A Results of CIFAR-10

To evaluate the generalizability of DPT across different datasets, we also conduct an experiment on CIFAR-10 [19].

A.1 Baselines

For semi-supervised classification, we consider a state-of-the-art method called FreeMatch [15] as the baseline. For conditional generation, we consider a state-of-the-art method called EDM [6] as the baseline. The training configuration of EDM is variance preserving (VP) [3], as it achieves slightly better generation performance compared to the alternative configuration of variance exploding (VE) [3].

A.2 Settings

In the second stage of DPT, we generate pseudo images using the same sampling process as EDM [6]. We set the number of augmented pseudo images per class, i.e., K, to 1001 as the default value if not specified. In the third stage of DPT, we replace $\mathcal{S}$ with $\mathcal{S} \cup \mathcal{S}_2$ to re-train FreeMatch [15].

A.3 Evaluation metrics

We use the error rate on the validation set to evaluate classification performance and consider the Fréchet inception distance (FID) score [21] to evaluate generation performance.

A.4 Image Generation with Few Labels

Tab. 5 presents a quantitative comparison of DPT with state-of-the-art generative models on the CIFAR-10 generation benchmark. In particular, DPT achieves an FID of 1.81 with only *four (i.e.,* 0.08%*) labels per class*, outperforming strong supervised generative models such as StyleGAN-XL [92] and IDDPM [23], and even demonstrating competitive performance compared to the state-of-the-art supervised generative model EDM [6].

Table 5: **Image generation results on CIFAR-10 32×32.**

Method	Model	Label fraction (# labels/class)	FID-50K ↓
StyleGAN2-ADA [96]	GAN	100%	2.92
StyleGAN-XL[92]	GAN	100%	1.85
EDM [6]	Diff.	0%	1.97
DDPM [2]	Diff.	100%	3.17
IDDPM [23]	Diff.	100%	2.90
U-ViT [5]	Diff.	100%	3.11
EDM [6]	Diff.	100%	**1.79**
DPT (**ours**, with EDM and FreeMatch)	Diff.	0.08% (4)	**1.81**

A.5 Image Classification with Few Labels

Tab. 6 presents a comparison of DPT with state-of-the-art semi-supervised classifiers on CIFAR-10. DPT outperforms competitive baselines [15, 14, 13] substantially with four labels per class, achieving the second-best error rate of $4.68_{\pm 0.17}\%$. Meanwhile, it's worth noting that the state-of-the-art method Full-flex and our work DPT are orthogonal. Since DPT is a flexible framework, integrating FullFlex [97] into DPT could potentially lead to further performance improvements.

Table 6: **Error rates on CIFAR-10 32×32. Bold** indicates the best result and underline indicates the second-best result. † labels the results taken from corresponding references, and ⋆ labels the baselines reproduced by us.

Method given # labels per class (label fraction)	Error rate ↓ 4 (0.08%)	 25 (0.5%)
Π Model [98]†	74.34±1.76	46.24±1.29
Pseudo Label [39]†	74.61±0.26	46.49±2.20
VAT [99]†	74.66±2.12	41.03±1.79
MeanTeacher [42]†	70.09±1.60	37.46±3.30
MixMatch [47]†	36.19±6.48	13.63±0.59
ReMixMatch [48]†	9.88±1.03	6.30±0.05
UDA [100]†	10.62±3.75	5.16±0.06
FixMatch [13]†	7.47±0.28	4.86±0.05
PPF [55] †	7.71±3.06	4.84±0.17
STOCO [52]†	7.18±1.95	4.78±0.30
Dash [101]†	8.93±3.11	5.16±0.23
MPL [102]†	6.62±0.91	5.76±0.24
FlexMatch [14]†	4.97±0.06	4.98±0.09
DST [54]†	5.00	-
FullFlex [97]†	**4.44**±0.15	**4.39**±0.04
FreeMatch [15]†	4.90±0.04	<u>4.88</u>±0.18
FreeMatch (**baseline**)⋆	4.93±0.13	-
DPT (**ours**) with EDM and FreeMatch	<u>4.68</u>±0.17	-

