Self-Guided Diffusion Models

Vincent Tao Hu[*] **David W. Zhang**[*]
University of Amsterdam

Yuki M. Asano
University of Amsterdam

Gertjan J. Burghouts
TNO

Cees G. M. Snoek
University of Amsterdam

Abstract

Diffusion models have demonstrated remarkable progress in image generation quality, especially when guidance is used to control the generative process. However, guidance requires a large amount of image-annotation pairs for training and is thus dependent on their availability, correctness and unbiasedness. In this paper, we eliminate the need for such annotation by instead leveraging the flexibility of self-supervision signals to design a framework for *self-guided* diffusion models. By leveraging a feature extraction function and a self-annotation function, our method provides guidance signals at various image granularities: from the level of holistic images to object boxes and even segmentation masks. Our experiments on single-label and multi-label image datasets demonstrate that self-labeled guidance always outperforms diffusion models without guidance and may even surpass guidance based on ground-truth labels, especially on unbalanced data. When equipped with self-supervised box or mask proposals, our method further generates visually diverse yet semantically consistent images, without the need for any class, box, or segment label annotation. Self-guided diffusion is simple, flexible and expected to profit from deployment at scale.

1 Introduction

The image fidelity of diffusion models is spectacularly enhanced by conditioning on class labels [12]. Classifier guidance goes a step further and offers control over the alignment with the class label, by using the classifier gradient to guide the image generation [12]. Classifier-free guidance [19] replaces the dedicated classifier with a diffusion model that is trained by randomly setting the condition to the special *non-label* class. This has proven a fruitful research line for several other condition modalities, such as text [38; 32], image layout [36], visual neighbors [1], and image features [14]. However, all these conditioning and guidance methods require ground-truth annotations. This is an unrealistic and too costly assumption in many domains. For example, medical images require domain experts to annotate very high-resolution data, which is infeasible to do exhaustively [29]. In this paper, we propose to remove the necessity of any ground-truth annotation for guidance diffusion models.

Inspired by progress in self-supervised learning [9; 7], in this paper, we propose *self-guided diffusion*, a framework for image generation using guided diffusion without the need for any annotated image-label pairs. The framework encompasses a feature extraction function and a self-annotation function, that are compatible with recent self-supervised learning advances. Furthermore, we leverage the flexibility of self-supervised learning to generalize the guidance signal from the holistic image level to (unsupervised) local bounding boxes and segmentation masks for more fine-grained guidance. We demonstrate the potential of our proposal on single-label and multi-label image datasets, where self-labeled guidance always outperforms diffusion models without guidance and may even surpass guidance based on ground-truth labels. When equipped with self-supervised box or mask proposals,

[*] Equal contribution. taohu620@gmail.com, w.d.zhang@uva.nl

NeurIPS 2022 Workshop on Score-Based Methods.

our method further generates visually diverse yet semantically consistent images, without the need for any class, box, or segment label annotation.

2 Approach

Before detailing our self-guided diffusion framework, if readers are not familiar with diffusion models and the classifier-free guidance technique, please refer to Appendix A.

2.1 Self-Guided Diffusion

The equations describing diffusion by classifier-free guidance implicitly assume dataset $\mathcal{D}$ and its images each come with a single manually annotated class label. We prefer to make the label requirement explicit. We denote the human annotation process as the function $\xi(\mathbf{x}; \mathcal{D}, \mathcal{C}) : \mathcal{D} \rightarrow \mathcal{C}$, where $\mathcal{C}$ defines the annotation taxonomy, and plug this into Equation (7):

$$\tilde{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t;\ \xi(\mathbf{x};\ \mathcal{D}, \mathcal{C}), w) = (1-w)\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) + w\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t;\ \xi(\mathbf{x};\ \mathcal{D}, \mathcal{C})). \tag{1}$$

We propose to replace the supervised labeling process ξ with a self-supervised process that requires *no* human annotation:

$$\tilde{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t; f_\psi(g_\phi(\mathbf{x};\ \mathcal{D});\ \mathcal{D}), w) = (1-w)\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) + w\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t;\ f_\psi(g_\phi(\mathbf{x};\ \mathcal{D});\ \mathcal{D}), \tag{2}$$

where g is a self-supervised feature extraction function parameterized by ϕ that maps the input data to feature space $\mathcal{H}$, $g : \mathbf{x} \rightarrow g_\phi(\mathbf{x}), \forall \mathbf{x} \in \mathcal{D}$, and f is a self-annotation function parameterized by ψ to map the raw feature representation to the ultimate guidance signal $\mathbf{k}$, $f_\psi : g_\phi(\cdot; \mathcal{D}) \rightarrow \mathbf{k}$. The guidance signal $\mathbf{k}$ can be any form of *annotation*, e.g., label, box, pixel, that can be paired with an image, which we derive by $\mathbf{k}$=$f_\psi(g_\phi(\mathbf{x};\ \mathcal{D});\ \mathcal{D})$. The choice of the self-annotation function f can be non-parametric by heuristically searching over dataset $\mathcal{D}$ based on the extracted feature $g_\phi(\cdot;\ \mathcal{D})$, or parametric by fine-tuning on the feature map $g_\phi(\cdot;\ \mathcal{D})$.

For the noise prediction function $\boldsymbol{\epsilon}_\theta(\cdot)$, we adopt the traditional UNet network architecture [37] due to its superior image generation performance, following [20; 43; 33; 38].

Stemming from this general framework, we present three methods working at different spatial granularities, all without relying on any ground-truth labels. Specifically, we cover image-level, box-level and pixel-level guidance by setting the feature extraction function $g_\phi(\cdot)$, self-annotation function $f_\psi(\cdot)$, and guidance signal $\mathbf{k}$ to an approximate form.

