Do text-free diffusion models learn discriminative visual representations?

Soumik Mukhopadhyay*[1] Matthew Gwilliam*[1] Yosuke Yamaguchi†[2] Vatsal Agarwal†[1]
Namitha Padmanabhan[1] Archana Swaminathan[1] Tianyi Zhou[1]
Abhinav Shrivastava[1]

[1]University of Maryland, College Park [2]Waseda University

Abstract

While many unsupervised learning models focus on one family of tasks, either generative or discriminative, we explore the possibility of a unified representation learner: a model which addresses both families of tasks simultaneously. We identify diffusion models, a state-of-the-art method for generative tasks, as a prime candidate. Such models involve training a U-Net to iteratively predict and remove noise, and the resulting model can synthesize high-fidelity, diverse, novel images. We find that the intermediate feature maps of the U-Net are diverse, discriminative feature representations. We propose a novel attention mechanism for pooling feature maps and further leverage this mechanism as DifFormer, a transformer feature fusion of features from different diffusion U-Net blocks and noise steps. We also develop DifFeed, a novel feedback mechanism tailored to diffusion. We find that diffusion models are better than GANs, and, with our fusion and feedback mechanisms, can compete with state-of-the-art unsupervised image representation learning methods for discriminative tasks – image classification with full and semi-supervision, transfer for fine-grained classification, object detection and segmentation, and semantic segmentation.

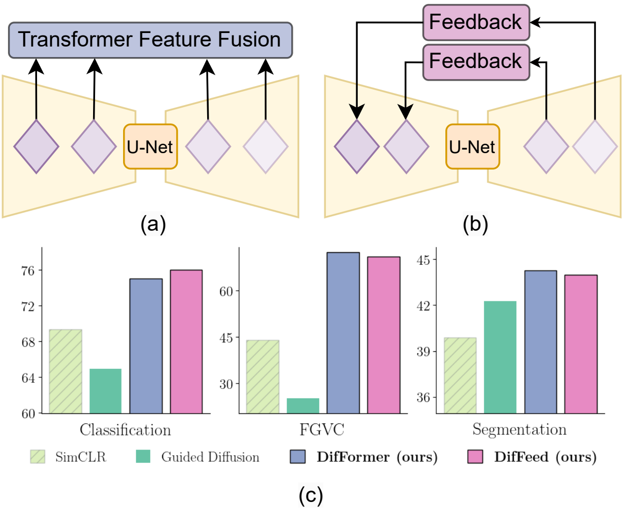

Figure 1. An overview of our method and results. We propose that diffusion models are unified self-supervised image representation learners, with impressive performance not only for generation, but also for discrimination. We improve on the promising results of out-of-the-box diffusion classifiers with our (a) fusion-based DifFormer, and (b) feedback-based DifFeed methods for intelligently utilizing the unique features of diffusion models. (c) We report exciting performances of our methods on multiple downstream benchmarks.

1. Introduction

For generative tasks, a deep learning model seeks to synthesize or edit parts of images, while for discriminative tasks, it learns to label images or parts of images. There have been works like [29] claiming that representations learned for one are not well-suited for the other because generative models rely on representations that capture low-level (pixel, texture) details as opposed to discriminative models requiring high-level (structural, object) details. But we argue that both of these can be considered complementary and intuitively should help each other because both need a semantic understanding of the underlying structure. Examples of this phenomenon can be seen in literature, *e.g.* classifier guidance (discriminative) and/or text conditioning help boost the generative performance of both Diffusion Models and StyleGANs [23, 70, 73] while generative augmentations and reconstructive objectives tend to enhance recognition capabilities [5, 37, 51, 72, 86]. Similar insights have been discovered in the field of Natural Language Processing (NLP) where the advent of masked modeling has led to methods like BERT [43] and GPT [7] which can solve both text generation as well as feature extraction at high-quality. Unified-IO [55] takes it a step further and solves diverse

*Equal contribution
†Equal contribution

tasks in both visual and language domains.

Unsupervised unified representation learning for generative and discriminative tasks, unsupervised unified representation learning in short, aims to learn general-purpose representations that can be used for various downstream tasks like image recognition, reconstruction, and synthesis [53, 81]. This overcomes the limitation of popular unsupervised computer vision methods with discriminative learning objectives, having downstream capabilities limited only to recognition tasks [10, 16, 17, 20, 21]. Such unified models can be efficiently finetuned for multiple downstream tasks, as opposed to having to pre-train large, expensive models separately for different tasks. Early examples of this paradigm jointly train a generator and an encoder network to complement each other. More recently, methods like iGPT [13] and MAE [34] have adapted masked language modeling (MLM) as masked image modeling (MIM). Unfortunately, these MIM-based models, in spite of their good classification results, underperform for generative tasks. This is perhaps because while the language-based representations used for MLM already encode meaningful concepts (*i.e.* words), MIM-based methods relies on raw pixels and hence lack this kind of structure. Masked visual token modeling (MVTM) methods rectify this lack of structure by using VQGAN tokens instead of raw pixels or patches, and achieve outstanding performance for both generation and classification [12, 53].

In this work, we tackle unified representation learning from a different point of view. We examine the embeddings of diffusion models, which iteratively denoise random noise to generate novel images without any text condition, and discover that these already have many of the properties required to be good unified representations out-of-the-box. In other words, the text-free diffusion process can be used as an unsupervised pretraining for generation and classification. Critically, the denoising process drives the diffusion models to learn semantic information necessary for discriminative tasks, without reliance on already-pretrained representations such as VQGAN tokens.

One of the main challenges with diffusion for unified representation is feature selection. In particular, the selection of noise steps and feature blocks is not trivial. We find that not only does performance drastically vary depending on block selection, noise time step, and feature pooling size, but that there is also substantial diversity of information among these features. Additionally, the optimal configuration of these features varies by dataset. So, we shift from a fixed pool and linear classification head, to a novel attention head for classification from fusion features. To incorporate the diversity among different blocks and noise steps, we propose two methods, shown at a high level in Figure 1. With the first, we extend our attention mechanism as a feature fusion head, which we call DifFormer. For the second, we propose a novel feedback mechanism for diffusion features, DifFeed, where we perform two forward passes on the diffusion model, the first as normal, and then the second where we perform learned combinations of U-Net decoder features at each encoder block.

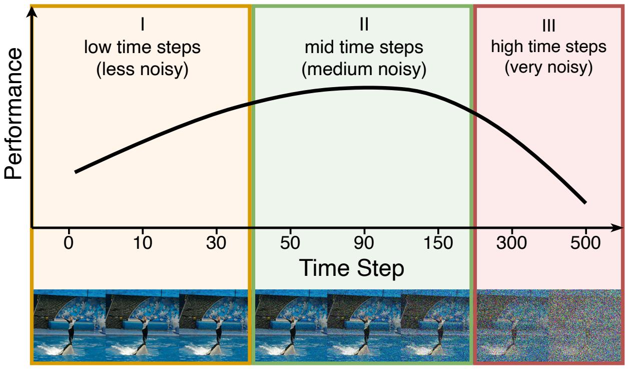

Figure 2. **Hypothesis:** Diffusion features from low (region I) and high time step (region III) are not the most discriminative and have lower performance. The best features can be found in middle time steps (region II) and vary based on tasks/datasets. At low time step, the diffusion model focuses more on stochastic details rather than structure, while at high time steps since the input is too noisy, the features may not be very semantically sound.