Algorithm 1 Pseudocode of DPT in a PyTorch style.

```
# Classifier: a classifier
# Generative_model: conditional generative models, such as diffusion models
# real_labeled_data: real labeled data
# real_unlabeled_data: real unlabeled data
# all_real_images: all images in real labeled and unlabeled data
# C: the number of classes in real labeled and unlabeled data
# K: the number of pseudo samples
# Uniform: uniform sampling function

### first stage:

# train a classifier
Classifier.train([(real_labeled_data.images, real_labeled_data.labels),
                  (real_unlabeled_data.images, )])

# predict pseudo labels for all real images
pseudo_labels = Classifier.predict(all_real_images)

### second stage

# train a conditional diffusion model
Generative_model.train([(all_real_images, pseudo_labels)])

uniform_labels = Uniform(C, K) # uniformly sample K labels from [0, C)

# sample K pseudo images by Generative_model
pseudo_images = Generative_model.sample(uniform_labels)

### third stage

# re-train the classifier
Classifier.train([(real_labeled_data.images,real_labeled_data.labels),
                  (pseudo_images, uniform_labels),
                  (real_unlabeled_data.images, )])
```

B Pseudocode of DPT

Algorithm 1 presents the pseudocode of DPT in the PyTorch style. Based on the implementation of the classifier and the conditional generative model, DPT is easy to implement with a few lines of code in PyTorch.

Table 7: **The code links and licenses.**

Method	Link	License
ADM	`https://github.com/openai/guided-diffusion`	MIT License
LDM	`https://github.com/CompVis/latent-diffusion`	MIT License
U-ViT	`https://github.com/baofff/U-ViT`	MIT License
DPM-Solver	`https://github.com/LuChengTHU/dpm-solver`	MIT License
FreeMatch	`https://github.com/TorchSSL/TorchSSL`	MIT License
Semi-ViT	`https://github.com/amazon-science/semi-vit`	Apache License
MSN	`https://github.com/facebookresearch/msn`	CC BY-NC 4.0
EDM	`https://github.com/NVlabs/edm`	CC BY-NC-SA 4.0

Table 8: **Model architectures in semi-supervised classifier and U-ViT.**

Model	Param	# Layers	Hidden Size	MLP Size	# Heads
Semi-Supervised Classifier (MSN)					
ViT B/4	86M	12	768	3072	12
ViT L/7	304M	24	1024	4096	16
Semi-Supervised Classifier (Semi-ViT)					
ViT Huge	632M	32	1280	5120	16
Conditional Diffusion Model (U-ViT)					
U-ViT-Large	371M	21	1024	4096	16
U-ViT-Huge	585M	29	1152	4608	16

C Experimental Setting

We implement DPT upon the official code of LDM [25], DPM-Solver [84], ADM [4], MSN [17], Semi-ViT [16], EDM [6], FreeMatch [15] and U-ViT [5], whose code links and licenses are presented in Tab. 7. All the architectures and hyperparameters are the same as the corresponding baselines [5, 17, 16, 15, 6]. For completeness, we briefly mention important settings and refer the readers to the original paper for more details. We also report the computational cost in Appendix. D.

SCDM. We extract features of ImageNet using the self-supervised method MSN [17] and perform k-means on these features to obtain meaningful cluster indices as conditions for training U-ViT-Large. Notably, in this way, we achieve an FID of 5.19 on ImageNet 256×256 without labels. However, the performance is still inferior to an FID of 3.31 achieved by supervised models.

The usage of pseudo images in the third stage. We focus on using pseudo images at a resolution of 256×256 because this resolution is closest to the commonly applied 224×224 resolution used for ImageNet classification. It is worth noting that for MSN based DPT, we utilize pseudo images generated by U-ViT-Large, except in cases where the DPT employs ViT-B/4 and has five labels per class and we use pseudo images generated by U-ViT-Huge instead. This is done to explore whether the pseudo images from the more powerful generative model can provide additional benefits to the classifier. For Semi-ViT based DPT, we employ pseudo images generated by U-ViT-Huge.

Network architectures. We present the network architectures in Tab. 8.

