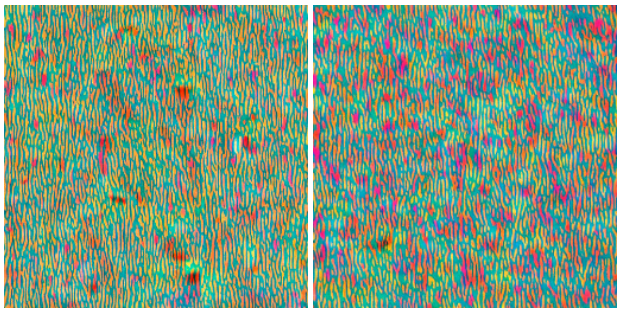
Supplementary Material: Towards Practical Plug-and-Play Diffusion Models

Contents



(a) Label: vulture (b) Label: ice lolly

Figure 12. Generated images from plug-and-play priors [5] with ImageNet pre-trained ResNet50. On the ImageNet dataset, we observe that plug-and-play priors fail to generate realistic images.

A. Detailed Comparison with Previous Methods

We compare our method to two baselines 1) *plug-and-play priors* [5] and 2) gradients on $\hat{x}_0$ by using the approach in [1]. This section describes the details of the comparison and their limitation. We note that these methods do not require training, but our method requires finetuning the off-the-shelf model.

First, we observe that plug-and-play priors fail to produce realistic images on the ImageNet dataset. We generate images with unconditional 256×256 ADM [4] and pre-trained ResNet50 [6] using their official code[1]. In the same setting as FFHQ experiments in their work, we try to generate images by guiding them to random class labels. As shown in Fig. 12, the generated images are not realistic. We also increase the optimization step of their method, but there are no significant differences. Correspondingly, their FID and IS are 358.20 and 1.55, respectively.

We provide quantitative results of gradients on $\hat{x}_0$ in Table 1 when $s = 7.5$. To investigate in further depth, we compare our method with theirs while adjusting the guidance scale s and utilizing the ResNet50 classifier. For implementing gradients on $\hat{x}_0$, we use the official code[2] of Blended Diffusion [1]. As shown in Fig. 13, gradients on $\hat{x}_0$ do not significantly improve FID, IS, and Precision, indicating that it does not guide the diffusion model to the class label. From this, we conjecture that it is effective in image editing with the assistance of several techniques [1] but not for guiding the diffusion without these techniques.

As the concurrent work, eDiff-I [2] replicates copies of a single diffusion model and specializes each copy on different time intervals. However, they apply the idea to the diffusion model, not the guidance model. Also, while eDiff-I applied its idea to the diffusion model by focusing on training efficiency through a branching scheme, we applied our idea to the guidance model to efficiently increase the number of experts with the parameter-efficient fine-tuning strategy.

[1] https://github.com/AlexGraikos/diffusion_priors

[2] https://github.com/omriav/blended-diffusion

B. Details of Observational Study

We use the ResNet50 model and unconditional diffusion model to illustrate our observation in Section 3.2. In this section, we discuss the details of the experimental setup.

For off-the-shelf ResNet50, we exploit Imagenet pre-trained ResNet50[3]. On the ImageNet training dataset, we fine-tune the off-the-shelf model for a single noise-aware ResNet50 that learns the entire timestep $t \in [1, ..., 1000]$. During 300k iterations, AdamW [10] with a learning rate of 1e-4 and weight decay of 0.05 is utilized as the optimizer. The batch size is set to 256.

For each expert ResNet50, we use the same optimizer as the single noise-aware model above. In order to make the total iterations for training five experts equal to those of the single noise-aware model, we fine-tune off-the-shelf ResNet50 with a batch size of 256 and 60k iterations.

During the reverse process with guidance, the guidance scale s is set as 7.5 since it achieves good results for most variants.

C. Details of Parameter Efficient Multi-Experts

For the parameter-efficient multi-experts strategy described in Section 4.1, we only fine-tune a small number of parameters while reusing most of the frozen off-the-shelf parameters. Here, we report how parameter efficient tunning is applied to each architecture used in our experiments.

We first apply LORA [7] to certain weight matrix of the off-the shelf model $W_0 \in \mathbb{R}^{d\times k}$, which updates it with low-rank decomposition as $W_0 + \alpha/rBA$ where $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$. During fine-tuning, A and B are updated, while W_0 is frozen. When inferencing the guidance model in the reverse process, the weight matrix W_0 is simply updated as $W = W_0 + BA$, resulting in no additional inference cost. We note that α is fixed as 8.

If batch normalization and bias terms are used in the architecture, we tune these as well. This incurs no additional inference costs because it does not alter the model design, such as layer expansion. We note that such parameter efficient strategy is applicable to various architectures.

C.1. ADM Guidance Models

We use ResNet-50 [6] and DeiT-S [16] architecture for the guidance model in ImageNet class conditional generation. For ResNet-50 architecture, LORA is applied to the first and the second convolutions of each block, and also to

[3] We use publicly available torchvision [12] ResNet50

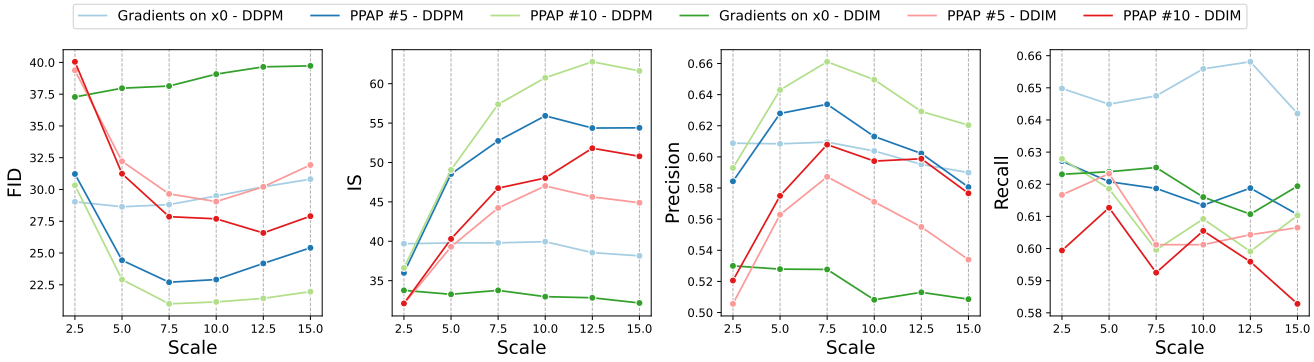


Figure 13. Quantitative comparison between PPAP and gradients on $\hat{x}_0$ according to guidance scale s.