We also investigate the performance of out-of-the-box diffusion features, DifFormer, and DifFeed, via benchmarking and analysis. We benchmark the unconditional diffusion models from guided diffusion (GD) (a.k.a. ablated diffusion model (ADM)) on a wide array of downstream tasks, including linear probing, finetuning, and semi-supervised classification on ImageNet [22], transfer learning for fine-grained visual classification (FGVC) on multiple popular datasets [44, 47, 56, 61, 78, 79], object detection/instance segmentation on COCO [54], and semantic segmentation on ADE20K [87]. Competitive performance on this diverse set of tasks bolsters our hypothesis that unsupervised text-free diffusion features can serve as a unified representation.

In summary, our contributions are as follows:

- We demonstrate that diffusion models learn discriminative visual representations, not only for ImageNet classification but also FGVC, semi-supervised classification, object detection, and semantic segmentation.
- We find that the discriminative power of diffusion features are distributed across network blocks and noise time steps, and feature resolutions.
- We propose an attention mechanism for handling feature resolutions, and incorporate this attention mechanism as DifFormer, to combine the power of features from different blocks and noise steps, with less fixed pooling.
- We propose a novel feedback module for diffusion, DifFeed, for better performance and higher efficiency.

2. Analysis

2.1. Preliminaries

Diffusion Models Fundamentals. Diffusion models first define a forward noising process where Gaussian noise is iteratively added to an image x_0, which is sampled from the data distribution $q(x_0)$, to get a completely noised image x_T in T steps. This forward process is defined as a Markov chain with latents $x_1, x_2 \ldots, x_t, \ldots, x_{T-1}, x_T$ which represent noised images. Formally, the forward diffusion process is

$$\begin{aligned} q(x_1, \ldots x_T|x_0) &:= \prod_{t=1}^{T} q(x_t|x_{t-1}) \\ q(x_t|x_{t-1}) &:= \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}) \end{aligned} \tag{1}$$

where $\{\beta_t\}_{t=1}^{T}$ is the variance schedule and $\mathcal{N}$ is a normal distribution. As $T \rightarrow \infty$, x_T nearly is equivalent to the isotropic Gaussian distribution. With $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{i=0}^{t} \alpha_i$ one can sample a noised image x_t at diffusion step t directly from a real image x_0 using

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \epsilon \sim \mathcal{N}(0, \mathbf{I}) \tag{2}$$

The reverse diffusion process aims to reverse the forward process and sample from the posterior distribution $q(x_{t-1}|x_t)$ which depends on the entire data distribution. Doing this iteratively can denoise a completely noisy image x_T, such that one can sample from the data distribution $q(x_0)$. This is approximated using a neural network ϵ_θ as

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}\left(x_{t-1}; \frac{x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)}{\sqrt{\alpha_t}}, \Sigma_\theta(x_t, t)\right) \tag{3}$$

When p and q are interpreted as a VAE, a simplified version of the variational lower bound objective turns out to be just a mean squared error loss [35]. This can be used to train ϵ_θ which learns to approximate the Gaussian noise ϵ added to the real image x_0 in Eq. 2 as

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{x_0,t,\epsilon}[||\epsilon_\theta(x_t, t) - \epsilon||_2^2] \tag{4}$$

$\Sigma_\theta(x_t, t)$ is either kept fixed [35] or is learned using the original variational lower-bound objective [23, 59].

Diffusion Models Feature Extraction. In this work, we use the guided diffusion (GD) implementation, which uses a U-Net-style architecture with residual blocks for ϵ_θ. This implementation improves over the original [35] architecture by adding multi-head self-attention at multiple resolutions, scale-shift norm, and using BigGAN [6] residual blocks for upsampling and downsampling. We consider each of these residual blocks, residual+attention blocks, and downsampling/upsampling residual blocks as individual blocks and number them as $b \in \{1, 2, ..., 37\}$ (where $b = 19$ corresponds to the output of the mid-block of the U-Net) for the pre-trained unconditional 256×256 guided diffusion model.

Our feature extraction is parameterized with the diffusion step t and model block number b. We show an illustration of how input images vary at different time steps in Figure 2 bottom. For feature extraction of image x_0, we use Eq. 2 to get noised image x_t. In the forward pass through the network $\epsilon_\theta(x_t, t)$, we use the activation after the block number b as our feature vector $f_\theta(x_0, t, b)$.

2.2. Our Key Findings

We hypothesize that early noise steps and middle U-Net blocks may provide ideal features for classification in a setting where only a single block number and noise time step may be used (Figure 2), as intuited in existing literature [49, 81]. First, we seek to verify that this phenomenon holds for larger images. Additionally, larger images naturally have larger feature maps, introducing an added axis which much be accounted for: pooling. So, we explore the suitability of features for classification (by learning only a linear layer on top of a frozen U-Net backbone) on the axis of block number, noise time step, and feature pooling size, for images from larger, more realistic datasets, starting with ImageNet. For full details on settings for training and inference, see Section 4. Figure 3 shows that, as expected, early noise steps (see Figure 2) and middle block numbers tend to work well. Furthermore, pooling to larger features is not strictly better, as the best results for ImageNet use the second smallest features.

We do not stop our exploration at ImageNet. We show a similar ablation for the Caltech Birds (CUB) in Figure 4. Critically, we find a shift in optimal block numbers, time steps, and feature sizes, where some of these prefer earlier blocks, later time steps, and larger feature sizes. Nevertheless, our claim in Figure 2 still holds as a general principle.

[81] provide some heuristics for choosing these, per-dataset, in a label-free manner, however, this still requires substantial intervention to adapt to each dataset. Furthermore, the selection of a single block or time step is somewhat restrictive. We thus investigate these representations further, this time in the form of Centered Kernel Alignment (CKA) [46], to judge, based on their similarities to each other, which set(s) we might use for varying tasks.

We use linear centered kernel alignment (CKA) to find the degree of similarity between the representations of different blocks of the diffusion model. Following conventions from prior work that use samples for CKA [31, 80], we use the 2,500 image test set of ImageNet-50 [76], a selection of 50 classes of ImageNet. These results, shown in Figure 5, provide clues that the representations are quite diverse. With the block number and time step comparisons, we see that diffusion U-Net features differ greatly from each

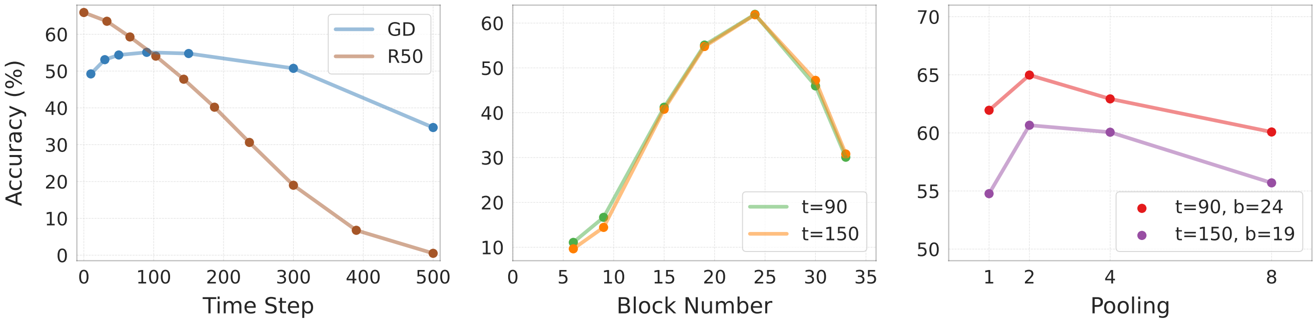

Figure 3. Ablations on ImageNet (1000 classes) with varying time steps, block numbers, and pooling size, for a linear classification head on frozen features. We find the model is least sensitive to pooling, and most sensitive to block number, although there is also a steep drop-off in performance as inputs and predictions become noisier. We further provide ResNet-50's (R50) performance over noisy time step images for comparison.