MSN. MSN adopts a warm-up strategy over the first 15 epochs of training, which linearly increases the learning rate from 0.0002 to 0.001, and then decays the learning rate to 0 following a cosine schedule. The total training epochs are 200 and 300 for the architecture of ViT L/7 and ViT B/4, separately. The batch size is 1024 for both two architectures. Actually, we reuse the two **pre-trained** models ViT L/7 and ViT B/4 provided by MSN [17] to reduce the training cost. After extracting the features by MSN, we use the `cyanure` package [103] to train the classifier following MSN [17]. In particular, we run logistic regression on a single CPU core based on `cyanure`.

U-ViT. U-ViT is based on the latent diffusion [25]. Specifically, we adopt two best configurations of U-ViT: U-ViT-Large and U-ViT-Huge. U-ViT-Large trains a transformer-based conditional generative

Table 9: The training time of DPT using U-ViT-H/2 and MSN with ViT-L/7 on ImageNet 256×256 with 5 labels per class. U-ViT-H/2 indicates that we use the U-ViT-Huge with the input patch size of 2×2. We present the percentage of additional computation cost of DPT in parentheses.

Model	Process	V100-hours	Cpu-hours
Classifier	Self supervised pre-training	2850	-
	Extracting features	30	-
	Linear classification	-	1
Generator	Generation	5760	-
DPT (extra cost)	Sampling	46	-
	Extracting features	4	-
	Linear classification	-	3
DPT	All training (sum all above)	8690 (**0.57%**)	4

model with a batch size of 1024, a training iteration of 300k, and a learning rate of 2e-4. U-ViT-Huge uses the same learning rate and batch size as U-ViT-Large but is trained for 500k iterations.

EDM. We use EDM for conditional generation on CIFAR-10 dataset. EDM trains a conditional diffusion model with a batch size of 512, a training duration of 200 Mimg, and a learning rate 1e-3.

Semi-ViT. We consider the best configuration of Semi-ViT with 1% labels, i.e., ViT-Huge. In the first stage, Semi-ViT uses the pre-training model of MAE. In the second stage, Semi-ViT trains a transformer-based classifier with a batch size of 128, a training epoch of 50, and a learning rate of 0.01. In the third stage, Semi-ViT trains a transformer-based classifier with a batch size of 64, a training epoch of 50, and a learning rate of 5e-3.

FreeMatch. FreeMatch trains a WRN-28-2 model with a batch size of 64, a training iteration of 2^{20}, and a learning rate 0.03.

D Computational Cost

We present the detailed computational cost of MSN based DPT and Semi-ViT based DPT on ImageNet 256×256 in Tab. 9 and Tab. 10, respectively.

As illustrated in Tab. 9, DPT with MSN introduces an additional computation cost of approximately $\frac{\text{DPT (extra cost)}}{\text{Classifier + Generator}} = \frac{50}{8640} = 0.57\%$, which is negligible. In particular, for conditional generation, the extra overhead we introduce is the cost of training the classifier. We reuse the pre-trained MSN to extract the features, and thus the training cost of the classifier can be reduced to only 30 V100-hours, which is negligible compared to the cost of the generator. For semi-supervised classification, the extra overhead we introduce is the cost of generative augmentation. The percentage of additional time cost over MSN is approximately 201.7%, calculated as $\frac{\text{Generator + DPT extra cost}}{\text{Classifier}} = \frac{5813}{2881} = 201.7\%$. Although DPT requires nearly twice the training time compared to the MSN baseline, it's still more time-efficient than other methods like Triple-GAN [11, 35], which demands at least 5 times the training time of its classifier.

Moreover, the percentage of additional computation cost of DPT with Semi-ViT is $\frac{\text{DPT (extra cost)}}{\text{Classifier + Generator}} = \frac{3886}{9664} = 40.21\%$, as shown in Tab. 10. Although Semi-ViT brings more accurate pseudo labels for conditional generation, it also needs more expensive training costs, creating a trade-off between label accuracy and computational expenses.

Furthermore, the computational cost of DPT on CIFAR-10 is presented in Tab. 11. The percentage of additional computation cost of DPT is $\frac{\text{DPT (extra cost)}}{\text{Classifier + Generator}} = \frac{169}{552} = 30.62\%$.