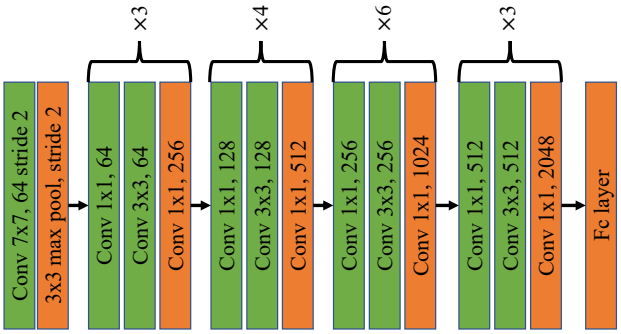


Figure 14. ResNet50 model with LORA. The green color is the layer where LORA is applied.

the first 7×7 convolution as shown in Fig. 14. We set the LORA rank as $r = 16$, and also tune the bias and batch normalization layers.

DeiT-S is a transformer-based model comprised of a self-attention module and an MLP module. We only apply LORA to self-attention module weights with a rank of $r = 32$. Since DeiT-S does not have batch normalization layers, we only fine-tune bias terms.

C.2. GLIDE Guidance Models

Image classifier We use the ResNet50 [6] classifier for guiding GLIDE to conduct class conditional image generation. All configuration of parameter-efficient multi-expert is identical to those described in Section C.1.

Depth estimator MiDaS-small [13] is utilized for guiding GLIDE. MiDaS is a monocular depth estimation, which is used for various tasks such as the single-image view synthesis [9, 11, 14] in a frozen state. MiDaS-small is compromised of two components: 1) Backbone network which is based on EfficientNet lite3 [15] and 2) CNN-based decoder network. We apply LORA [7] with $r = 8$ to point-wise convolution layers of the backbone network as well as convolution layers of the CNN-based decoder network. Also, bias terms and batch normalization layers are fine-tuned. The total number of trainable parameters used per expert is 0.7M, which is 3.73% compared to the MiDaS-small parameter of 21M.

Semantic segmentation DeepLabv3-ResNet50 [3] is exploited for guiding GLIDE. DeepLabv3-ResNet50 is compromised of two components: 1) Backbone network which is based on ResNet-50 [6] and 2) Atrous Spatial Pyramid Pooling (ASPP) segmentation classifier decoder. We apply LORA to the ResNet-50 backbone. All configuration of LORA is the same as in Section C.1. As a result, we introduce 0.88M trainable parameters, which is 2.15% compared to the DeepLabv3 parameter of 41.95M.

D. Guidance Formulation of PPAP for Other Tasks

Here, we explain how the monocular depth estimator and the semantic segmentation model can be incorporated into the guidance model.

Monocular Depth Estimation A monocular depth estimation model takes an image as input and outputs a depth map $f(x) \in \mathbb{R}^{H \times W}$, where H and W represent the height and width of the input image, respectively. We formulate knowledge transfer loss $\mathcal{L}_{de}$ for depth estimation models as:

$$\mathcal{L}_{de} = \|\mathrm{sg}\big(f_\phi(\tilde{x}_0)\big) - f_{\phi_n}(\tilde{x}_t)\|_1. \tag{1}$$

Then, we guide image generation so that the image has some desired depth map $D_{target} \in \mathbb{R}^{H \times W}$ as follows:

$$\mathcal{L}_{gd} = \|f_{\phi_n}(x_t) - D_{target}\|_1. \tag{2}$$

Semantic Segmentation A semantic segmentation model takes an image as input and outputs a segmentation map, having classification logit vector at the pixel level, $f(x) \in$

$\mathbb{R}^{C\times H\times W}$. We formulate knowledge transfer loss $\mathcal{L}_{ss}$ for semantic segmentation models as:

$$\mathcal{L}_{ss} = \|\mathrm{sg}\big(f_{\phi}(\tilde{x}_0)\big) - f_{\phi_n}(\tilde{x}_t)\|_1. \tag{3}$$

Then, we guide image generation so that the image is aligned with the segmentation map $S_{target} \in \mathbb{R}^{C\times H\times W}$ as follows:

$$\mathcal{L}_{gs} = \|\mathrm{s}(f_{\phi_n}(x_t)) - S_{target}\|_1. \tag{4}$$

E. Experimental Details

E.1. Guiding ADM for ImageNet Class Conditional Generation

When we train the models, AdamW optimizer [10] is commonly used with a learning rate of 1e-4, weight decay of 0.05, and batch size of 256. All variants are trained with the same total iterations of 300k. For implementing LORA, we use the official code of LORA[4] (See the details of parameter-efficient multi-expert in Section C.1). We utilize torchvision pre-trained ResNet50 and timm [17] pre-trained DeiT-S for the off-the-shelf model. We calculate FID, IS, Precision, and Recall with 10k generated samples with random class labels. All experiments are conducted on 8×A100 GPUs.

E.2. Guiding GLIDE for Various Downstream Tasks

Same with the ImageNet classifier guidance setting, we use an AdamW optimizer with a learning rate of 1e-4 and weight decay of 0.05 to train the models.

In image classifier guidance for GLIDE, we use the same setting as in ImageNet class conditional generation with ADM [4]. For depth estimation and semantic segmentation models, we use MIDAS-small and DeepLabv3-ResNet50, publicly available in torch-hub [12]. We train these with a batch size of 128 and 300k iterations.

For guidance scale s, we use norm-based scaling for guidance gradients [8]. We set the gradient ratio as 0.3.

F. More Ablation Study on ImageNet Conditional Generation

F.1. Effect of Guidance Scale s

Here, we change the guidance scale s from 2.5 to 15.0 for identifying performance according to the guidance scale s.

Figure 15 shows the results according to the guidance scale s where DDIM sampler with 25 steps is used. From these results, we can see that multi-expert and PPAP greatly outperform the single-noise aware model in most guidance scales. Second, it can be seen that there is a sweet spot around $s = 7.5$, where the guidance scale is neither too large nor too small, and where neither the image quality nor the guidance capability is compromised. Therefore, we set the default guidance scale as 7.5 in our experiments.

[4] https://github.com/microsoft/LoRA

Sampler	Method	FID	IS	Precision	Recall
DDIM 25 step	ResNet152 + PPAP	**26.78**	**49.59**	**0.5976**	0.6208
	ResNet50 + PPAP	29.65	44.23	0.5872	0.6012
	ResNet152 + single noise aware	29.63	42.09	0.5601	0.6305
	ResNet152 + off-the-shelf	40.15	33.33	0.5074	0.6175
DDPM 250 step	ResNet152 + PPAP	**20.34**	**61.01**	**0.6401**	0.6164
	ResNet50 + PPAP	22.70	52.74	0.6338	0.6187
	ResNet152 + single noise aware	22.75	53.38	0.6261	0.6363
	ResNet152 + off-the-shelf	31.44	38.28	0.5861	0.6529

Table 2. Quantitative results on ADM guidance by increasing the size of classifiers. Increasing the size of the model from ResNet50 to ResNet152 improves the performance of guidance, and PPAP with ResNet152 outperforms single noise-aware ResNet152. Although PPAP is trained in an unsupervised manner, it outperforms the single noise-aware model.