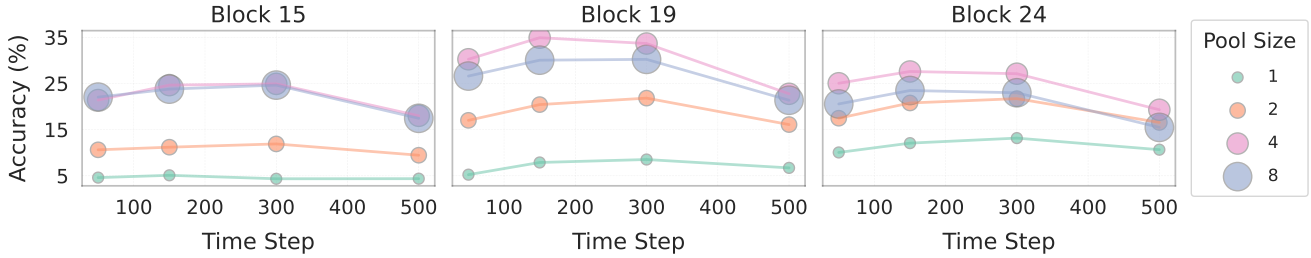

Figure 4. FGVC feature extraction analysis. We show accuracy for different block numbers, time steps, and pooling sizes. Block 19 is superior for FGVC, in contrast to ImageNet where 24 was ideal.

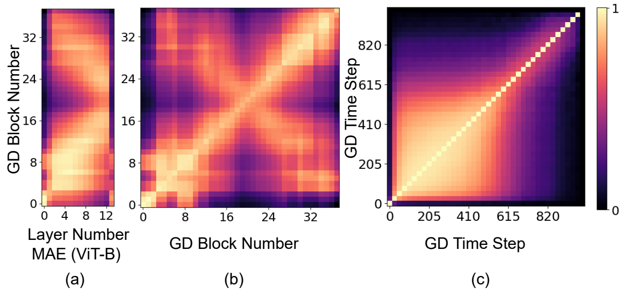

Figure 5. Feature representation comparisons via centered kernel alignment (CKA). (a) Similarity of diffusion U-Net features across blocks at $t = 90$ with features from MAE (ViT-B) layers. (b) Similarity across blocks of the diffusion U-Net at $t = 90$. (c) Similarity across timesteps of features from U-Net block $b = 24$.

other depending on these two crucial settings. Furthermore, the comparison with ViT features, particularly comparing the last, most discriminative layer of the ViT with the layers near the U-Net bottleneck ($b = 19$) suggests there is a varietal of semantic information distributed across the features at varying blocks and noise steps. We hypothesize that ensembling these features is the key for unlocking the power of diffusion models as representation learners. Thus, we propose the feature attention, fusion, and feedback methods detailed in the following sections.

3. Our Proposed Feature Fusion

3.1. Attention-based Classification Head

The two most common methods for evaluating the effectiveness of self-supervised pre-training for classification are linear probing and finetuning. The first involves learning a linear head on some frozen backbone to predict a class label; the second is the same except the backbone is not frozen. Since the diffusion U-Net is a convolutional architecture, we must use a combination of pooling and flattening to yield a feature vector representation for each image. However, as we note in Section 2, the selection of these pools is a non-trivial detail. So, we propose an Attention head to act as a learnable pooling mechanism.

Specifically, as shown in Figure 6 (a), we first use a tokenizer, consisting of adaptive average pooling, layer normalization, and 1×1 convolution, to reduce the selected feature map $f_\theta(x_0, t, b)$ into $N_b \times N_b \times 1024$, where N_b represents feature map size after pooling. We adopt a large pool size of 16 and only apply pooling when the feature map size is larger than this. We then flatten the feature map to generate N_b^2 tokens and append a CLS token before providing it as an input to our Transformer layer. We follow the standard Transformer architecture consisting of layer normalization and QKV-Attenton. Finally, the CLS token is extracted and used for classification by a linear layer.

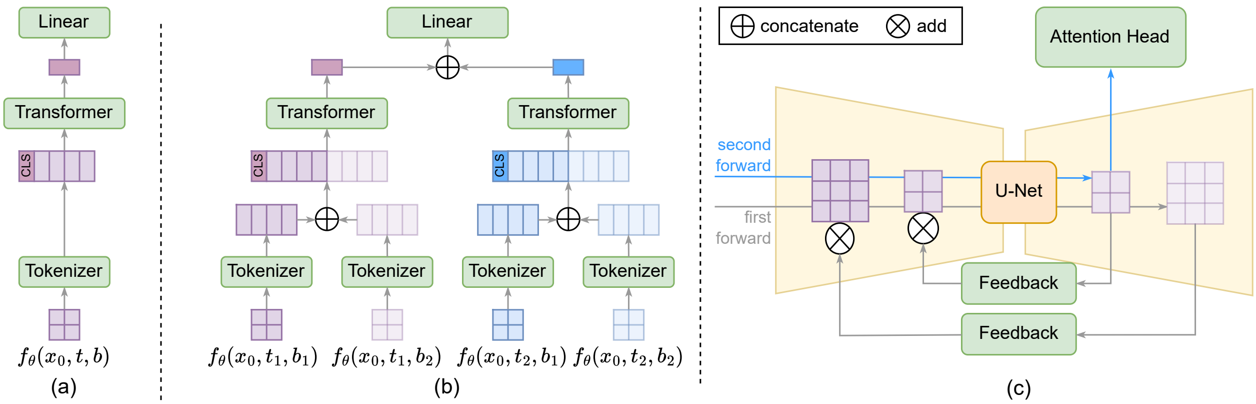

Figure 6. Architecture of 3 proposed methods for utilizing diffusion U-Net features. (a) **Attention head** takes a feature $f_\theta(x_0, t, b)$ of image x_0, diffusion time step t and U-Net block number b, tokenizes it, adds a CLS token, passes it through a Transformer, and finally feeds the CLS token to a linear layer. (b) **DifFormer**, similar to Attention head, first tokenizes features from different blocks of the same timestep, concatenates them before feeding them to a Transformer. The CLS tokens from the Transformer outputs from various timesteps are concatenated before feeding it to a final linear layer. (Here, features of $\mathcal{T} = \{t_1, t_2\}, \mathcal{B} = \{b_1, b_2\}$ are selected.) (c) **DifFeed**, after the first forward pass extracts the decoder features, which are then passed through a feedback network (a convolution layer with normalization and activation) and fed back into corresponding encoder blocks. In the second forward pass, the feedback is added to the encoder blocks to refine and fuse features from decoder layers with those of the encoder, finally to extract the features from a specific decoder block, *e.g.* $\mathcal{B} = \{b_i\}$, which is passed to an Attention head (a).