E Thought experiment

Classification and class-conditional generation are dual tasks that characterize opposite conditional distributions, e.g., $p(\text{label}|\text{image})$ and $p(\text{image}|\text{label})$. Learning such conditional distributions is con-

Table 10: The training time of DPT using Semi-ViT and U-ViT-H/2 on ImageNet 256×256 with 1% labels. U-ViT-H/2 indicates that we use the U-ViT-Huge with the input patch size of 2×2. We present the percentage of additional computation cost of DPT in parentheses.

Model	Process	V100-hours
Classifier	Supervised fine-tuning	64
	Semi-supervised fine-tuning	3840
Generator	Generation	5760
DPT (extra cost)	Sampling	46
	Semi-supervised fine-tuning	3840
DPT	All training (sum all above)	13550 (**40.21%**)

Table 11: The training time of DPT using FreeMatch and EDM on CIFAR-10 with 4 labels per class. We present the percentage of additional computation cost of DPT in parentheses.

Model	Process	V100-hours
Classifier	Classification	168
Generator	Generation	384
DPT (extra cost)	Sampling	1
	Classification	168
DPT	All training (sum all above)	721 (**30.62%**)

ceptually natural given a sufficient amount of image-label pairs. Recent advances in self-supervised learning[5] [104, 105, 106, 28, 107, 29] and diffusion probabilistic models [1, 2, 3, 4, 5, 6] achieve excellent performance in the two tasks respectively. However, both learning tasks are nontrivial in semi-supervised learning, where only a small fraction of the data are labeled (see Sec. 4 for a comprehensive review).

Most previous work solves the two tasks independently in semi-supervised learning while they can benefit mutually in intuition. The idea is best illustrated by a thought experiment with infinite model capacity and zero optimization error. Let $p(\text{image})$ be the true marginal distribution, from which we obtain massive samples in semi-supervised learning. Suppose we have a sub-optimal conditional distribution $p_c(\text{label}|\text{image})$ characterized by a classifier, a joint distribution $p_c(\text{image}, \text{label}) = p_c(\text{label}|\text{image})p(\text{image})$ is induced by predicting pseudo-labels for unlabeled data. Meanwhile, a conditional generative model trained on sufficient pseudo data from $p_c(\text{image}, \text{label})$ can induce the same joint distribution, as long as it is Fisher consistent[6] [108]. Because the generative model can further leverage the real data in a complementary way to the classifier, the induced joint distribution (denoted as $p_g(\text{image}, \text{label})$) is probably closer to the true distribution than $p_c(\text{image}, \text{label})$. Similarly, the classifier can be enhanced by training on pseudo data sampled from $p_g(\text{image}, \text{label})$). In conclusion, the classifier and the conditional generative model can benefit mutually through pseudo-labels and data in the ideal case.

F Additional Results and Discussions

F.1 More Samples and Failure Cases

Fig. 4 shows more random samples generated by DPT, which are natural, diverse, and semantically consistent with the corresponding classes.

[5] Self-supervised methods learn representations without labels but require full labels to obtain $p(\text{label}|\text{image})$.

[6] It means that the returned hypothesis in a sufficiently expressive class can recover the true distribution given infinite data.

(a) Random samples with *one* label per class. *Left:* "Gondola". *Right:* "Yellow lady's slipper".

(b) Random samples with *two* labels per class. *Left:* "Triceratops". *Right:* "Echidna".

(c) Random samples with *five* labels per class. *Left:* "School bus". *Right:* "Fig".

Figure 4: **More random samples in specific classes from DPT.**

Moreover, Fig. 5 depicts the randomly generated images in selected classes, from DPT trained with one, two, and five real labels per class. As shown in Fig. 5 (a), if the classifier can make accurate predictions given one label per class, then DPT can generate images of high quality. However, we find failure cases of DPT in Fig. 5 (d) and (g), where the samples are of incorrect semantics due to the large noise in the pseudo-labels. Nevertheless, as the number of labels increases, the generation performance of DPT becomes better due to more accurate predictions.

Notably, in Fig. 5, we fix the same random seed for image generation in the same class across DPT with a different number of labels (e.g., Fig. 5 (a-c)) for a fair and clear comparison. The samples of different models given the same random seed are similar because all models attempt to characterize the same diffusion ODE and the discretization of the ODE does not introduce extra noise [84], as observed in existing diffusion models [3, 5].