Sampler	Method	Trainable Parameters	FID	IS	Precision	Recall
DDIM 25 step	ResNet18×5	58.4M	**20.38**	**63.74**	**0.6638**	0.5898
	ResNet152×1	60M	29.63	42.09	0.5601	0.6305
	ResNet50×5	109.9M	**19.98**	**74.77**	**0.6476**	0.5887
	ResNet50×1	25.5M	30.42	43.05	0.5509	0.6187
DDPM 250 step	ResNet18×5	58.4M	**16.65**	**76.68**	**0.7182**	0.579
	ResNet152×1	60M	22.75	53.38	0.6261	0.6363
	ResNet50×5	109.9M	**16.37**	**81.47**	**0.7216**	0.5805
	ResNet50×1	25.5M	38.15	31.29	0.5426	0.6321

Table 3. Quantitative comparison between multi-expert strategy and single noise-aware model. ×5 represents using five experts and ×1 represents using single noise-aware model. A multi-expert configuration with the same architecture and parameter fair significantly outperforms a single noise-aware model.

F.2. Effect of Guidance Model Size

We also analyze the guidance when the size of the model increases. Instead of using ResNet50, we use ResNet152 for training the single noise-aware model and PPAP with five experts. Specifically, we implement LORA with the same configuration in ResNet50, resulting in 8.6% trainable parameters per expert compared to the parameter of ResNet152. Training settings such as the optimizer and iterations are the same as ADM guidance with ResNet50.

As shown in Table 2, we observe that 1) increasing the classifier size from ResNet50 to ResNet152 improves the performance of guidance and 2) PPAP also outperforms the single noise-aware model even when ResNet152 is used.

F.3. Effect of Multi-Experts

In Section 5.1, we validate the efficacy of the multi-experts strategy by comparing the results with a range of expert numbers [1, 2, 5, 8, 10]. Here, we present more results for supporting the effectiveness of the multi-experts.

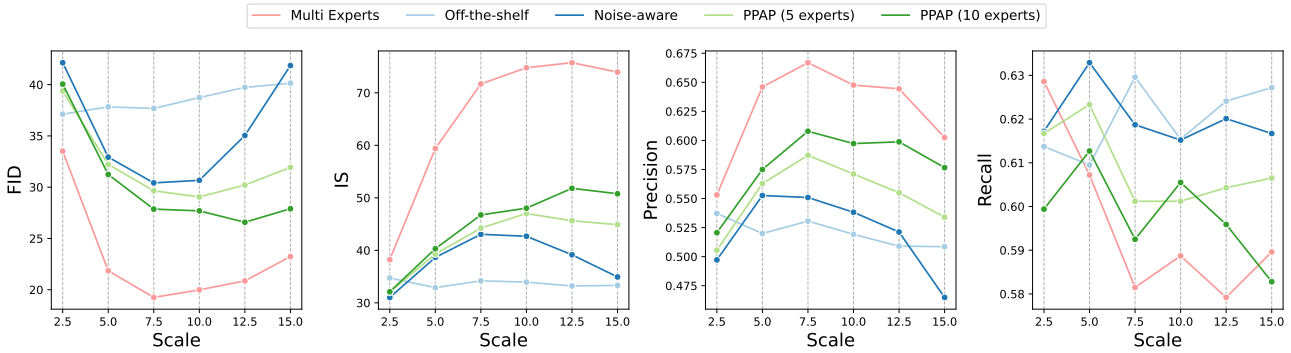


Figure 15. Quantitative results on guiding unconditional ADM with ResNet50 and DDIM 25 steps according to guidance scale s.

Method	t-conditioned	FID	IS	Precision	Recall
Multi-expert-5	✓	19.98	74.78	0.6476	0.5887
	✗	19.54	73.23	0.6509	0.5916
Single noise aware	✓	30.08	43.00	0.6017	0.6059
	✗	30.42	43.05	0.6009	0.6187

Table 4. Impacts of t-condition on ResNet50 with DDIM 25 step sampler.

Method	Parameter-efficient	Dataset	FID	IS	Precision	Recall
Multi-experts-5	✗	ImageNet	19.98	74.78	0.6476	0.5887
		Data-Free	29.87	44.10	0.5837	0.5912
	✓	ImageNet	20.32	72.30	0.6333	0.5898
		Data-Free	29.65	44.23	0.5872	0.6012

Table 5. Impacts of the data-free strategy on ResNet50 with DDIM 25 step sampler.

We compare the multi-expert strategy with 5 experts and the single-noise-aware model in a setting where the trainable parameters are fair. For fair trainable parameters, we fine-tune 5 experts ResNet18 and single noise-aware ResNet152 [6]. Both models are trained with a total of 300k iterations on the ImageNet train dataset using the AdamW optimizer, a learning rate of 1e-4, and a weight decay of 0.05.

Table 3 shows the quantitative comparison between the multi-expert strategy and a single noise-aware model. Through these results, we empirically confirm that the multi-expert strategy with the same architecture and parameter fair setting significantly outperforms a single noise-aware model. Specifically, in parameter fair setting, five experts with ResNet18 significantly outperform single noise-aware ResNet152. We note that a multi-expert strategy does not incur additional inference time costs, but increasing the size of a single noise-aware model can incur additional inference time costs. Considering these, it seems much better to construct a multi-expert rather than increase the size of a single noise-aware model.

F.4. Effect of Timestep Conditioned Guidance Model

Multi-experts strategy can be seen as a piecewise function w.r.t time step t. From this point of view, one can wonder about the advantage of multi-experts over t-conditioned models. To answer this and show the effectiveness of the multi-expert strategy, we compare the performance of the model when it is conditioned on t and it is not. We add the t-embeddings of [4] to the input of the ResNet50 model for conditioning and generate images with DDIM 25-step sampler. As shown in Table 4, naively conditioning the model on t does not improve the performance.

F.5. Effect of Data-Free Knowledge Transfer

We conducted an ablation study for understanding the impacts of the data-free strategy. We used five ResNet50 experts with and without parameter efficient strategy applied in them. Table 5 shows the quantitative results when these five experts were supervisely trained on ImageNet and trained on data-free knowledge transfer. The results indicate that the models trained on labeled data produce better guidance than those trained with the data-free strategy.

G. More Qualitative Results on ADM

G.1. Guided ADM with ResNet50

Due to limited spaces, we only show qualitative results with DDPM 250 steps in Section 5.1. We illustrate more qualitative results with DDPM 250 steps and DDIM 25 steps in Fig. 16 and Fig. 17, respectively.

G.2. Guided ADM with DeiT-S

Only qualitative results with DDPM 250 steps are presented in Section 5.1, because of limited spaces. Fig. 18 and Fig. 19 show qualitative results with DDIM 250 steps and DDIM 25 steps, respectively.

H. More Qualitative Results on GLIDE

We note that the reference batch for measuring quantitative results such as FID, IS, Precision, and Recall is not valid since data for training GLIDE is not publicly available. Instead, we present various qualitative results to show that PPAP can guide GLIDE.