Self-labeled guidance. To achieve self-labeled guidance, we need a self-annotation function f that produces a representative guidance signal $\mathbf{k} \in \mathbb{R}^K$. Firstly, we need an embedding function $g_\phi(\mathbf{x}), \mathbf{x} \in \mathcal{D}$ which provides semantically meaningful image-level guidance for the model. We obtain $g_\phi(\cdot)$ in a self-supervised manner by mapping from image space, $g_\phi(\cdot) : \mathbb{R}^{W\times H\times 3} \rightarrow \mathbb{R}^C$, where W and H are image width and height and C is the feature dimension. We may use any type of feature for the feature embedding function g, which we will vary and validate in the experiments. As the image-level feature $g_\phi(\cdot;\ \mathcal{D})$ is not compact enough for guidance, we further conduct a non-parametric clustering algorithm, e.g., k-means, as our self-annotation function f. For all features $g_\phi(\cdot)$, we obtain the self-labeled guidance via self-annotation function $f_\psi(\cdot) : \mathbb{R}^C \rightarrow \mathbb{R}^K$. Motivated by Rolfe [35], we use a one-hot embedding $\mathbf{k} \in \mathbb{R}^K$ for each image to achieve a compact guidance.

We inject the guidance information into the noise prediction function $\boldsymbol{\epsilon}_\theta$ by concatenating it with timestep embedding t and feed the concatenated information $\texttt{concat}[t, \mathbf{k}]$ into every block of the UNet. Thus, the noise prediction function $\boldsymbol{\epsilon}_\theta$ is rewritten as:

$$\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t;\ \mathbf{k}) = \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \texttt{concat}[t,\ \mathbf{k}]), \tag{3}$$

where $\mathbf{k}$=$f_\psi(g_\phi(\mathbf{x};\ \mathcal{D});\ \mathcal{D})$ is the self-annotated image-level guidance signal. For simplicity, we ignore the self-annotation function $f_\psi(\cdot)$ here and in the later text. Self-labeled guidance focuses on the image-level global guidance. For more fine-grained spatial guidance e.g., box, mask, please check Appendix B.

3 Experiments

Evaluation metric & Baselines & implementation details. To evaluate both diversity and fidelity of samples, we use the Fréchet Inception Distance (FID) [18] and Inception Score (IS) [39] as our

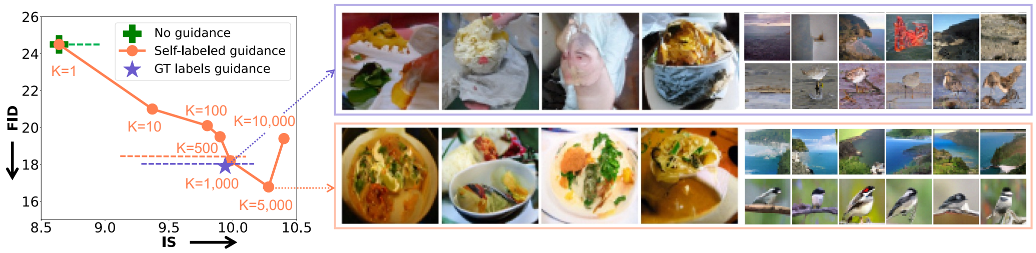


Figure 1: **Effect of number of clusters.** Self-labeled guidance outperforms DDPM without any guidance beyond a single cluster, is competitive with classifier-free guidance beyond 1,000 clusters and is even able to outperform guidance by ground-truth (GT) labels for 5,000 clusters. We visualize generated samples from ImageNet64 (middle) and ImageNet32 (right) for ground-truth labels guidance (top) and self-labeled guidance (bottom). More qualitative results in Appendix Figure 14.

main metrics. Throughout our experiments, we always use the diffusion model following [20]. We train with timestep T=1000, guidance drop probability p=0.1 and the linear variance schedule. For sampling, we set the guidance strength w=2 and deploy a clipping operation in every sampling timestep. As the baseline for a guided diffusion model with ground-truth labels we follow classifier-free guidance [19]. We use DDIM [42] samplers with 250 steps, σ_t=0. All hyperparameter of our self-guided diffusion and the baselines are the same, allowing us to compare methods under a fixed compute budget. For details of the learning rate, optimizer, and hyperparameters, we refer to Appendix F. All code will be released.

3.1 Self-labeled Guidance

We use ImageNet32/64 [11] to validate the efficacy of self-labeled guidance. For better evaluation of sampling quality, we also adopt the Inception Score (IS) [39], following the common practice on this dataset [12; 22; 5]. IS measures how well a model fits into the full ImageNet class distribution.

Table 1: **Choice of feature extraction function** on ImageNet32. DINO and MSN `ViT-B/16` obtain good trade-offs between FID and IS.

	FID↓	IS↑
Label-supervised		
ResNet50	22.00	8.23
ViT-B/16	22.30	7.81
Self-supervised		
MAE ViTBase	32.58	8.20
SimCLR-v2	23.96	9.35
MSN ViT-B/16	21.16	**10.59**
DINO ViT-B/16	**19.35**	10.41

Choice of feature extraction function *g*. We first measure the influence of the feature extraction function g used before clustering. We consider two supervised feature backbones: `ResNet50` [16] and `ViT-B/16` [13], and four self-supervised backbones: `SimCLR` [9], `MAE` [17], `MSN` [2] and `DINO` [7]. To assure a fair comparison we use 10k clusters for all architectures. From the results in Table 1, we make the following observations. First, features from the supervised `ResNet50`, and `ViT-B/16` lead to a satisfactory FID performance, at the expense of relatively limited diversity (low IS). However, they still require label annotation, which we strive to avoid in our work. Second, among the self-supervised feature extraction functions, the `MSN`- and `DINO`-pretrained ViT backbones have the best trade-off in terms of both FID and IS. They even improve over the label-supervised backbones. This implies the label assignment for the guidance is not unique, pretext labels on top of self-supervised learning can still provide influential guidance signal in comparison with human annotated labels. Also, the diversity of label-supervised `ViT-B/16` is much lower than self-supervised `ViT-B/16` with an IS of 7.81 vs. 10.41, suggesting that self-supervised guidance leads to more unbiased representation in comparison to supervised guidance. From now on we pick the `DINO ViT-B/16` architecture as our self-supervised feature extraction function g.