(a) *One* label per class, P (0.93), R (0.98) (b) *Two* labels per class, P (0.97), R (0.97) (c) *Five* labels per class, P (0.97), R (0.97)

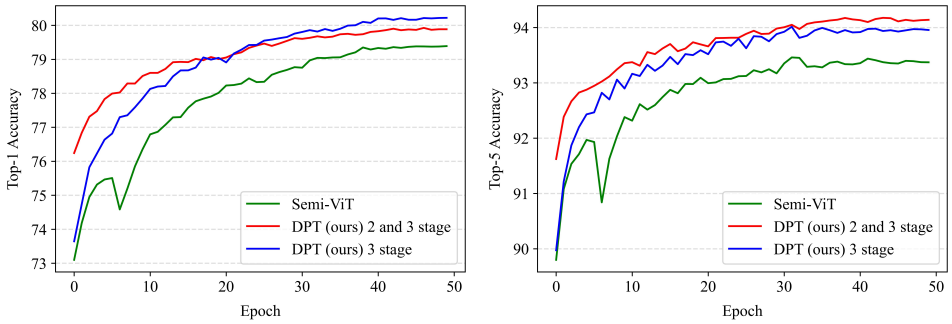
(d) *One* label per class, P (0.08), R (0.00) (e) *Two* labels per class, P (0.79), R (0.85) (f) *Five* labels per class, P (0.94), R (0.99)

(g) *One* label per class, P (0.02), R (0.02) (h) *Two* labels per class, P (0.60), R (0.98) (i) *Five* labels per class, P (0.65), R (0.98)

Figure 5: **Random samples by varying the number of real labels in the first stage.** More real labels result in smaller noise in pseudo-labels and samples of better visual quality and correct semantics. *Top:* "Custard apple". *Middle:* "Geyser". *Bottom:* "Goldfish".

F.2 How to use pseudo images in Semi-ViT

For Semi-ViT based DPT, in order to fully leverage the pseudo images, we consider the two settings: (1) replaces $\mathcal{S}$ with $\mathcal{S} \cup \mathcal{S}_2$ in the third stage of Semi-ViT, which is mainly considered in the main text. (2) replaces $\mathcal{S}$ with $\mathcal{S} \cup \mathcal{S}_2$ in the two and third stages of Semi-ViT. As shown in Fig. 6, pseudo images indeed improve the performance of Semi-ViT. Besides, we also find that although the utilization of pseudo images in the second stage of Semi-ViT can provide initial points with high classification accuracy for the third stage, the final top-1 accuracy is lower than just utilizing pseudo images in the third stage of Semi-ViT.



(a) Top-1 Accuracy (b) Top-5 Accuracy

Figure 6: **Semi-ViT based DPT.** 2 and 3 stage means that we replaces $\mathcal{S}$ with $\mathcal{S} \cup \mathcal{S}_2$ in the two and third stage of Semi-ViT. 3 stage means that we replaces $\mathcal{S}$ with $\mathcal{S} \cup \mathcal{S}_2$ in the third stage of Semi-ViT. **(a-b)** These two settings both improve the performance of Semi-ViT and stabilize the training.

Table 12: **Comparison with the state-of-the-art fully supervised models on ImageNet classification** † labels the results taken from corresponding references and ⋆ labels the baselines reproduced by us.

Method	Data	Top-1	Top-5
ResNet-50 [109]†	ImageNet	76.0	93.0
ResNet-152 [109]†	ImageNet	77.8	93.8
Inception-v3 [110]†	ImageNet	78.8	94.4
Inception-v4 [95]†	ImageNet	80.0	95.0
SENet-154 [111]†	ImageNet	81.3	95.5
EfficientNet-L2 [112]†	ImageNet	85.5	97.5
DeiT-B [113]†	ImageNet	81.8	-
Swin-B [114]†	ImageNet	83.3	-
MAE [29]†	ImageNet	86.9	-
Semi-ViT [16]†	1% ImageNet	80.0	93.1
Semi-ViT [16]⋆	1% ImageNet	79.4	93.4
DPT (ours)	1% ImageNet	80.2	94.0

We also compare DPT with Semi-ViT and state-of-the-art fully supervised models (see Tab. 12) and find that DPT performs comparably to Inception-v4 [95], using only 1% labels.

