 14668–14678, 2022. 1

[65] Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm. *arXiv preprint arXiv:2110.00476*, 2021. 14

[66] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. *arXiv preprint arXiv:2301.00808*, 2023. 2

[67] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 5, 13

[68] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 418–434, 2018. 4, 6, 8, 13, 14, 15

[69] Tete Xiao, Xiaolong Wang, Alexei A Efros, and Trevor Darrell. What should not be contrastive in contrastive learning. *arXiv preprint arXiv:2008.05659*, 2020. 2

[70] Enze Xie, Jian Ding, Wenhai Wang, Xiaohang Zhan, Hang Xu, Peize Sun, Zhenguo Li, and Ping Luo. Detco: Unsupervised contrastive learning for object detection. In *ICCV*, pages 8392–8401, 2021. 1, 2

[71] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *CVPR*, pages 16684–16693, 2021. 1, 2, 6, 7

[72] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *CVPR*, pages 9653–9663, 2022. 1

[73] Yuwen Xiong, Mengye Ren, and Raquel Urtasun. Loco: Local contrastive representation learning. *NeurIPS*, 33:11142–11153, 2020. 1, 2

[74] Yinghao Xu, Yujun Shen, Jiapeng Zhu, Ceyuan Yang, and Bolei Zhou. Generative hierarchical features from synthesizing images. In *CVPR*, 2021. 3

[75] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017. 12

[76] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*, 2019. 6, 13

[77] 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*, pages 2636–2645, 2020. 2, 5, 6, 7, 12, 13

[78] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 2, 5, 7

[79] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021. 2

[80] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016. 3, 4

[81] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, pages 649–666. Springer, 2016. 2

[82] Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. Datasetgan: Efficient labeled data factory with minimal human effort. In *CVPR*, pages 10145–10155, 2021. 2, 4, 5, 7, 8

[83] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 4, 13

[84] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019. 2

[85] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *International Conference on Learning Representations (ICLR)*, 2022. 6, 14