Effect of number of clusters. Next, we ablate the influence of the number of clusters on the overall sampling quality. We consider 1, 10, 100, 500, 1,000, 5,000 and 10,000 clusters on the extracted `CLS` token from the `DINO ViT-B/16` feature. For efficient, yet uniform, comparison we only run 20 epochs on ImageNet32. To put our sampling results in perspective, we also provide FID

Table 2: **Self-labeled comparisons on ImageNet32/64.** Self-labelled guidance surpasses the no-guidance baseline by a large margin on both datasets and even outperforms the guided diffusion model trained using ground-truth class labels.

		ImageNet32		**ImageNet64**	
Diffusion Method	**Annotation-free?**	FID↓	IS ↑	FID↓	IS↑
Ground-truth labels Guidance	✗	10.2	19.0	16.8	18.6
No Guidance	✓	14.3	10.8	36.1	10.4
Self-labeled Guidance (***this paper***)	✓	**7.3**	**20.3**	**12.1**	**23.1**

and IS results for DDPM and classifier-free guidance. From the result in Figure 1 we first observe that when the cluster number is ranging from 1 to 5,000, our model's performance monotonously increases and always surpasses the DDPM model. Beyond 1,000 clusters, we are competitive with classifier-free guidance using ground-truth labels. For 5,000 clusters, there is a sweet spot where we outperform classifier-free guidance with an FID of 16.4 vs. 17.9 and an IS of 9.94 vs. 10.35, see also the generated images in Figure 1. The performance of FID starts to deteriorate from 5,000 to 10,000 clusters. We conclude that self-labeled guidance outperforms DDPM without any guidance beyond a single cluster, is competitive with classifier-free guidance beyond 1,000 clusters, and is even able to outperform guidance by ground-truth labels for 5,000 clusters. From now on we use 5,000 clusters for self-labeled guidance on ImageNet.

Varying guidance strength w. Next we consider the influence of the guidance strength w on our sampling results. As the validation set of ImageNet32 is strictly balanced, we also consider an unbalanced setting which is more similar to real-world deployment. Under both settings we compare the FID between our self-labeled guidance and ground-truth guidance. We train both models for 100 epochs. For the standard ImageNet32 validation setting in Figure 2a, our method achieves a 17.8% improvement for respective optimal guidance strength of the two methods. Self-labeled guidance is especially effective for lower values of w. We observe similar trends for the unbalanced setting in Figure 2b, be it that the overall FID results are slightly higher for both methods. The improvement increases to 18.7%. We conjecture this is due to the unbalanced nature of the k-means algorithm [24], and clustering based on the statistics of the overall dataset can potentially lead to more robust performance in unbalanced setting.

Self-labeled comparisons on ImageNet32/64. We compare our self-labeled guidance with ground-truth labels guidance, which utilizes the technique of classifier-free guidance [19]. We train all experiments for 100 epochs which take about 6 days to converge on four RTX A5000 GPUs. All hyperparameters are the same between the two methods to make the comparison as fair as possible. Results on ImageNet32 and ImageNet64 are in Table 2. Similar to Dhariwal & Nichol [12], we observe that any guidance setting improves considerably over the unconditional & no-guidance model. Surprisingly, our self-labeled model even outperforms the ground-truth labels by a large gap in terms of FID of 2.9 and 4.7 points respectively. We hypothesize that the ground-truth taxonomy might be suboptimal for learning generative models and the self-supervised clusters offer a better guidance signal due to better alignment with the visual similarity of the images. This suggests that the label-conditioned guidance from [19] can be completely replaced by guidance from self-supervision, which would enable guided diffusion models to learn from even larger (unlabeled) datasets than feasible today.

4 Conclusion

We have explored the potential of self-supervision signals for diffusion models and propose a framework for self-guided diffusion models. By leveraging a feature extraction function and a self-annotation function, our framework provides guidance signals at various image granularities: from the level of holistic images to object boxes and even segmentation masks. Our experiments indicate that self-supervision signals are an adequate replacement for existing guidance methods that generate images by relying on annotated image-label pairs during training. Furthermore, both self-boxed and self-segmented approaches demonstrate that we can acquire fine-grained control over the image content, without any ground-truth bounding boxes or segmentation masks. Ultimately, our goal is to enable the benefits of self-guided diffusion for unlabeled and more diverse datasets at scale, wherein we believe this work is a promising first step.

Acknowledgments

The work of DWZ is part of the research programme Perspectief EDL with project number P16-25 project 3, which is financed by the Dutch Research Council (NWO) domain Applied and Engineering Sciences (TTW).

References

[1] Oron Ashual, Shelly Sheynin, Adam Polyak, Uriel Singer, Oran Gafni, Eliya Nachmani, and Yaniv Taigman. Knn-diffusion: Image generation via large-scale retrieval. *arXiv preprint arXiv:2204.02849*, 2022.

[2] 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. *arXiv preprint arXiv:2204.07141*, 2022.

[3] Shane Barratt and Rishi Sharma. A note on the inception score. *arXiv preprint arXiv:1801.01973*, 2018.

[4] Andreas Blattmann, Robin Rombach, Kaan Oktay, and Björn Ommer. Retrieval-augmented diffusion models. *arXiv preprint arXiv:2204.11824*, 2022.

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, 2020.

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *International Conference on Computer Vision*, 2021.

[8] Arantxa Casanova, Marlene Careil, Jakob Verbeek, Michal Drozdzal, and Adriana Romero Soriano. Instance-conditioned gan. In *Advances in Neural Information Processing Systems*, 2021.