F.3 Results with More Stages

According to Tab. 4, we find that using generative augmentation can lead to more accurate predictions on unlabeled images. Therefore, we attempt to add a further stage employing the refined classifier to predict pseudo-labels for all data, and then re-train the conditional generative model on them. As shown in Tab. 13, these refined pseudo-labels indeed bring a consistent improvement on all quantitative metrics, showing promising promotion of more-stage training. However, note that re-training the conditional generative model is time-consuming and we focus on the three-stage strategy in this paper for simplicity and efficiency.

Table 13: **Effect of refined pseudo-labels.** Results on ImageNet 256×256 benchmark with one label per class.

Method	FID↓	sFID ↓	IS ↑	Precision ↑	Recall ↑
DPT	4.34	6.68	162.96	0.80	0.53
DPT with refined pseudo-labels	4.00	6.56	178.05	0.81	0.53

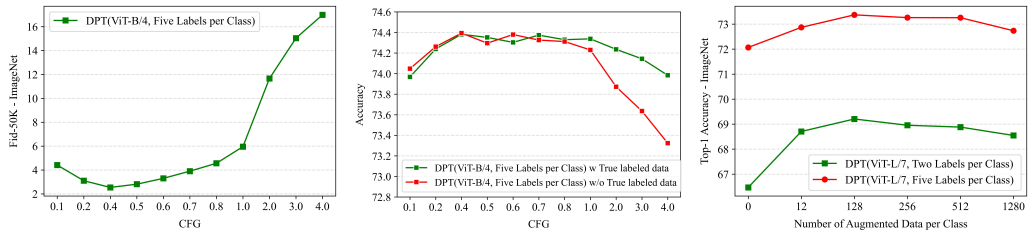
F.4 Can DPT Improve the Upper Bound of Generation Quality

When all labels are available, the second stage of DPT becomes equivalent to training a supervised diffusion model with real labels. This is essentially the same as a supervised conditional baseline. Therefore, by combining fully supervised labeled data, DPT will not surpass the baseline (e.g. 2.29 FID of U-ViT).

G Ablation Studies

Sensitivity of *K*. The most important hyperparameter in DPT is the number of augmented pseudo images per class, i.e., K. In order to analyze the sensitivity of DPT with respect to K, we perform a simple grid search on $\{12, 128, 256, 512, 1280\}$ in MSN (ViT-L/7) with two and five labels per class and find that $K = 128$ is the best choice. Therefore, we set $K = 128$ as the default value if not specified (see Fig. 7 (c)). We observed that $K = 128$ was the optimal choice in both settings. Intuitively, an overly large K would cause the classifier to be dominated by pseudo images and ignore real data, which explains the sub-optimal performance with $K > 128$. Nevertheless, according to Fig. 7 (c), DPT consistently and substantially improved the baselines ($K = 0$) over a large range of values in {12, 128, 256, 512, 1280}.

Sensitivity of *CFG*. In the third stage of DPT, we replaces $\mathcal{S}$ with $\mathcal{S} \cup \mathcal{S}_2$. The choice of w is highly non-trivial for the quality of pseudo images, Therefore, it is also significant for classification. We conducted experiments on the ImageNet dataset with five labels per class, sweeping over a range of values for w from 0.1 to 4.0, and evaluated the performance of the model in terms of FID-50K and top-1 Accuracy. The results are presented in Figure 7 (a-b). Moreover, we find that the choice of CFG that minimizes FID will lead to the best accuracy. Specifically, we find that $CFG = 0.4$ achieves the best performance for ImageNet 256×256, while $CFG = 0.8$ and $CFG = 0.7$ are the optimal choices for ImageNet 128×128 and 512×512, respectively.



(a) (b) (c)

Figure 7: **Sensitivity. (a-b)** CFG is highly non-trivial for FID-50K and accuracy. When choosing the CFG that minimizes FID, accuracy tends to be higher, and to some extent, the higher the FID, the worse the accuracy will be. For ImageNet 256×256, $CFG = 0.4$ is the best choice. **(c)** DPT improves the baselines ($K = 0$) with K in a large range. $K = 128$ is the best choice.

H How Does Classifier Benefit Generation?