H.1. GLIDE + ResNet-50

To show the effectiveness of PPAP, we illustrate more qualitative results in Fig. 20. Furthermore, we also show multiple generated images per class in Fig. 21.

H.2. GLIDE + Depth

We provide more qualitative results in guiding GLIDE with depth estimator in Fig. 22 and Fig. 23. As illustrated in Fig. 22, compared to off-the-shelf that does not reflect the given depth at all, it is confirmed that the proposed PPAP framework works well as a guide. We also provide multiple results from the same depth map, shown in Fig. 23. Interestingly, our framework not only generates the proper images corresponding to the given guidance with depth but also infers the diverse objects that suit the given depth.

H.3. GLIDE + Segmentation

More qualitative results in guiding GLIDE with semantic segmentation are illustrated in Fig. 24 and Fig. 25. As illustrated in Fig. 24, our PPAP framework can generate images suited to given segmentation maps. It shows that our PPAP framework is capable of both semantic-level guidance and pixel-level guidance at once. We also provide in-class multiple images in Fig. 25.

I. Limitation and Discussion

Guiding GLIDE with our PPAP often generates images with the style of data that trains the diffusion model but not the guidance model. On the one hand, it means that the guidance model can leverage various data covered by diffusion, but on the other hand, it can be interpreted that the guidance model cannot perfectly guide the diffusion model to the data it has learned. Considering this, addressing the train dataset mismatch between the diffusion model and the off-the-shelf model can be the future direction of this work.

Also, we only use the guidance models that take a single image as input. There are several publicly available off-the-shelf models which take not only the image but also other inputs. With designing suitable knowledge transfer loss and guidance loss, collaborating with these off-the-shelf models will produce various applications. Therefore, applying it to various applications can be future work.

J. Acknowledgement

We thank Minsam Kim, Minhyuk Choi, Eunsam Lee, and Jaeyeon Park for the inspiration for this work.

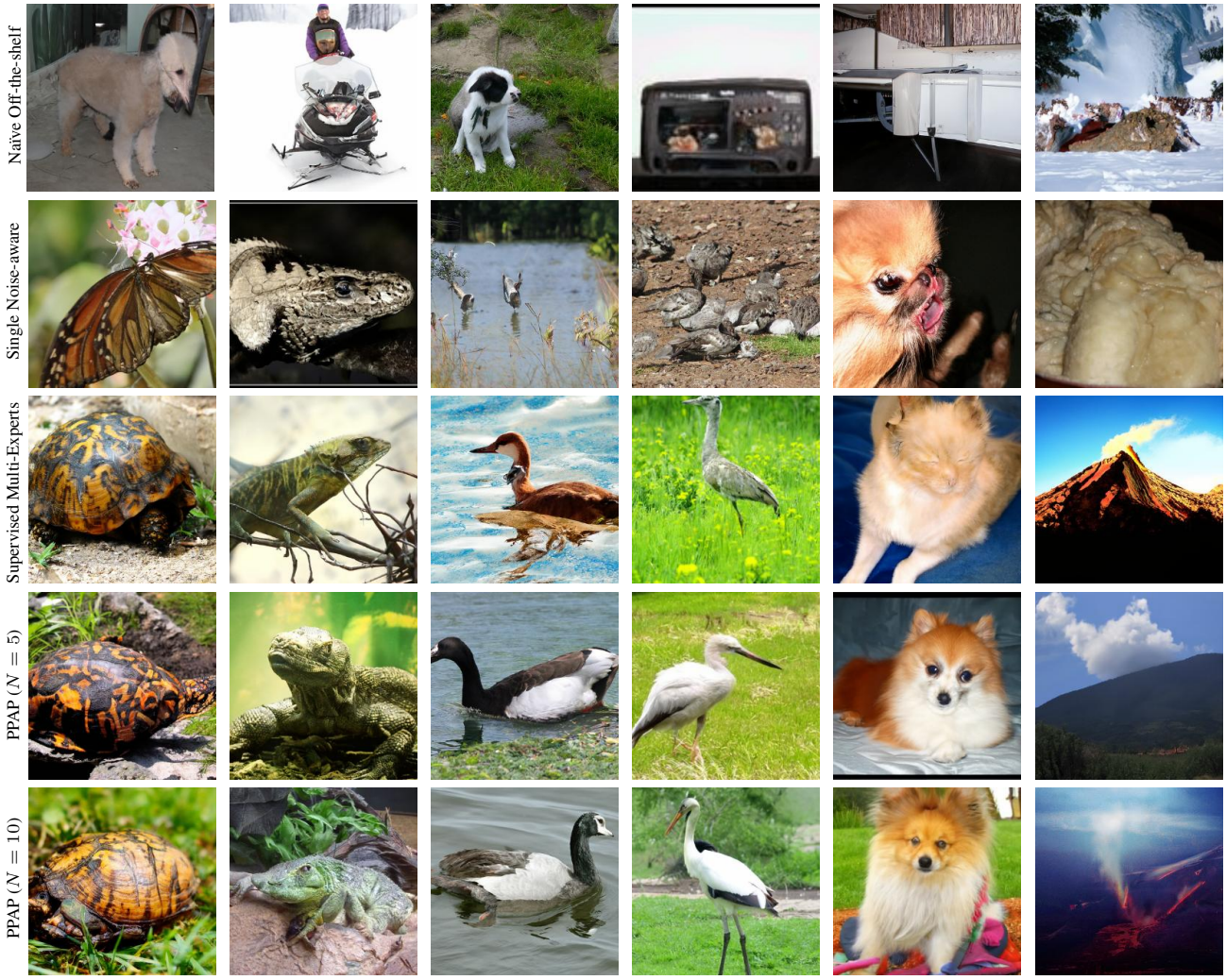


Figure 16. Qualitative results on ImageNet class conditional generation with DDPM 250 steps by guiding unconditional ADM with ResNet50.