[9] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. In *Advances in Neural Information Processing Systems*, 2020.

[10] Jang Hyun Cho, Utkarsh Mall, Kavita Bala, and Bharath Hariharan. Picie: Unsupervised semantic segmentation using invariance and equivariance in clustering. In *Computer Vision and Pattern Recognition*, 2021.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition*, 2009.

[12] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, 2021.

[13] 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. *International Conference on Learning Representations*, 2021.

[14] Giorgio Giannone, Didrik Nielsen, and Ole Winther. Few-shot diffusion models. *arXiv preprint arXiv:2205.15463*, 2022.

[15] Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T Freeman. Unsupervised semantic segmentation by distilling feature correspondences. In *International Conference on Learning Representations*, 2022.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition*, 2016.

[17] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Computer Vision and Pattern Recognition*, 2022.

[18] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017.

[19] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshop*, 2021.

[20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

[21] Xu Ji, Joao F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *International Conference on Computer Vision*, 2019.

[22] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Computer Vision and Pattern Recognition*, 2019.

[23] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, 2020.

[24] Felix Last, Georgios Douzas, and Fernando Bacao. Oversampling for imbalanced learning based on k-means and smote. *arXiv preprint arXiv:1711.00837*, 2017.

[25] 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 *European Conference on Computer Vision*, 2014.

[26] Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision*, 2022.

[27] Jiachen Lu, Jinghan Yao, Junge Zhang, Xiatian Zhu, Hang Xu, Weiguo Gao, Chunjing Xu, Tao Xiang, and Li Zhang. Soft: softmax-free transformer with linear complexity. In *Advances in Neural Information Processing Systems*, 2021.

[28] Anton Obukhov, Maximilian Seitzer, Po-Wei Wu, Semen Zhydenko, Jonathan Kyl, and Elvis Yu-Jing Lin. High-fidelity performance metrics for generative models in pytorch, 2020. URL `https://github.com/toshas/torch-fidelity`. Version: 0.3.0, DOI: 10.5281/zenodo.4957738.

[29] Andreas Panteli, Jonas Teuwen, Hugo Horlings, and Efstratios Gavves. Sparse-shot learning with exclusive cross-entropy for extremely many localisations. In *International Conference on Computer Vision*, 2021.

[30] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In *Computer Vision and Pattern Recognition*, 2022.

[31] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Computer Vision and Pattern Recognition*, 2022.

[32] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, 2021.

[33] 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.

[34] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015.

[35] Jason Tyler Rolfe. Discrete variational autoencoders. In *International Conference on Learning Representations*, 2017.

[36] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Computer Vision and Pattern Recognition*, 2022.

[37] O Ronneberger, P Fischer, and T Brox. Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, 2015.

[38] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *arXiv preprint arXiv:2205.11487*, 2022.

[39] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, 2016.

[40] Oriane Siméoni, Gilles Puy, Huy V Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce. Localizing objects with self-supervised transformers and no labels. In *British Machine Vision Conference*, 2021.

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

[42] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[43] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

[44] Huy V Vo, Patrick Pérez, and Jean Ponce. Toward unsupervised, multi-object discovery in large-scale image collections. In *European Conference on Computer Vision*, 2020.

[45] Van Huy Vo, Elena Sizikova, Cordelia Schmid, Patrick Pérez, and Jean Ponce. Large-scale unsupervised object discovery. In *Advances in Neural Information Processing Systems*, 2021.

[46] Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. Dense siamese network. In *European Conference on Computer Vision*, 2022.

Contents

1 Introduction **1**

2 Approach **2**

2.1 Self-Guided Diffusion 2

3 Experiments **2**

3.1 Self-labeled Guidance 3

4 Conclusion **4**

A Background **9**

B Extra Method **9**

C Extra results. **10**

C.1 Self-boxed Guidance 10

C.2 Self-segmented Guidance 11

D Main framework **13**

E More quantitative results **14**

E.1 Correlation between NMI and FID in different feature backbones. 14

E.2 Precision and Recall in ImageNet32/64 dataset 14

E.3 Cluster number ablation in self-boxed guidance 14

E.4 Trend visualization of training loss and validation FID 14

F More experimental details **15**

F.1 UNet structure 16

F.2 Training Parameter 16

F.3 Dataset preparation 17

F.4 LOST, STEGO algorithms 17

G More qualitative results **17**

A Background

Diffusion models. Diffusion models [41; 20] gradually add noise to an image $\mathbf{x}_0$ until the original signal is fully diminished. By learning to reverse this process one can turn random noise $\mathbf{x}_T$ into images. This diffusion process is modeled as a Gaussian process with Markovian structure:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}), \quad q(\mathbf{x}_t|\mathbf{x}_0) := \mathcal{N}(\mathbf{x}_t; \sqrt{\overline{\alpha}_t}\mathbf{x}_0, (1-\overline{\alpha}_t)\mathbf{I}, \tag{4}$$

where $\beta_1, \ldots, \beta_T$ is a fixed variance schedule on which we define $\alpha_t := 1 - \beta_t$ and $\overline{\alpha}_t := \prod_{s=1}^{t} \alpha_s$. All latent variables have the same dimensionality as the image $\mathbf{x}_0$ and differ by the proportion of the retained signal and added noise.

Learning the reverse process reduces to learning a denoiser $\mathbf{x}_t \sim q(\mathbf{x}_t|\mathbf{x}_0)$ that recovers the original image as $(\mathbf{x}_t - (1-\overline{\alpha}_t)\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t))/\sqrt{\overline{\alpha}_t} \approx \mathbf{x}_0$. [20] optimize the network parameters $\boldsymbol{\theta}$ by minimizing the error of the noise prediction:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\epsilon},\mathbf{x},t}\left[||\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) - \boldsymbol{\epsilon}||_2^2\right], \tag{5}$$

in which $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\mathbf{x} \in \mathcal{D}$ is a sample from the training dataset $\mathcal{D}$ and the noise prediction function $\boldsymbol{\epsilon}_\theta(\cdot)$ are encouraged to be as close as possible to $\boldsymbol{\epsilon}$.