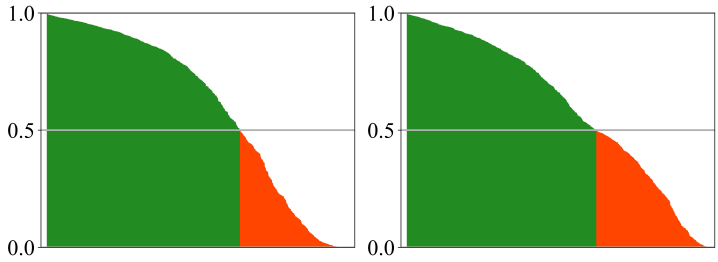
We explain why the classifier can benefit the generative model through class-level visualization and analysis based on precision and recall on training data. For a given class y, the precision and recall w.r.t. the classifier is defined by $P = TP/(TP + FP)$[7] and $R = TP/(TP + FN)$, where TP, FP, and FN denote the number of true positive, false positive, and false negative samples respectively. Intuitively, higher P and R suggest smaller noise in pseudo-labels and result in better samples. Therefore, we employ strong semi-supervised learners [17] in the first stage to reduce the noise.

We select three representative classes with different values of P and R and visualize the samples in Fig. 8. In particular, the pseudo-labels in a class with both high P and R contain little noise, leading to good samples (Fig. 8 (a)). In contrast, on one hand, a low P means that a large fraction of images labeled as y in $\mathcal{S}_1$ are not actually in class y, and the samples from the generative model given the

[7]We omit the dependence on y for simplicity.

(a) P(0.99),R(1.00) "Giant Panda" (b) P(0.16),R(0.75) "Purse" (c) P(0.99),R(0.23) "Hornbill"

Figure 8: **Random samples in selected classes with different *P* and *R*.** (a) High P and R ensure good samples. (b) Low P leads to semantical confusion. (c) Low R lowers the visual quality.



(a) Distribution of R (b) Distribution of P

Figure 9: **Distributions of *R* and *P***. The vertical axis represents the values of P and R w.r.t the classifier trained in the first stage. The horizontal axis represents all classes sorted by the values.

label y can be semantically wrong (Fig. 8 (b)). On the other hand, a low R means that a large fraction of images in class y are misclassified as other labels and the samples from the generative model can be less realistic due to the lack of training images in class y (Fig. 8 (c)).

Through the analysis of the three representative classes above, the classifier benefits the generator by bringing more accurate and low-noise pseudo labels to the generator. In particular, with *one label per class*, MSN [17] with ViT-L/7 achieves a top-1 training accuracy of 60.3%. As presented in Fig. 9, R and P of most classes are higher than 0.5. Quantitatively, despite using only $< 0.1\%$ of the labels, DPT achieves an FID of 3.08, compared to the FID of 2.29 achieved by U-ViT-Huge with full labels. The reduction in FID is not significant. This demonstrates that although noise exists, such a strong classifier can benefit the generative model overall and reduce the usage of labels.

I How Does Generative Model Benefit Classification?

Similarly to Appendix. H, we explain why the generative model can benefit the classifier through class-level visualization and analysis based on precision (P) and recall (R).

We select three representative classes with different values of change of R for visualization in Fig. 10. If the pseudo images in class y are realistic, diverse, and semantically correct, then it can increase the corresponding R as presented in Fig. 10 (a-b). Instead, poor samples may hurt the classification performance in the corresponding class, as shown in Fig. 10 (c).