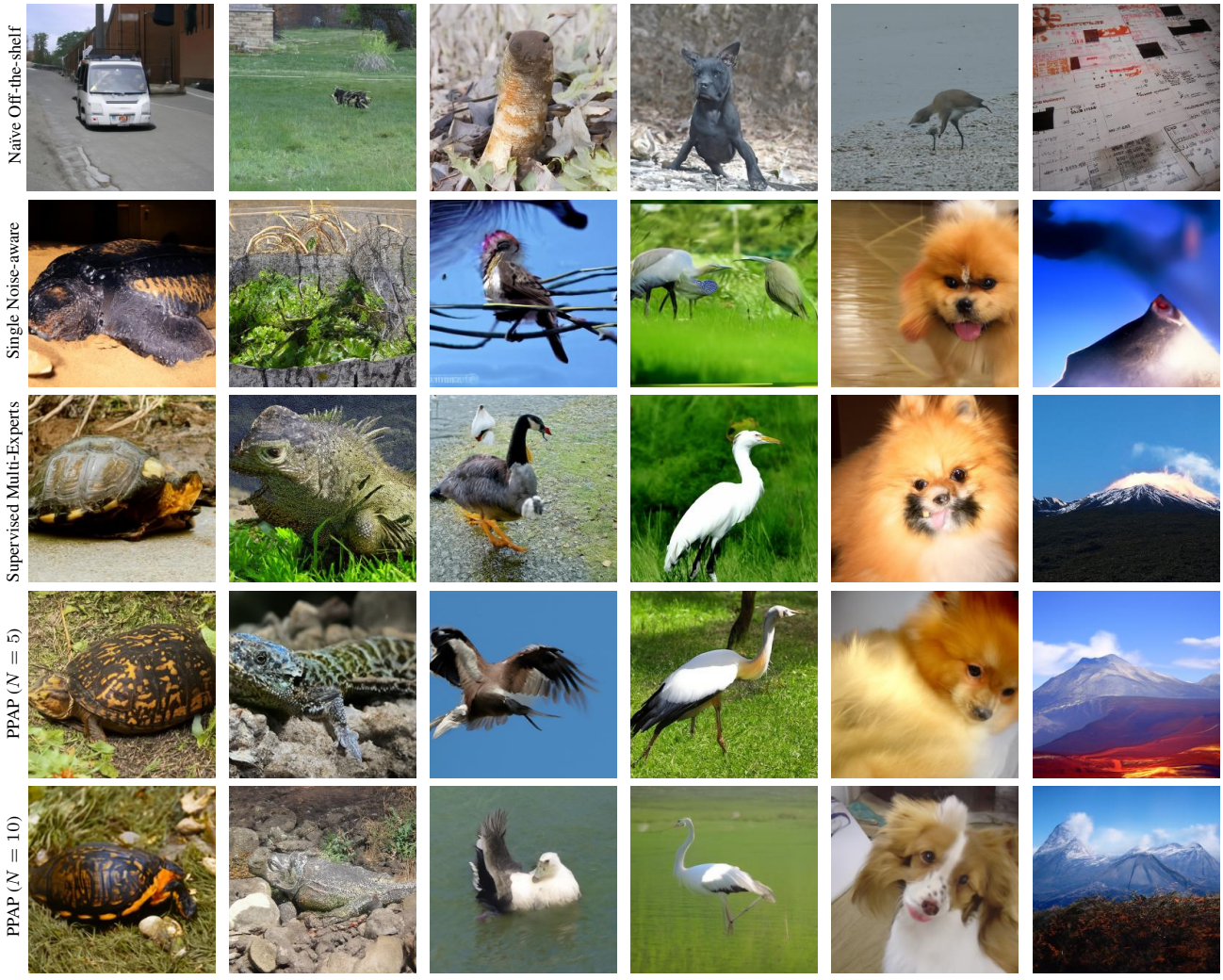


Figure 17. Qualitative results on ImageNet class conditional generation with DDIM 25 steps by guiding unconditional ADM with ResNet50.

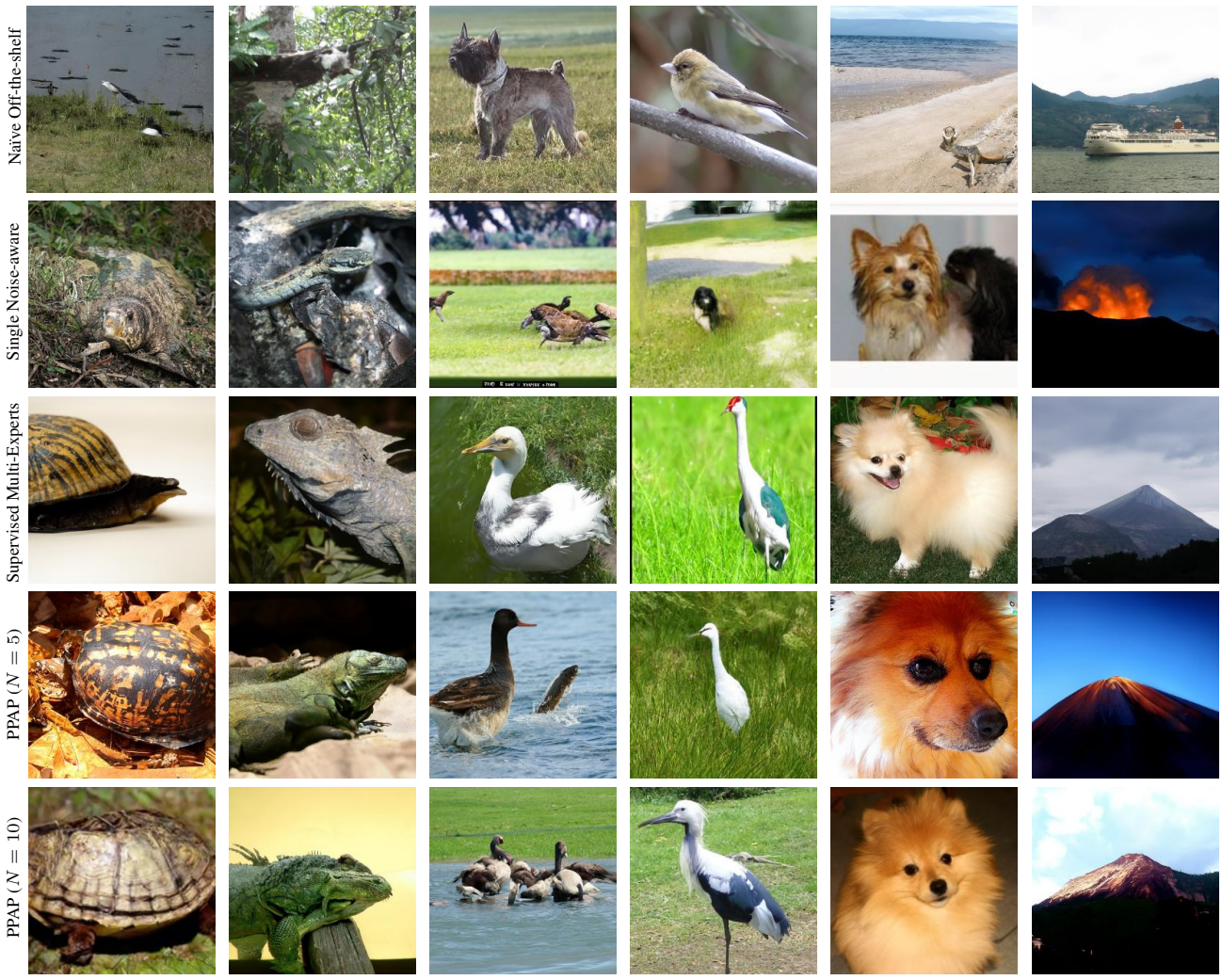


Figure 18. Qualitative results on ImageNet class conditional generation with DDPM 250 steps by guiding unconditional ADM with DeiT-S.