The standard sampling [20] requires many neural function evaluations to get good quality samples. Instead, the faster Denoising Diffusion Implicit Models (DDIM) sampler [42] has a non-Markovian sampling process:

$$\mathbf{x}_{t-1} = \sqrt{\overline{\alpha}_{t-1}}\left(\frac{\mathbf{x}_t - \sqrt{1-\overline{\alpha}_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)}{\sqrt{\overline{\alpha}_t}}\right) + \sqrt{1-\overline{\alpha}_{t-1}-\sigma_t^2} \cdot \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) + \sigma_t\boldsymbol{\epsilon}, \tag{6}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is standard Gaussian noise independent of $\boldsymbol{x}_t$.

Classifier-free guidance. To trade off mode coverage and sample fidelity in a conditional diffusion model, Dhariwal & Nichol [12] propose to guide the image generation process using the gradients of a classifier, with the additional cost of having to train the classifier on noisy images. Motivated by this drawback, [19] introduce label-conditioned guidance that does not require a classifier. They obtain a combination of a conditional and unconditional network in a single model, by randomly dropping the guidance signal $\mathbf{c}$ during training. After training, it empowers the model with progressive control over the degree of alignment between the guidance signal and the sample by varying the guidance strength w:

$$\tilde{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t;\ \mathbf{c}, w) = (1-w)\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) + w\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t;\ \mathbf{c}). \tag{7}$$

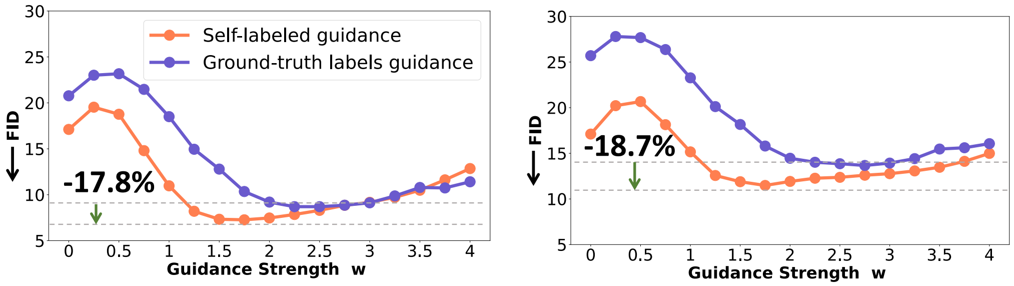
A larger w leads to greater alignment with the guidance signal, and vice versa. Classifier-free guidance [19] provides progressive control over the specific guidance direction at the expense of labor-consuming data annotation. In this paper, we propose to remove the necessity of data annotation using a self-guided principle based on self-supervised learning.

B Extra Method

Self-boxed guidance. Bounding boxes specify the location of an object in an image [34; 6] and complements the content information of class labels. Our self-boxed guidance approach aims to attain this signal via self-supervised models. We represent the bounding box as a feature map ($W \times H$) rather than coordinates ($[X, Y, W, H]$). We propose the self-annotation function f that obtains a bounding box $\mathbf{k}_s \in \mathbb{R}^{W \times H}$ by mapping from feature space $\mathcal{H}$ to the bounding box space via $f_\psi(\cdot;\ \mathcal{D}) : \mathbb{R}^{W \times H \times C} \to \mathbb{R}^{W \times H}$, and inject the guidance signal by concatenating in the channel dimension: $\mathbf{x}_t := \mathtt{concat}[\mathbf{x}_t, \mathbf{k}_s]$ Usually in self-supervised learning, the derived bounding box is class-agnostic [44; 45]. To inject a self-supervised pseudo label to further enhance the guidance signal, we again resort to clustering to obtain $\mathbf{k}$ and concatenate it with the time embedding $t := \mathtt{concat}[t, \mathbf{k}]$. To incorporate such guidance, we reformulate the noise prediction function $\boldsymbol{\epsilon}_\theta$ as:

$$\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t;\ \mathbf{k}_s, \mathbf{k}) = \boldsymbol{\epsilon}_\theta(\mathtt{concat}[\mathbf{x}_t, \mathbf{k}_s], \mathtt{concat}[t, \mathbf{k}]), \tag{8}$$

in which $\mathbf{k}_s$ is the self-supervised box guidance obtained by self-annotation functions f_ψ, $\mathbf{k}$ is the self-supervised image-level guidance from clustering. The design of f_ψ is flexible as long as it obtains self-supervised bounding boxes by $f_\psi(\cdot;\ \mathcal{D}) : \mathbb{R}^{W \times H \times C} \to \mathbb{R}^{W \times H}$. Self-boxed guidance guides the diffusion model by boxes, which specifies the box area in which the object will be generated. Sometimes, we may need an even finer granularity, e.g., pixels, which we detail next.



(a) Setting I: ImageNet32 balanced

(b) Setting II: ImageNet32 unbalanced

Figure 2: **Varying guidance strength w.** Self-labeled guidance surpasses the guidance based on ground-truth labels for both (a) ImageNet32 balanced and (b) ImageNet32 unbalanced. The dotted gray line indicates the best achieved performance of both methods under various guidance strength. The difference between them is slightly more prominent for unbalanced data, we conjecture that is because our self-labeled guidance is obtained by clustering based on the statistics of the overall dataset, which can potentially lead to more robust performance in unbalanced setting.