3.2. DifFormer: Transformer Feature Fusion

To consolidate features $f_\theta(x_0, t, b)$ from multiple times $t \in \mathcal{T} \subset \{1, 2, \cdots, T\}$ and blocks $b \in \mathcal{B} \subset \{1, 2, ..., 37\}$, we propose an extension of our Attention head, as shown in Figure 6 (b). After the tokenizer, we concatenate all tokens from the same noise step, resulting in a set of $\sum_{b\in\mathcal{B}} N_b^2$ tokens for each time $t \in \mathcal{T}$. Each set is processed by a Transformer layer. Finally, all CLS tokens from multiple time steps are concatenated into a $1024 \times |\mathcal{T}|$ dimensional vector which is then input to a linear layer. Tokenizer and Transformer use the same parameters across all noise steps.

3.3. DifFeed: Feedback Feature Fusion

To enable more performant fusion of features from various blocks, we propose a dynamic feedback mechanism, shown in Figure 6 (c) which takes advantage of the U-Net architecture to refine diffusion features for the specific downstream task. Specifically, we decouple the feature extraction process into two forward passes. In the first forward pass, we store the feature maps generated by the decoder at a selected set of blocks. We then feed these features to a light-weight feedback network, which has unique layers consisting of 1×1 convolution, batch normalization, and ReLU for each selected decoder block. These feedback layers learn to map decoder features to a suitable space for adding them to the corresponding encoder features. Our key intuition behind this design is that, as we find in Section 2, the decoder features contain important semantic information, unique to each block, that can be used to enrich the encoder image representations. To finally obtain the features for the downstream task, we perform a second forward pass and add the feedback features to the encoder features based on various feedback strategies (see Section 4.1.1). We use the attention head on top of one of the second forward pass features.

Table 1. Main results. We compare self-supervised and unified learners in terms of classification and generation at resolution 256×256.

Method	Type	Accuracy	FID
SimCLR†	Self-Supervised	69.3%	n/a
SwAV‡	Self-Supervised	75.3%	n/a
MAE$^\flat$	Self-Supervised	73.5%	n/a
BigBiGAN*	Unified	60.8%	28.54
MAGE$^\natural$	Unified	**78.9%**	**9.10**
U-Net Encoder	Supervised	64.3%	n/a
GD	Unified	64.9%	26.21§
Attention	Unified	74.6%	26.21§
DifFormer	Unified	76.0%	26.21§
DifFeed	Unified	77.0%	26.21§

†Result from [16]. ‡Result from [10]. $^\flat$Result from [34]. $^\natural$Results from [53]. §Results from [23]. *Results from [24]. BigBiGAN's best FID is at generator resolution 128.

4. Experiments

In Section 4.1, we compare our diffusion extraction to baselines and competing unified representation methods. We provide ablations in Section 4.1.1 to justify key design choices for Attention head, DifFormer, and DifFeed. We benchmark diffusion for key downstream tasks inSection 4.3, Section 4.4, and Section 4.5.

Experiment Details. Unless otherwise specified, we use the unconditional ADM U-Net architecture from Guided Diffusion [23] with total timesteps $T = 1000$. We use the 256×256 checkpoint; thus we resize all inputs to this size and use center-crop and flipping for data augmentation. We use $f_\theta(x_0, t = 150, b = 24)$ as a default feature map. We use cross entropy loss with an Adam optimizer [45], follow the VISSL protocol for linear probing – 28 epochs, with StepLR at 0.1 gamma every 7 epochs. However, we do not use random cropping or batch norm. For hardware, most of our experiments are run on 4 NVIDIA RTX A5000 GPUs.

Datasets. The dataset we use for our main result is ImageNet-1k [22] which contains 1.3M training images over 1000 classes. Additionally, we run ablations and similar explorations on Caltech-UCSD (CUB) [79] and ImageNet-1k with 1% label (IN-1%). CUB is a fine-grained visual classification dataset consisting of 200 classes. IN-1% uses all the training images of ImageNet-1k but only 1% of the labels. Please see the appendix for details.

4.1. Main Results: ImageNet Classification

First, we show the linear probing performance of diffusion in Table 1. For guided diffusion with linear head (GD), we use a 2 × 2 adaptive average pooling to reduce the spatial dimension. As a baseline, we compare to the pre-trained guided diffusion classifier (used for classfier-guidance [23]), since it uses the same U-Net Encoder. We also offer a comparison to other unified models: BigBiGAN [24] and MAGE [53] as well as self-supervised models like SimCLR [15] and SwAV [10]. GD outperforms BigBiGAN in terms of both generation and classification, especially when BigBiGAN is forced to handle the higher resolution, 256×256 images. Hence, diffusion models beat GANs for image classification (and generation). This is not yet state-of-the-art compared to classification-only models, with a gap of over 10% top-1 accuracy, or compared to the powerful unified MAGE model.

As described previously, we propose several approaches to deal with the large spatial and channel dimensions of U-Net representations as well as to fuse features from various blocks. For these proposals, we use settings selected via the ablations described in Section 4.1.1. In Table 1, we try the best-performing Attention head on ImageNet-1k and find it significantly outperforms linear probe. This suggests the classification head is an important mechanism for extracting useful representations from diffusion models, and it could be extended to other generative models. Similarly, using better feature fusion mechanisms like DifFormer and DifFeed, with frozen backbones, can lead to significant improvement in performance to the level that it is comparable to MAGE, while better than SimCLR and SwAV (see Table 1).

Table 2. Attention Head Ablation. We compare the effect of the number of layers on top-1 accuracy and on the number of Attention head network parameters. We freeze the U-Net and train the Attention heads for 15 epochs.

# Layers	Accuracy		# Params
	IN-1%	CUB	
1	40.4%	39.4%	15.2M
2	45.7%	47.4%	27.8M
4	**47.9%**	**52.0%**	53.0M

Table 3. DifFormer Ablation. We compare various combinations of time and block fusion strategies on top-1 accuracy and number of forward passes required through the U-Net backbone.

$\mathcal{T}$	$\mathcal{B}$	Accuracy		# Forwards
		IN-1%	CUB	
$\{150\}$	$\{24\}$	45.7%	47.4%	1
$\{90, 150, 300\}$	$\{24\}$	49.0%	54.1%	3
$\{150\}$	$\{19, 24, 30\}$	43.2%	48.8%	1
$\{90, 150, 300\}$	$\{19, 24, 30\}$	**50.4%**	**56.8%**	3

Table 4. DifFeed Ablation. We compare various feedback block sampling strategies on top-1 accuracy, on the number of forward passes required through the U-Net backbone, and on the number of feedback network parameters.

Block Sampling	Accuracy		# Forwards	# Params
	IN-1%	CUB		
all	31.1%	57.2%	2	7.6M
bottleneck	**51.5%**	**71.0%**	2	2.8M
windowed	49.4%	67.1%	2	5.2M
multi-scale	49.4%	66.0%	2	5.2M

4.1.1 Ablations

Here, we give results of ablation study and show the best settings for our Attention head, DifFormer, and DifFeed. Note that all ablation results are obtained with frozen backbones. From Table 2, we find that using multiple Transformer layers in Attention head leads to better accuracy. Considering the increased number of parameters, we adopt two Transformer layers in our Attention head. Table 3 compares the effect of various combinations of time and block in DifFormer. We find that noise step fusion substantially improves the accuracy, and combining it with block fusion increases the accuracy further. Thus we adopt $\mathcal{T} = \{90, 150, 300\}$ and $\mathcal{B} = \{19, 24, 30\}$ in DifFormer.