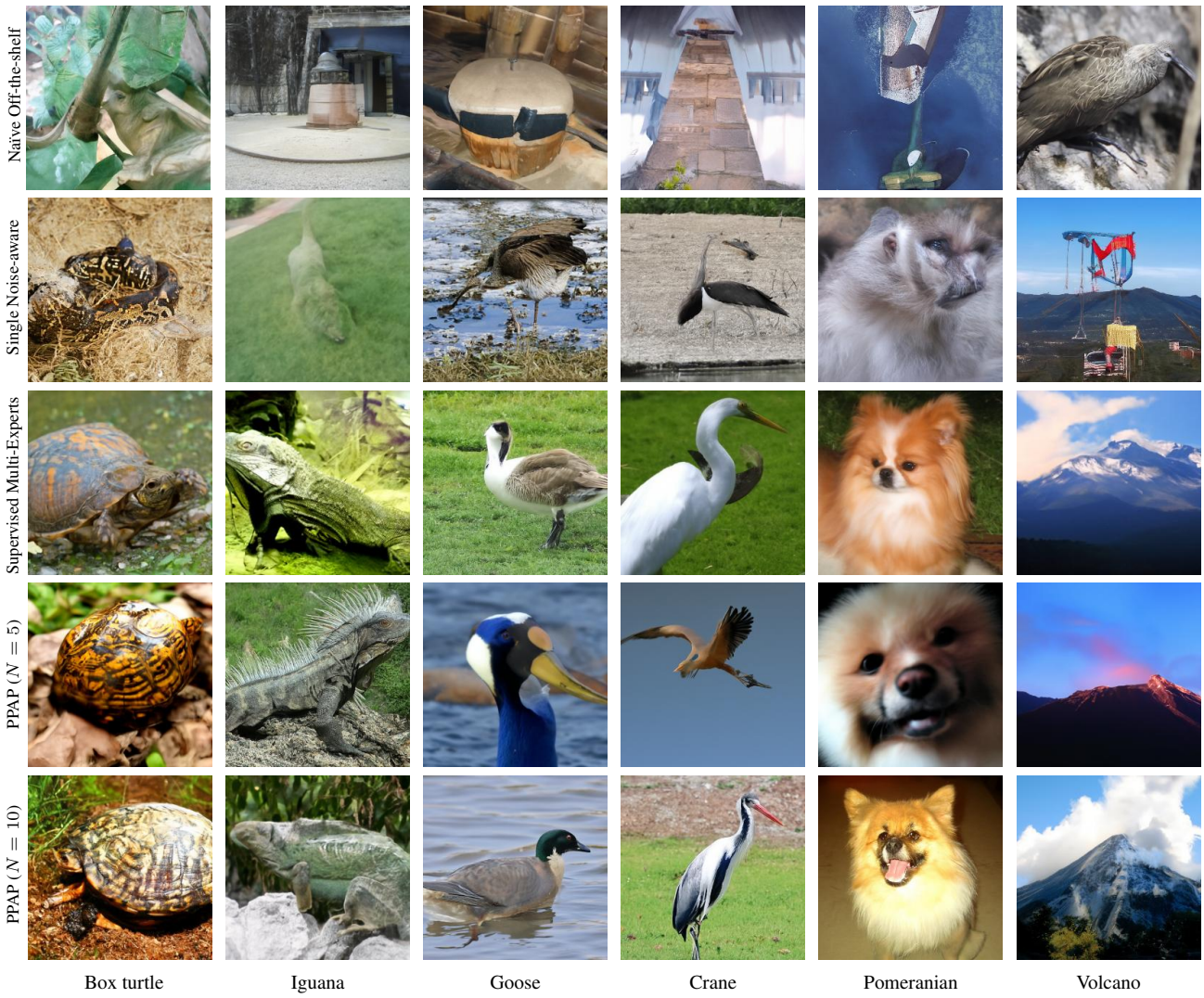


Figure 19. Qualitative results on ImageNet class conditional generation with DDIM 25 steps by guiding unconditional ADM with DeiT-S.

Figure 20. Generated images by guiding GLIDE with ResNet50 classifier. Our framework PPAP-5 succeeds in the guidance of diffusion, but the naïve off-the-shelf model fails.

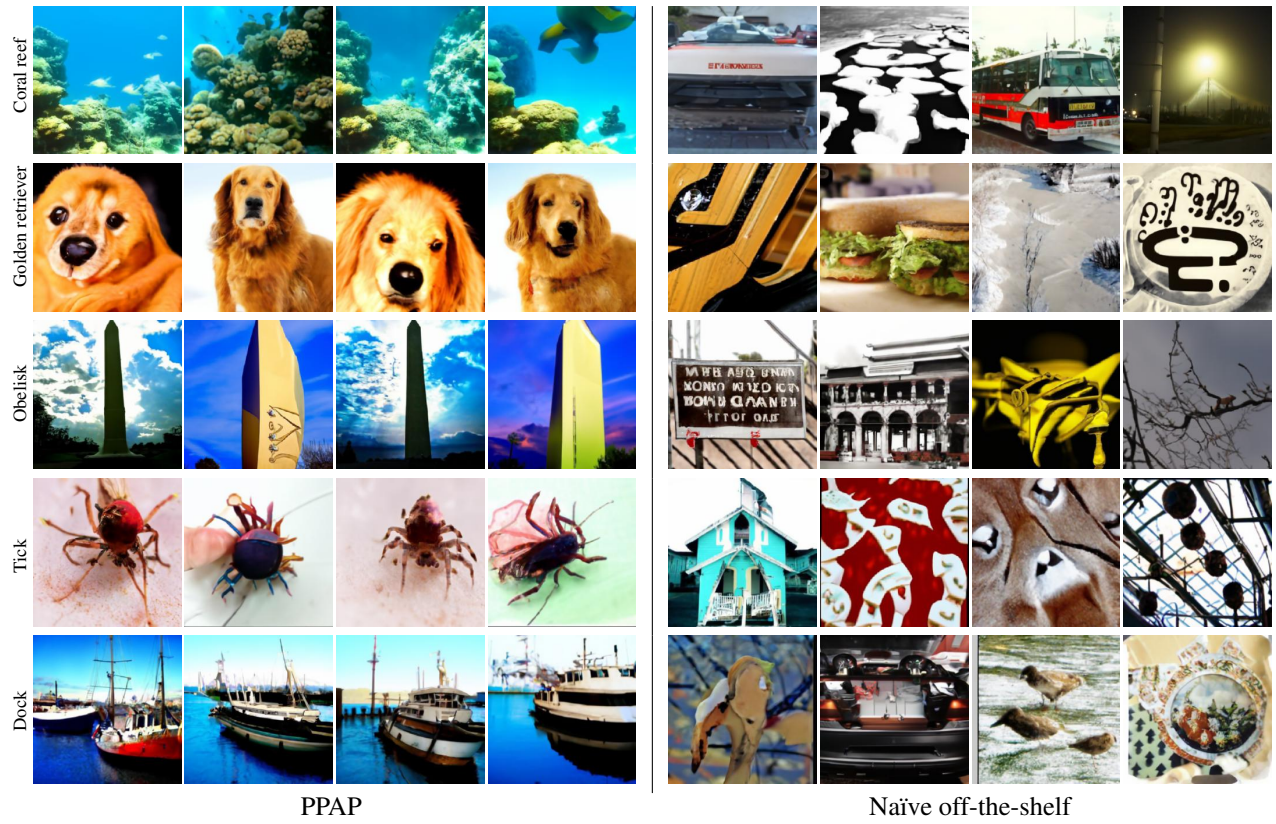


Figure 21. Qualitative results of in-class variations by guiding GLIDE with ResNet50 classifier. Our PPAP framework can generate proper images corresponding to the given class, but the naïve off-the-shelf fails.