Self-segmented guidance. Compared to a bounding box, a segmentation mask is a more fine-grained signal. Additionally, a multichannel mask is more expressive than a binary foreground-background mask. Therefore, we propose a self-annotation function f that acts as a plug-in built on feature $g_\phi(\cdot;\ \mathcal{D})$ to extract the segmentation mask $\mathbf{k}_s$ via function mapping $f_\psi(\cdot;\ \mathcal{D}) : \mathbb{R}^{W\times H\times C} \to \mathbb{R}^{W\times H\times K}$, where K is the number of segmentation clusters.

To inject the self-segmented guidance into the noise prediction function $\boldsymbol{\epsilon}_\theta$, we consider two pathways for injection of such guidance. We first concatenate the segmentation mask to $\mathbf{x}_t$ in the channel dimension, $\mathbf{x}_t := \texttt{concat}[\mathbf{x}_t,\ \mathbf{k}_s]$, to retain the spatial inductive bias of the guidance signal. Secondly, we also incorporate the image-level guidance to further amplify the guidance signal along the channel dimension. As the segmentation mask from the self-annotation function f_ψ already contains image-level information, we do not apply the image-level clustering as before in our self-labeled guidance. Instead, we directly derive the image-level guidance from the self-annotation result $f_\psi(\cdot)$ via spatial maximum pooling: $\mathbb{R}^{W\times H\times K} \to \mathbb{R}^K$, and feed the image-level guidance $\hat{\mathbf{k}}$ into the noise prediction function via concatenating it with the timestep embedding t:=$\texttt{concat}[t, \hat{\mathbf{k}}]$. The concatenated results will be sent to every block of the UNet. In the end, the overall noise prediction function for self-segmented guidance is formulated as:

$$\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t;\ \mathbf{k}_s, \hat{\mathbf{k}}) = \boldsymbol{\epsilon}_\theta(\texttt{concat}[\mathbf{x}_t, \mathbf{k}_s],\ \texttt{concat}[t, \hat{\mathbf{k}}]), \tag{9}$$

in which $\mathbf{k}_s$ is the spatial mask guidance obtained from self-annotation function f, $\hat{\mathbf{k}}$ is a multi-hot image-level guidance derived from the self-supervised learning mask $\mathbf{k}_s$.

We have described three variants of self-guidances by setting the feature extraction function $g_\phi(\cdot)$, self-annotation function $f_\psi(\cdot)$, guidance signal $\mathbf{k}$ to an approximate form. In the end, we arrive at three noise prediction functions $\boldsymbol{\epsilon}_\theta$, which we utilize for diffusion model training and sampling, following the standard guided [19] diffusion approach as detailed in Appendix A.

C Extra results.

C.1 Self-boxed Guidance

We report on Pascal VOC and COCO_20K to validate the efficacy of self-boxed guidance. To obtain class-agnostic object bounding boxes, we use LOST [40] as our self-annotation function f. We report train FID for Pascal VOC and train/validation FID for COCO_20K. For image-level clustering to attain the guidance signal, we empirically found k=100 works best on both datasets as those datasets are relatively small-scale in images and labels compared to ImageNet. We train our diffusion model for 800 epochs with input image size 64×64. See Appendix F for more details.

Self-boxed comparisons on Pascal VOC and COCO_20K. For the ground-truth labels guidance baseline, we condition on a class embedding. Since there are now multiple objects per image, we

Table 3: **Self-boxed comparisons on Pascal VOC and COCO_20K.** Self-boxed guidance outperforms the no-guidance baseline FID considerably for multi-label datasets and is even better than a label-supervised alternative.

Diffusion Method	**Annotation-free?**	**Pascal VOC**	**COCO_20K**
Ground-truth labels Guidance	✗	23.5	19.3
No Guidance	✓	58.6	42.5
Self-boxed Guidance (***this paper***)	✓	**18.4**	**16.0**

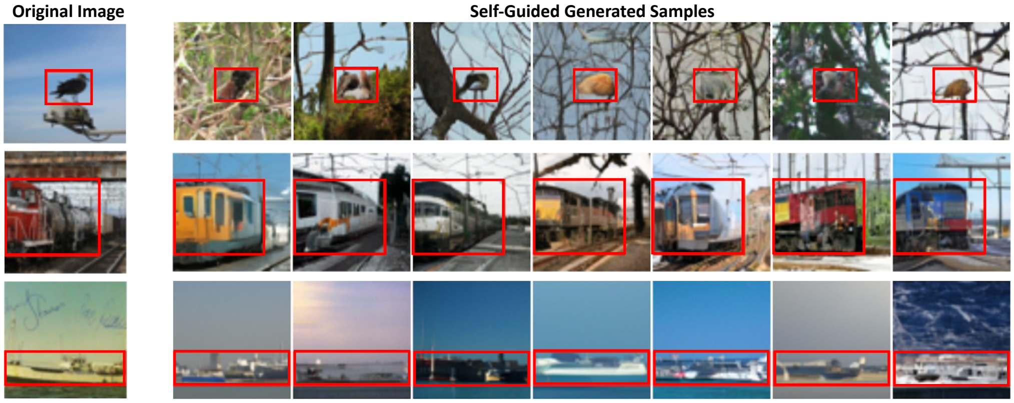


Figure 3: **Self-boxed guided diffusion results on Pascal VOC.** Each column is sampled using different random noise. Our method generates visually diverse and semantically consistent images.

represent the ground-truth class with a multi-hot embedding. Aside from the class embedding which is multi-hot in our method, all other settings remain the same for a fair comparison. The results in Table 3, confirm that the multi-hot class embedding is indeed effective for multi-label datasets, improving over the no-guidance model by a large margin. This improvement comes at the cost of manually annotating multiple classes per image. Self-boxed guidance further improves upon this result, by reducing the FID by an additional 5.1 and 3.3 points respectively without using any ground-truth annotation. In Figure 3, we show our method generates diverse and semantically well-aligned images.