In Table 4, we examine four block selection strategies for generating feedback features that can be fed to the encoder layers. For all experiments, we use $t = 150$ and set $b = 24$ for extracting the final feature. The naive "all" strategy uses all decoder blocks to generate feedback features and maps them to symmetrically corresponding encoder blocks. The rest of the proposed sampling methods aim to find an optimal sparser selection of blocks. Our "bottleneck" strategy uses only the bottleneck blocks $\mathcal{B} = \{21, 24, 27, 30, 33, 36\}$ while "windowed" strategy

Table 5. Semi-supervised results. We give accuracy results on ImageNet-1k with 1% labels and 10% labels for fine-tuning.

Method	1% label	10% label
SimCLR†	48.3%	65.6%
SwAV‡	**53.9%**	**70.2%**
GD	46.7%	64.4%
DifFormer	51.8%	66.2%
DifFeed	52.9 %	66.6%

†Results from [16]. ‡Results from [10].

only generates feedback from the first 5 blocks of the decoder. Our "multi-scale" strategy uses the same blocks as "window" strategy but feeds them back to a singular block in the encoder. We find that "bottleneck" strategy works the best. Thus, we adopt "bottleneck" strategy in DifFeed.

4.2. Semi-Supervised Classification

In this subsection, we evaluate the power of our proposed methods on datasets with limited labels. Following SimCLR [16], we sample 1% and 10% of the labeled ImageNet-1k in a class-balanced way and call them IN-1% and IN-10% respectively. Table 5 shows the results of SimLR, SwAV, and our proposed models, with fine-tuned backbones. We observe that GD has worse performance compared to SimCLR but using DifFeed and DifFeed for feature fusion improves the performance both on 1% and 10%. We get top-1 accuracy on the ImageNet validation set better than SimCLR and slightly worse than SwAV.

4.3. Fine-grained Visual Classification (FGVC)

Here, we give results for applying our method in the transfer setting to the datasets defined in the appendix. We use both standard linear probing, as well as our proposed Attention head, DifFormer, and DifFeed with frozen backbones. We show these results in Figure 7. We observe a gap in performance between GD and SimCLR. However, all our methods - Attention head, DifFormer, and DifFeed, cover this performance gap and outperform self-supervised methods like SimCLR and SwAV on all datasets except Flowers.

4.4. Semantic Segmentation

To evaluate the power of GD features and DifFeed on dense prediction task we choose ADE20K semantic segmentation using UperNet [82] following MAE [34]. Note that the UperNet setup already has a component that fuses features from various spatial resolutions and hence we don't use DifFormer here. We use batch size 2 due to resource constraints (instead of 16) in all the models we train. In Table 6, we observe that using only one block of GD for all UperNet inputs performs comparably to DreamTeacher (which uses distilled GD features), while better than other ResNet-50 based model supervised, SimCLR [16], and SwAV [10]. When multiple GD decoder blocks are used we observe a 2% boost. We observe a similar boost even with DifFeed, where we extract features from $\mathcal{B} = \{33, 30, 27, 24\}$ at $t = 90$ for UperNet inputs. Even though these performances are better than MAE (ViT-B), MAE (ViT-L) has a superior performance. More training details are present in the appendix.

Table 6. Semantic segmentation results. ADE20K [87] results using UperNet [82] finetuned for 16K iterations. We report mean IoU at a single scale.

Method	mIoU
Supervised	40.9%$^\sharp$
SimCLR	39.9%$^\sharp$
SwAV	41.2%$^\sharp$
DreamTeacher	42.5%$^\sharp$
MAE (ViT-B)	40.8%
MAE (ViT-L)	**45.8%**
GD ($t = 90, b = 24$)	42.3%
GD ($t = 90, \mathcal{B} = \{33, 30, 27, 24\}$)	44.3%
DifFeed	44.0 %

$^\sharp$ Results from [52].

4.5. Object Detection and Segmentation

To continue our benchmark of diffusion model features, we evaluate them for object detection and segmentation. Specifically, borrowing the settings of [34], we compare guided diffusion networks, with features taken at time step $t = 75$ and block number $b = 24$, to MAE ViT-L [26] and supervised ResNet-50 [32] for AP on COCO in Figure 8. While worse than MAE, this is still a promising result for diffusion, since with only due diligence hyperparameter tuning, it is already able to outperform the ResNet-based MaskR-CNN for these dense prediction tasks. Due to the overall expense of training for these tasks, we leave further explorations to future work, and strongly encourage the community to examine this direction further as a promising application for diffusion as unified representation. For more details on settings and ablations, refer to the appendix.

5. Related Work

Generative Models. Generative models learn to sample novel images from a data distribution. Generative Adversarial Networks (GANs) [6, 28, 38–42, 73] are a class of such models that are trained by optimizing a min-max game between a generator, which synthesises images and a discriminator, which classifies it as real or fake. Diffusion denoising probabilistic models (DDPM) [35], a.k.a. diffusion models, are a class of likelihood-based generative models which learn a denoising Markov chain using variational inference. These models enjoy the benefit of having a likelihood-based objective like VAEs as well as high visual sample qual-

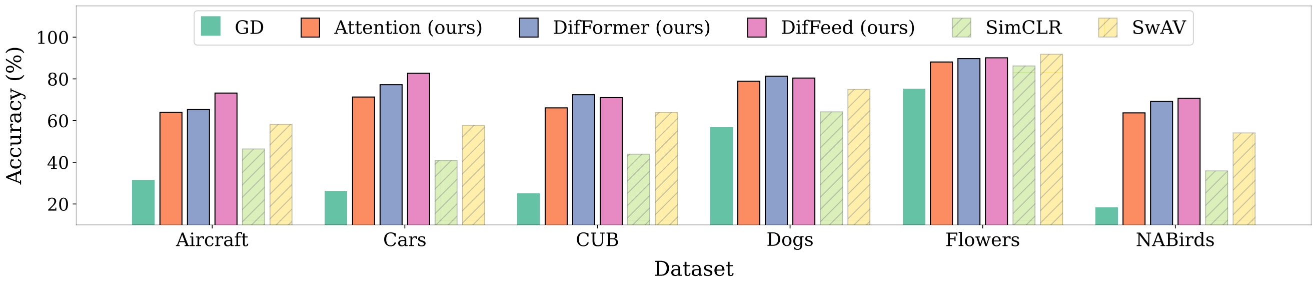

Figure 7. Fine-Grained Visual Classification (FGVC) results. We compare the diffusion baseline, GD, to our Attention head, DifFormer, and DifFeed in terms of accuracy for these popular FGVC datasets. We show that our methods are typically better than even SimCLR and SwAV, especially for Aircraft, Cars, and NABirds.