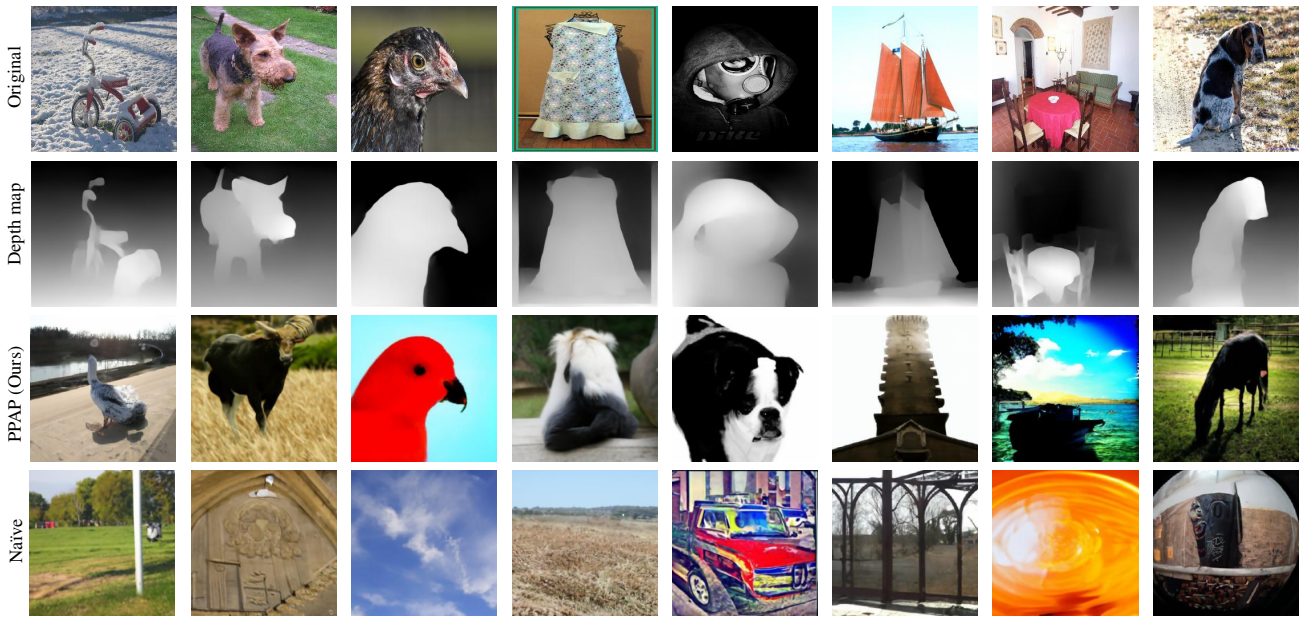


Figure 22. More qualitative results by guiding GLIDE with MiDaS depth estimation. Naïve off-the-shelf generates images that do not reflect the guidance with depth, whereas the proposed method succeeds. As shown in the third column, through the proposed framework, the generative model creates similar objects in the original image, but as we can see in the first and second columns, the images are generated by inferring the object only with depth.

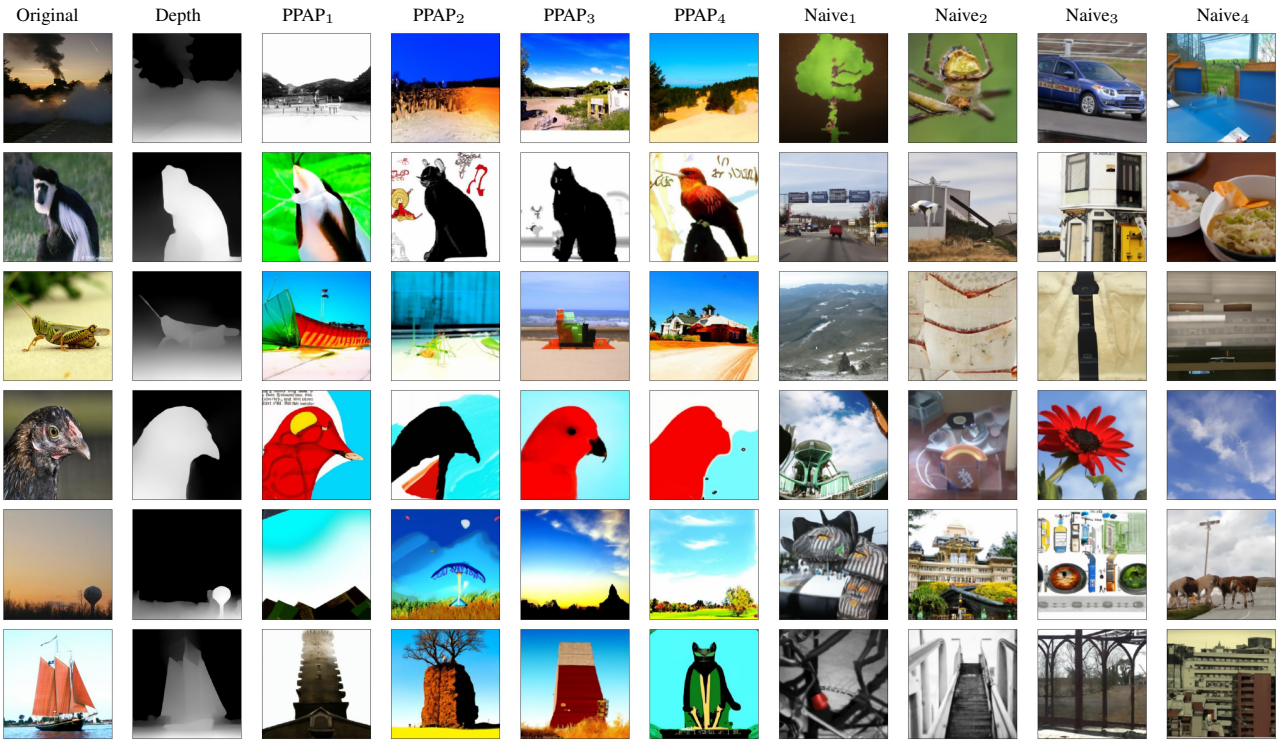


Figure 23. Qualitative results of in-batch variations by guiding GLIDE with MiDaS depth estimation. Our PPAP framework can generate proper images corresponding to the given depth, but off-the-shelf fails. PPAP generates not only various views of objects with a given depth as in the fourth row, but also diverse objects shown in the sixth row.