C.2 Self-segmented Guidance

Finally, we validate the efficacy of self-segmented guidance on Pascal VOC and COCO-Stuff. For COCO-Stuff we follow the split from [15; 21; 10; 46], with a train set of 49,629 images and a validation set of 2,175 images. Classes are merged into 27 (15 stuff and 12 things) categories. For self-segmented guidance we apply `STEGO` [15] as our self-annotation function f. We set the cluster number to 27 for COCO-Stuff, and 21 for Pascal VOC, following `STEGO`. We train all models on images of size 64×64, for 800 epochs on Pascal VOC, and for 400 epochs on COCO-Stuff. We report the train FID for Pascal VOC and both train and validation FID for COCO-Stuff. More details on the dataset and experimental setup are provided in Appendix F.

Self-segmented comparisons on Pascal VOC and COCO-Stuff. We compare against both the ground-truth labels guidance baseline from the previous section and a model trained with ground-truth semantic masks guidance. The results in Table 4 demonstrate that our self-segmented guidance still outperforms the ground-truth labels guidance baseline on both datasets. The comparison between ground-truth labels and segmentation masks reveals an improvement in image quality when using the more fine-grained segmentation mask as the condition signal. But these segmentation masks are one of the most costly types of image annotations that require every pixel to be labeled. Our self-segmented approach avoids the necessity for annotations while narrowing the performance gap, and more importantly offering fine-grained control over the image layout. We demonstrate this controllability with examples in Figure 4. These examples further highlight a robustness against

Table 4: **Self-segmented comparisons on Pascal VOC and COCO-Stuff.** Any form of guidance results in a considerable FID reduction over the no-guidance model. Self-segmented guidance improves over ground-truth multi-hot labels guidance and narrows the gap with guidance by annotation-intensive ground-truth masks.

Diffusion Method	**Annotation-free?**	**Pascal VOC**	**COCO-Stuff**	
			Train	Val
Ground-truth labels Guidance	✗	23.5	16.3	20.5
No Guidance	✓	58.6	29.1	34.1
Self-segmented Guidance (***this paper***)	✓	**17.1**	**12.5**	**17.7**
Ground-truth masks	✗	12.5	8.1	11.2

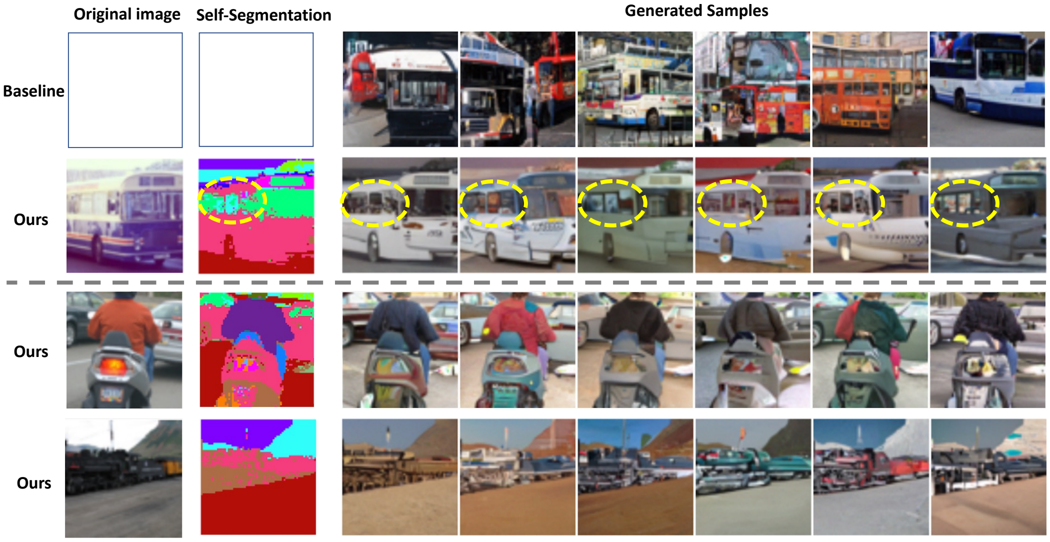


Figure 4: **Self-segmented guided diffusion results on Pascal VOC.** Each column is sampled using different random noise. The first two rows indicate our self-segmented guidance provides more fine-grained guidance than ground-truth labels guidance for the generation of bus images. Note how the noisy window-bar in the self-segmented mask (marked by dotted ellipse) still results in plausible window separations in the generated image samples. The two bottom rows shows our method creates visually diverse examples that are well aligned with the self-segmented guidance signal. We provide more examples in Appendix Figure 11.

noise in the segmentation masks, which our method acquires naturally due to training with noisy segmentations.

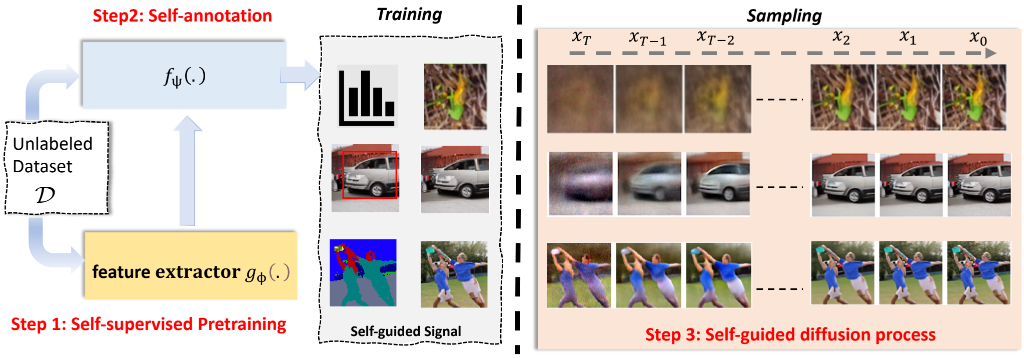

Step2: Self-annotation
$f_\psi(.)$
Unlabeled Dataset $\mathcal{D}$
feature extractor $g_\phi(.)$
Step 1: Self-supervised Pretraining
Training
Self-guided Signal
Sampling
x_T
x_{T-1}
x_{T-2}
x_2
x_1
x_0
Step 3: Self-guided diffusion process

Figure 5: **Main pipeline of our method.** Starting from unlabeled dataset $\mathcal{D}$, we apply unsupervised feature extractor $g_\psi(.)$ to post-processing function $f_\phi(.)$, which yield to different forms of self-guidance signal from image-level, to box, mask-level. Those freely-obtained signals can be further utilized for training and sampling stage of diffusion model.