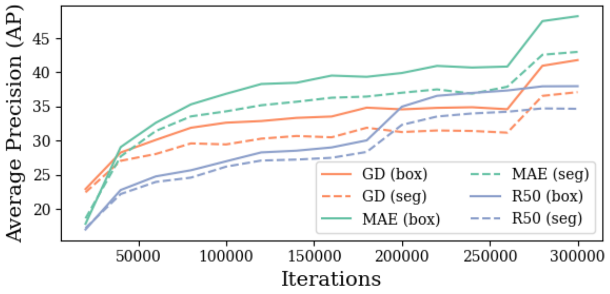

Figure 8. Detection results on COCO. Diffusion outperforms the supervised ResNet-50 [32] Mask R-CNN [33] in terms of both object detection and instance segmentation, but is surpassed by the MAE-trained ViT-L Mask R-CNN.

ity like GANs even on high variability datasets. Diffusion models have proven to produce high-quality images [23], beating previous SOTA generative models [6, 69] for FID on ImageNet [22] and have also achieved amazing results in text-to-image generation [68, 70, 71]. Application of these models is not just limited to generation but spans tasks like object detection [14], and image segmentation [8]. We continue these lines of application with our comprehensive study and proposed methods.

Discriminative Models. Discriminative models extract useful information from images that can then be used to solve downstream recognition tasks. Early self-supervised learning (SSL) methods attempt to learn image representations by training neural network backbones with partially degraded inputs to predict the missing information [57, 62, 65, 85]. More recently, many approaches revolve around a contrastive loss objective, min-maxing distance between positive and negative pairs [15, 17, 20, 21, 75, 84]. On the other hand, some methods operates without negative pairs [4, 18, 29], and others use clustering-style objectives [9, 10]. MAE [34] and iBOT [88] train an autoencoder via masked image modeling [1, 2, 36]. DINO [11] uses self-supervised knowledge distillation between various image views in Visual Transformers [26]. With all the recent advances, the latest SSL methods surpass supervised methods on many key discriminative baselines [50, 63, 64, 89] but not generation.

Unified Models. Works like [19, 25, 27, 60] leverage the unsupervised nature of GANs to learn good image representations. BiGAN [25], ALI [27], ALAE [66], and BigBiGAN [24] do joint Encoder-Generator training with a discriminator on image-latent pairs. PatchVAE [30] improves performance by encouraging the model to learn good mid-level patch representations. ViT-VQGAN [83] learns to predict VQGAN tokens autoregressively in a rasterized fashion. MAGE [53], which uses a variable masking ratio during training and iterative decoding for inference, is the first method to achieve both high-quality unconditional image generation and good classification results.

Diffusion Features. Recently, the use of diffusion models' intermediate activation features for various non-generative tasks is also gaining popularity. Works like [3, 52, 67] focus on segmentation. Diffusion Classifier [49], which solves zero-shot classification using Stable Diffusion (SD) noise prediction, also provides the performance of training a ResNet-50 using SD features as their baseline comparison for ImageNet classification (see appendix for our linear probing results with SD). DIFT [74] employs SD features for semantic correspondence. DDAE [81] also finds that using an unconditional diffusion model's features yields competitive results on CIFAR10 [48] and TinyImageNet [58]. However, we observe that this competitiveness does not hold anymore for datasets with higher variability.

6. Conclusion

In this paper, we present an approach for using the representations learned by diffusion models for recognition tasks. This re-positions diffusion models as potential state-of-the-art unified self-supervised representation learners. We propose the novel DifFormer and DifFeed methods for learned feature extraction, pooling, and synthesis. We demonstrate promising results for ImageNet classification, semi-supervised classification, FGVC transfer learning, object detection, and semantic segmentation. Finally, to answer our titular question – yes, text-free diffusion models do learn discriminative visual features and we suggest that with such promising out-of-the-box capabilities, diffusion is a prime candidate for unified representation learning, and with our initial benchmark results, hope to encourage others to explore these directions.

References

[1] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Michael Rabbat, and Nicolas Ballas. Masked siamese networks for label-efficient learning, 2022. 8

[2] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers, 2022. 8

[3] Dmitry Baranchuk, Andrey Voynov, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. In *International Conference on Learning Representations*, 2021. 8

[4] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *ArXiv*, abs/2105.04906, 2021. 8

[5] Victor Besnier, Himalaya Jain, Andrei Bursuc, Matthieu Cord, and Patrick Pérez. This dataset does not exist: training models from generated images. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2020. 1

[6] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 3, 7, 8

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1

[8] Ryan Burgert, Kanchana Ranasinghe, Xiang Li, and Michael S Ryoo. Peekaboo: Text to image diffusion models are zero-shot segmentors. *arXiv preprint arXiv:2211.13224*, 2022. 8

[9] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018. 8

[10] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020. 2, 5, 6, 7, 8

[11] 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 International Conference on Computer Vision (ICCV)*, 2021. 8

[12] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. Maskgit: Masked generative image transformer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[13] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International conference on machine learning*, pages 1691–1703. PMLR, 2020. 2

[14] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection. *arXiv preprint arXiv:2211.09788*, 2022. 8

[15] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 6, 8

[16] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1597–1607. PMLR, 2020. 2, 5, 7, 1, 3

[17] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020. 2, 8

[18] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021. 8

[19] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016. 8

[20] Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *CoRR*, abs/2003.04297, 2020. 2, 8

[21] Xinlei Chen*, Saining Xie*, and Kaiming He. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021. 2, 8

[22] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2, 6, 8, 1

[23] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021. 1, 3, 5, 6, 8

[24] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. *Advances in neural information processing systems*, 32, 2019. 5, 6, 8

[25] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016. 8

[26] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 7, 8

[27] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016. 8

[28] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 7

[29] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. *CoRR*, abs/2006.07733, 2020. 1, 8

[30] Kamal Gupta, Saurabh Singh, and Abhinav Shrivastava. Patchvae: Learning local latent codes for recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 8

[31] Matthew Gwilliam and Abhinav Shrivastava. Beyond supervised vs. unsupervised: Representative benchmarking and analysis of image representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9642–9652, 2022. 3

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 7, 8

[33] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018. 8

[34] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. *CoRR*, abs/2111.06377, 2021. 2, 5, 7, 8, 3

[35] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 3, 7

[36] Zhicheng Huang, Xiaojie Jin, Chengze Lu, Qibin Hou, Ming-Ming Cheng, Dongmei Fu, Xiaohui Shen, and Jiashi Feng. Contrastive masked autoencoders are stronger vision learners, 2022. 8

[37] Ali Jahanian, Xavier Puig, Yonglong Tian, and Phillip Isola. Generative models as a data source for multiview representation learning. In *International Conference on Learning Representations*, 2022. 1

[38] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 7

[39] 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 (CVPR)*, 2019.

[40] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems*, 33:12104–12114, 2020.

[41] 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.