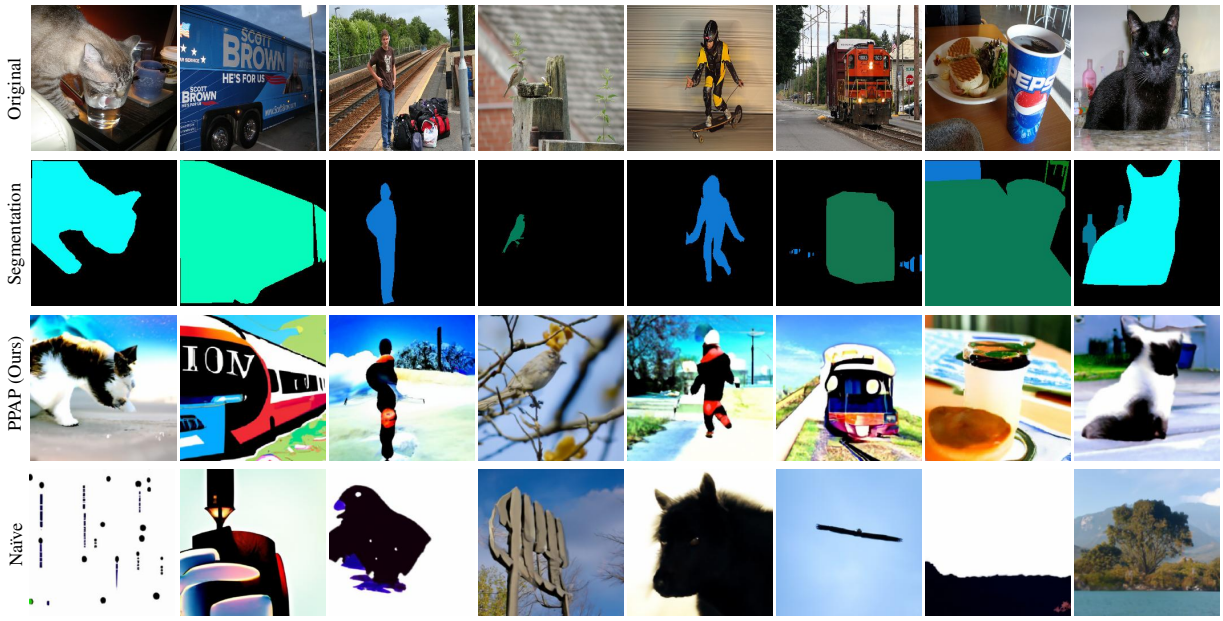


Figure 24. More qualitative results by guiding GLIDE with DeepLabv3 semantic segmentation. Our framework PPAP succeeds in the guidance of diffusion, but the naïve off-the-shelf model fails.

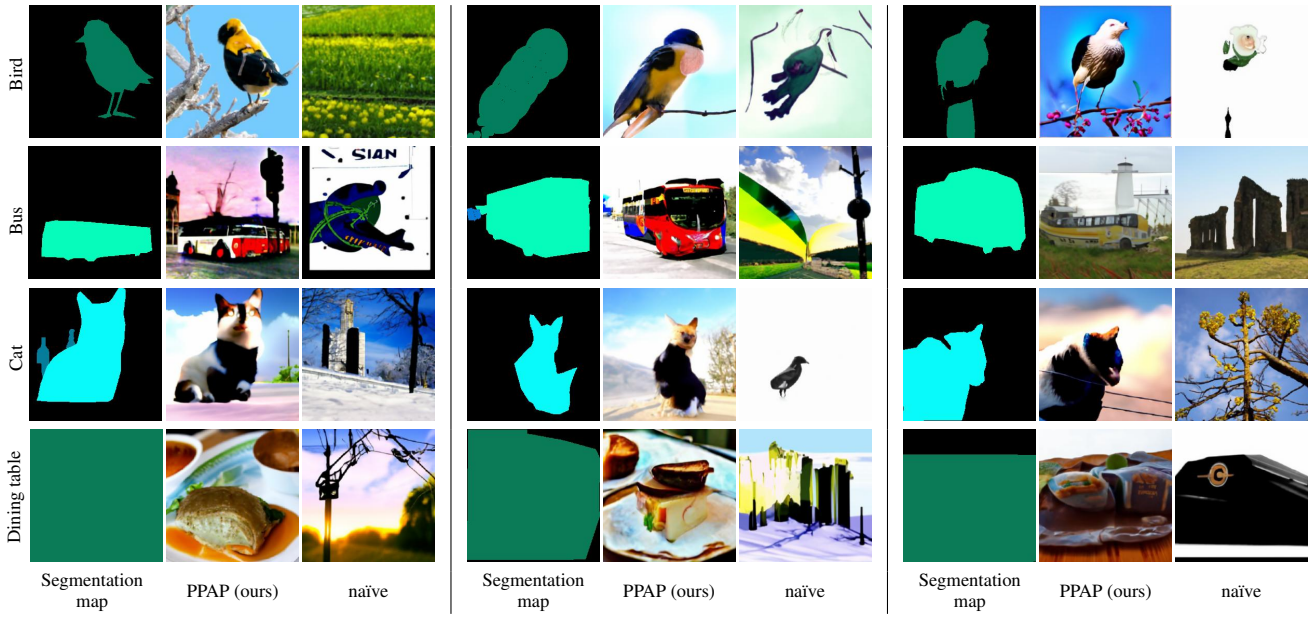


Figure 25. Qualitative results of in-class variations by guiding GLIDE with DeepLabv3 semantic segmentation. Our framework PPAP succeeds in the guidance of diffusion, but the naïve off-the-shelf model fails.

References

[1] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18208–18218, 2022. 2

[2] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022. 2

[3] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 3

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

[5] Alexandros Graikos, Nikolay Malkin, Nebojsa Jojic, and Dimitris Samaras. Diffusion models as plug-and-play priors. *arXiv preprint arXiv:2206.09012*, 2022. 2

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2, 3, 5

[7] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 2, 3

[8] Heeseung Kim, Sungwon Kim, and Sungroh Yoon. Guided-tts: A diffusion model for text-to-speech via classifier guidance. In *International Conference on Machine Learning*, pages 11119–11133. PMLR, 2022. 4

[9] Andrew Liu, Richard Tucker, Varun Jampani, Ameesh Makadia, Noah Snavely, and Angjoo Kanazawa. Infinite nature: Perpetual view generation of natural scenes from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14458–14467, 2021. 3

[10] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 2, 4

[11] Byeongjun Park, Hyojun Go, and Changick Kim. Bridging implicit and explicit geometric transformations for single-image view synthesis. *arXiv preprint arXiv:2209.07105*, 2022. 3

[12] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 2, 4

[13] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 3

[14] Robin Rombach, Patrick Esser, and Björn Ommer. Geometry-free view synthesis: Transformers and no 3d priors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14356–14366, 2021. 3

[15] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 3

[16] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 2

[17] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. 4