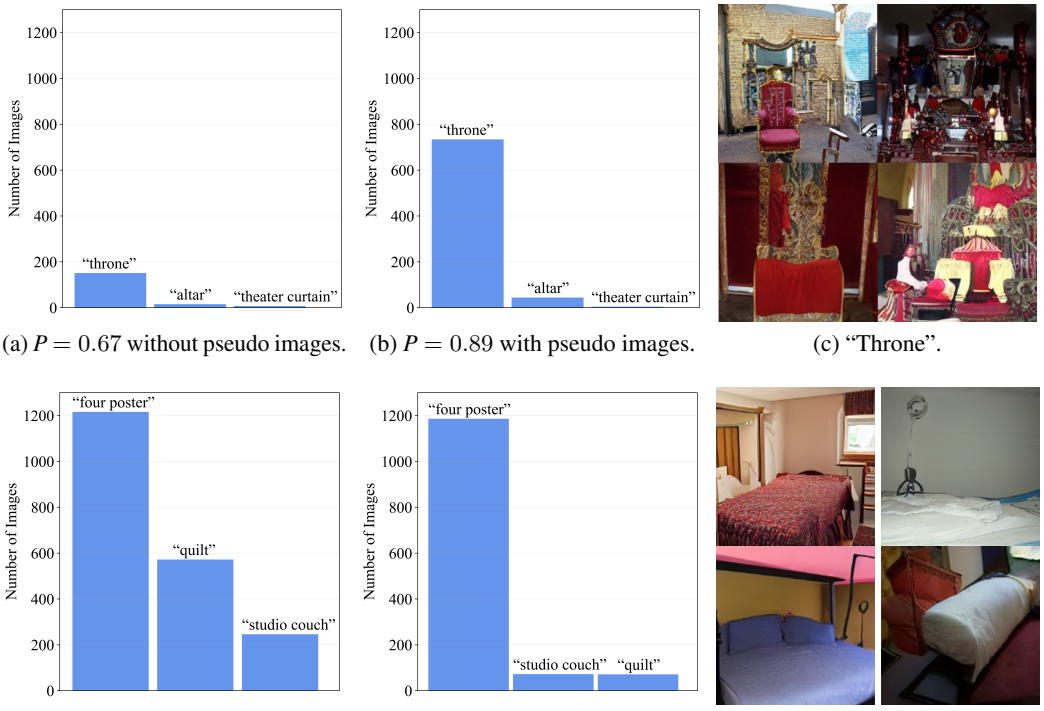
The analysis of P involves pseudo images in multiple classes. According to the definition of precision (i.e. $P = TP/(TP + FP)$), the pseudo images can affect P through not only TP but also FP. We select two representative classes with positive changes of P to visualize both cases, as shown in Fig. 11. We select the top-three classes according to the number of images classified as "throne" and present the numbers w.r.t. the classifier after the first and third stages in Fig. 11 (a) and (b) respectively. As

(a) R (0.24 $\rightarrow$ 0.87) "Albatross" (b) R (0.22 $\rightarrow$ 0.75) "Timber wolf" (c) R (0.26 $\rightarrow$ 0.01) "Bathtub"

Figure 10: **Random samples in selected classes with different values of change of R.** Values of change are presented in parentheses. (a-b) If pseudo images are realistic and semantically correct, they can benefit classification. (c) Otherwise, they hurt performance.

shown in Fig. 11 (c), high-quality samples in the class "throne" directly increases TP (i.e., the number of images in class "throne" classified as "throne") and improves P. We also present the top-three classes related to "four poster" in Fig. 11 (d) and (e). It can be seen that P in class "four poster" increases because of FP (of class "quilt" especially) decreases. We visualize random samples in both "four poster" and "quilt" in Fig. 11 (f). These high-quality samples help the classifier to distinguish the two classes, which explains the change of FP and P. A similar analysis can be conducted for classes with negative change of P.

We mention that we analyze the change of P and R in the third stage on the training set instead of the validation set. This is because the training set is of a much larger size and therefore leads to a much smaller variance in the estimate of P and R. Since most of the data are unlabeled in the training set, this does not introduce a large bias in the estimate of P and R.



(a) $P = 0.67$ without pseudo images. (b) $P = 0.89$ with pseudo images. (c) "Throne".

(d) $P = 0.52$ without pseudo images. (e) $P = 0.86$ with pseudo images. (f) *Left:* "Four poster". *Right:* "Quilt".

Figure 11: **Detailed analysis in selected classes with a positive change of *P*.** *Top:* "Throne". High-quality samples in the class "throne" (c) directly increase *TP* and improve *P* (a-b). *Bottom:* "Four poster". The samples in both "four poster" and "quilt" are of high quality (f). The classifier reduces *FP* with such pseudo samples and improves *P* (d-e).

(a) (b)

Figure 12: 512×512 samples of DPT trained with five labels per class. *CFG* = 3.0

(a) (b)

Figure 13: 512×512 samples of DPT trained with five labels per class. *CFG* = 3.0

(a) (b)

Figure 14: 512×512 samples of DPT trained with five labels per class. *CFG* = 3.0