[42] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863, 2021. 7

[43] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, page 2, 2019. 1

[44] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, 2011. 2, 1

[45] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 6

[46] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019. 3

[47] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013. 2, 1

[48] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 8

[49] Alexander Cong Li, Mihir Prabhudesai, Shivam Duggal, Ellis Langham Brown, and Deepak Pathak. Your diffusion model is secretly a zero-shot classifier. In *ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2023. 3, 8, 2

[50] Chunyuan Li, Jianwei Yang, Pengchuan Zhang, Mei Gao, Bin Xiao, Xiyang Dai, Lu Yuan, and Jianfeng Gao. Efficient self-supervised vision transformers for representation learning, 2022. 8

[51] Daiqing Li, Huan Ling, Seung Wook Kim, Karsten Kreis, Sanja Fidler, and Antonio Torralba. Bigdatasetgan: Synthesizing imagenet with pixel-wise annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21330–21340, 2022. 1

[52] Daiqing Li, Huan Ling, Amlan Kar, David Acuna, Seung Wook Kim, Karsten Kreis, Antonio Torralba, and Sanja Fidler. Dreamteacher: Pretraining image backbones with deep generative models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16698–16708, 2023. 7, 8, 1

[53] Tianhong Li, Huiwen Chang, Shlok Kumar Mishra, Han Zhang, Dina Katabi, and Dilip Krishnan. Mage: Masked generative encoder to unify representation learning and image synthesis, 2022. 2, 5, 6, 8

[54] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 2, 1

[55] Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. UNIFIED-IO: A unified model for vision, language, and multi-modal tasks. In *The Eleventh International Conference on Learning Representations*, 2023. 1

[56] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013. 2, 1

[57] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020. 8

[58] Mohammed Ali mnmoustafa. Tiny imagenet, 2017. 8

[59] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 3

[60] Weili Nie, Tero Karras, Animesh Garg, Shoubhik Debnath, Anjul Patney, Ankit B Patel, and Anima Anandkumar. Semi-supervised stylegan for disentanglement learning. In *Proceedings of the 37th International Conference on Machine Learning*, pages 7360–7369, 2020. 8

[61] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, 2008. 2, 1

[62] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016. 8

[63] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 8

[64] Bo Pang, Yifan Zhang, Yaoyi Li, Jia Cai, and Cewu Lu. Unsupervised visual representation learning by synchronous momentum grouping, 2022. 8

[65] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016. 8

[66] Stanislav Pidhorskyi, Donald A Adjeroh, and Gianfranco Doretto. Adversarial latent autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14104–14113, 2020. 8

[67] Koutilya Pnvr, Bharat Singh, Pallabi Ghosh, Behjat Siddiquie, and David Jacobs. Ld-znet: A latent diffusion approach for text-based image segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4157–4168, 2023. 8

[68] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 8

[69] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019. 8

[70] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 1, 8

[71] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 8

[72] Mert Bulent Sariyildiz, Karteek Alahari, Diane Larlus, and Yannis Kalantidis. Fake it till you make it: Learning transferable representations from synthetic imagenet clones. In *CVPR 2023-IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–11, 2023. 1

[73] Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. 2022. 1, 7

[74] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. *arXiv preprint arXiv:2306.03881*, 2023. 8

[75] Nenad Tomasev, Ioana Bica, Brian McWilliams, Lars Buesing, Razvan Pascanu, Charles Blundell, and Jovana Mitrovic. Pushing the limits of self-supervised resnets: Can we outperform supervised learning without labels on imagenet?, 2022. 8

[76] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *Proceedings of the European Conference on Computer Vision*, 2020. 3, 1

[77] Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 595–604, 2015. 1

[78] Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[79] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 2, 6, 1

[80] Matthew Walmer, Saksham Suri, Kamal Gupta, and Abhinav Shrivastava. Teaching matters: Investigating the role of supervision in vision transformers, 2023. 3

[81] Weilai Xiang, Hongyu Yang, Di Huang, and Yunhong Wang. Denoising diffusion autoencoders are unified self-supervised learners. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15802–15812, 2023. 2, 3, 8

[82] 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. 7

[83] 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. In *International Conference on Learning Representations*, 2021. 8

[84] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stephane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *Proceedings of the 38th International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021. 8

[85] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016. 8

[86] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10145–10155, 2021. 1

[87] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 2, 7, 1

[88] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer, 2022. 8

[89] Pan Zhou, Yichen Zhou, Chenyang Si, Weihao Yu, Teck Khim Ng, and Shuicheng Yan. Mugs: A multi-granular self-supervised learning framework, 2022. 8

Do text-free diffusion models learn discriminative visual representations?

Supplementary Material

7. Training Details Details

In this section, we provide training details for our benchmarking over various tasks. We provide our code in the supplementary bundle. Please refer to the code for exact details of our methods and their training.

7.1. Datasets and Parameter counts

Table 7. Classification dataset details.

Dataset	#Cls	#Train	#Test
Aircraft [56]	100	6,667	3,333
Cars [47]	196	8,144	8,041
CUB [79]	200	5,994	5,794
Dogs [44]	120	12,000	8,580
Flowers [61]	102	2,040	6,149
NABirds [77]	555	23,929	24,633
ImageNet-1k [22]	1000	1.28M	50,000
ImageNet 1% labels [16]	1000	12811	50,000
ImageNet 10% labels [16]	1000	128,116	50,000
ImageNet-50 [76]	50	64,274	2,500

Table 8. Parameter counts of major unified unsupervised representation learning methods. For each, we consider the whole system, not just the encoding network.

Method	# Params
BigBiGAN	502M
MAGE	439M
GD	553M
DifFormer	585M
DifFeed	583M

For classification, we use the datasets shown in Table 7. For object detection and semantic segmentation, COCO [54] dataset was used which consists of 118K training and 5K validation images containing 1.5M object instances over 80 object categories. For semantic segmentation, we use the SceneParse150 benchmark of ADE20K [87] dataset which contains 20,210 training and 3,169 validation images having 150 categories which occupy the most pixels in the images.

Please find the details of the parameter counts of various unsupervised unified representation learners in Table 8.

7.2. Object Detection Details

For all object detection results in Figure 8, we train on the equal number of epochs – 10. We scale the step learning rate schedulers for ViT and ResNet accordingly, and borrow the ViT settings for the guided diffusion backbone. We use a batch size of 4 for both ViT and diffusion (1 per GPU), and choose learning rate based on a hyperparameter search for both models. With ResNet, we are able to use the original batch size (16).

7.3. Semantic Segmentation Details

The official semantic segmentation code for MAE has not been made available. Hence, we use mae_segmentation repository (https://github.com/implus/mae_segmentation) for the settings for MAE (ViT-B) on MMSegmentation framework which reproduces the results from the MAE paper. It uses 2 conv-transpose layers, 1 conv-transpose layer, identity, a maxpool layer respectively to convert their embeddings from different layers to UperNet inputs. We use batch size 2 due to resource constraints (instead of 16) in all the models we train. We tried tuning the learning rate for updated batch size and the default worked the best. For MAE (ViT-L), we update the hyperparameters according to ViT-Det COCO Object Detection following a comment by the MAE authors comment by the MAE authors and increase the UperNet channel dimension to 1024 instead of 768 for MAE (ViT-B). For GD, we convert the GD U-Net's features to 1024 using 1×1 conv layer and then use bilinear interpolation to get the shapes of features identical to that of MAE (ViT-L)'s inputs to the UperNet. We also performed learning rate and block-choice hyperparameter tuning. We finally use the learning rate 2e-5. Please note that the ResNet50 backbone results are picked from DreamTeacher [52] which might have different hyperparameter settings given their code is unavailable (e.g. they use the default batch size 16).

8. Additional Results

8.1. K-Nearest Neighbor Results

Table 9. kNN Results on ImageNet-1k.

b	t	pool	Top1@20	Top5@20	Top1@200	Top5@200
19	10	1	41.14	62.49	39.04	65.40
19	30	1	46.30	68.47	43.65	70.94
19	50	1	47.89	70.10	45.06	72.36
19	90	1	49.04	71.40	46.17	73.65
19	150	1	48.78	71.27	46.29	73.76
19	300	1	45.18	67.28	43.04	70.09
19	500	1	29.96	48.64	29.68	53.12
24	90	1	48.34	70.02	46.22	72.57

Apart from linear probing results in the main paper we provide kNN results in Table 9. We observe the trends exactly match that in Figure 3 from the main paper. There are minor deviations, the most significant being that $t = 150$, $b = 24$ is now the best setting by a slightly larger margin (1%), but the trends support the same conclusion we draw w.r.t. feature selection.

8.2. ImageNet-1k Fine-tuning

Table 10. ImageNet-1k classification top-1 accuracy results for full-finetuning.

Method	Full
Supervised	78.8%
SimCLR	76.0%
Supervised	82.5%
MAE	**84.9%**
MAGE	84.3%
GD (L)	73.50%

Table 10 compares the finetuning results of various SSL and unified methods with our DifFormer and DifFeed on ImageNet-1k. For finetuning DifFeed, we enable gradient calculation only on the second forward pass.

8.3. Stable Diffusion features

Table 11. Stable Diffusion linear probe results.

Condition	b	Size	Accuracy
Null Text	18	512	64.67%
Null Text	15	512	55.77%
Null Text	18	256	41.37%
Learnable	18	512	**65.18%**
Guided Diffusion	24	256	**61.86 %**

As a proof of concept, we also test our hypothesis on Stable Diffusion (SD) features trained on ImageNet-1k for 15 epochs. Note that Stable Diffusion has been trained on paired text-image data, and hence has indirectly seen a lot of labels in the form of text captions, and hence can not be called unsupervised. Nonetheless, we show the results in Table 11) as an alternative diffusion model to see the generalizability of diffusion-based pertaining. For text conditioning we try both null text as well as learnable text embedding. We see that SD U-Net's mid-block $b = 15$ under-performs $b = 18$ which is consistent with GD. Further, we see that a larger image size and learnable condition are better. Our results are also consistent with SD Features baseline in [49] at a comparable setting, but its performance can be further improved by optimizing the choice of block.

9. Ablations

9.1. Attention Head Alternatives

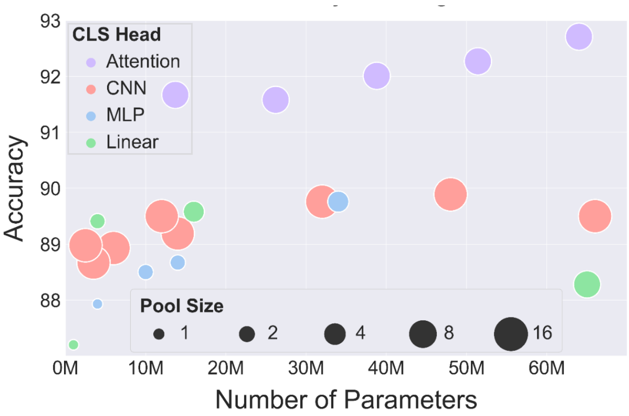

Figure 9. Head results. We show the results of explorations for various classification heads – Linear, MLP, CNN, Attention, trained on frozen features for ImageNet-50, computed at block number 24 and noise time step 90.

Table 12. Linear and MLP results. For linear, 1k, 4k, 16k, and 65k indicate the size of the feature after pooling and flattening. For MLP, the first number is the size of the feature after pooling and flattening, and the succeeding numbers are hidden sizes of layers before the last (classification) layer.

Head	Params	Accuracy
Linear-1k	1M	87.20%
Linear-4k	4M	89.41%
Linear-16k	16M	89.58%
Linear-65k	65M	88.28%
MLP-1k-2k	4M	87.93%
MLP-4k-2k	10M	88.50%
MLP-4k-2k-2k	14M	88.67%
MLP-16k-2k	34M	89.76%

Table 13. CNN head results. Channel sizes are separated by dashes. The first is the channel size of the input feature maps, the next is the output channels from the first convolution, and the last is the output dimension of the second convolution. These are treated as hyperparameters. For more detail, see our code.

Head	Params	Accuracy
CNN-1k-256-256	2.5M	88.98%
CNN-1k-512-256	3.5M	88.67%
CNN-1k-1k-256	6M	88.93%
CNN-1k-1k-1k	12M	89.50%
CNN-1k-2k-512	14M	89.19%
CNN-1k-2k-2k	32M	89.76%
CNN-1k-4k-2k	48M	89.89%
CNN-1k-4k-2.5k	66M	89.50%

Table 14. Attention head results. The hyperparameters are denoted following the dashes. The first hyperparameter is similarly the channel size of the input feature maps and the next is the number of Transformer blocks used. For fair comparison with other heads, we remove 1×1 convolution in tokenizer for this analysis.

Head	Params	Accuracy
Attention-1K-1	13.7M	91.67%
Attention-1K-2	26.2M	91.58%
Attention-1K-3	38.8M	92.01%
Attention-1K-4	51.4M	92.27%
Attention-1K-5	64.0M	92.71%

Apart from the Attention head presented in the main paper, we also experimented with other heads like Linear, MLP, and CNN of various powers on ImageNe-50 as seen in Figure 9. More information is present in Table 12, Table 13, and Table 14. We show exact accuracies and parameter counts. We also specify our hyperparameter selection for each head. It can be clearly seen that at the same level of parameters Attention head outperforms the other alternatives.

9.2. Feedback Feature Extraction

Table 15. DifFeed feature extraction ablation.

	b	21	24	27	30	33	36
Accuracy	CUB	65.1%	**71.0%**	70.6%	67.2%	56.6%	49.8%
	Cars	68.3%	**82.7%**	73.3%	-	-	-

We provide the ablation for various strategies for feedback mechanisms in DifFeed in Section 4.1.1 of the main paper. Here we provide our ablation for the choice of $b = 24$ for extracting the final feature in the second pass in Table 15. The results are presented for the "bottleneck" strategy of feedback.

9.3. Transformer Feature Fusion comparisons

Table 16. Transformer feature fusion results.

Method	Backbone	$\mathcal{T}$	$\mathcal{B}$	Params	Accuracy
SimCLR	ResNet-50	-	$\{4\}^{\dagger}$	27.5M	93.6%
SimCLR	ResNet-50	-	{1,2,3,4}*	30.0M	93.6%
MAE	ViT-B	-	$\{12\}^{\dagger}$	26.0M	93.6%
MAE	ViT-B	-	{9,10,11,12}*	28.4M	94.1%
Attention	GD	{150}	{24}	26.8M	91.1%
DifFormer	GD	{90,150,300}	{19,24,30}	29.3M	94.2%
DifFeed	GD	{150}	{24}	29.6M	**94.9%**

†Attention head is applied.
*Transformer feature fusion mechanism is applied.

For fair comparison in terms of the number of learnable parameters, we provide results of applying our Attention head and transformer feature fusion mechanisms on top of baselines . We number blocks in ResNet-50 as $b \in \{1, 2, 3, 4\}$ based on the feature map size and blocks in ViT-B as $b \in \{1, 2, ..., 12\}$ based on the attention layers. We fixed the backbones and trained heads on ImageNet-50 for 28 epochs and compare them against our approaches. The results for the best learning rate for each of the methods are provided in Table 16. We see that transformer feature fusion mechanism does not help SimCLR [16] but has slight improvement in MAE [34]. In the case of our methods, we see that feature fusion and feedback mechanisms provide performance boosts of more than 3% over just using Attention head and consequently outperform both the baselines with or without feature fusion. This shows the efficacy of both these mechanisms in the case of guided diffusion.