D Main framework

We illustrate the pipeline of our framework in Figure 5.

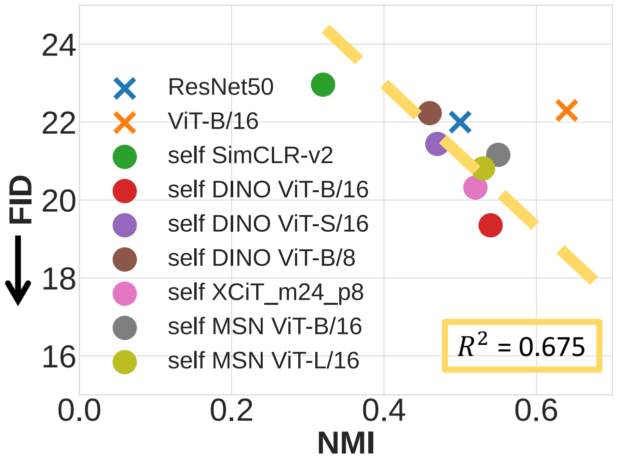


Figure 6: **Correlation between NMI and FID** on ImageNet32. The Normalized Mutual Information (NMI) is not related to FID for supervised backbones, while, for the self-supervised model, NMI and FID are negatively correlated. **Conclusion:** Self-supervised representation learning measures the progress via NMI, thus the (negative) correlation between NMI and FID suggest that future progress in self-supervised learning will also translate to improvements to self-labeled guidance.

Table 5: **Comparison with baseline on ImageNet32 and ImageNet64 dataset with FID, IS, Precision (P), Recall (R).**

Diffusion Method	**Annotation-free?**	**ImageNet32**				**ImageNet64**			
		FID↓	IS ↑	P ↑	R ↑	FID↓	IS↑	P↑	R ↑
Ground-truth labels Guidance	✗	10.2	19.0	0.71	0.62	16.8	18.6	0.71	**0.62**
No Guidance	✓	14.3	10.8	0.49	0.61	36.1	10.4	0.59	0.60
Self-labeled Gudiance (***this paper***)	✓	**7.3**	**20.3**	**0.77**	**0.63**	**12.1**	**23.1**	**0.78**	**0.62**

E More quantitative results

E.1 Correlation between NMI and FID in different feature backbones.

To assess the correlation between cluster quality and sample fidelity, we consider the Normalized Mutual Information (NMI), which is commonly adopted as a mutual information-derived metric to assess clustering quality based on provided ground truth labels. In Figure 6 we plot the connection between NMI and FID. For the label-supervised functions, the NMI is unrelated to the FID, but for the self-supervised functions, NMI and FID are negatively correlated, suggesting that NMI — a metric commonly applied to assess the quality of self-supervised methods — is also predictive of the model's usefulness in our setting.

E.2 Precision and Recall in ImageNet32/64 dataset

We show the extra results of ImageNet on precision and recall in Table 5. We follow the evaluation code of precision and recall from ICGAN [8], our self-labeled guidance also outperforms ground-truth labels in precision and remains competitive in recall.

E.3 Cluster number ablation in self-boxed guidance

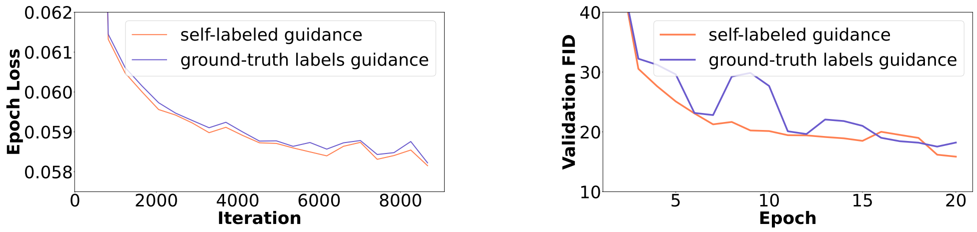
In table 6, we empirically evaluate the performance when we alter the cluster number in our self-boxed guidance. We find the performance will increase from $k = 21$ to $k = 100$, and saturated at $k = 100$.

E.4 Trend visualization of training loss and validation FID

We visualize the trend of training loss and validation FID in Figure 7.

Table 6: **Cluster number ablation on Pascal VOC dataset for self-boxed guidance.**

Cluster number k	FID ↓
21	22.5
50	18.6
100	18.5



(a) Epoch loss trend.

(b) Validation FID trend.

Figure 7: **Training epoch loss and and validation FID by total 20 epochs.** For epoch loss, our self-labeled guidance monotonously lower than ground-truth labels. For validation FID, our validation FID is more stable than our baseline.

F More experimental details

Training details. For our best results, we train 100 epochs on 4 GPUs of A5000 (24G) in ImageNet. We train 800/800/400 epochs on 1GPU of A6000 (48G) in Pascal VOC, COCO_20K, COCO-Stuff, respectively. All qualitative results in this paper are trained in the same setting as mentioned above. We train and evaluate the Pascal VOC, COCO_20K, COCO-Stuff in image size 64, and visualize them by bilinear up sampling to 256, following [26].

Sampling details. We sample the guidance signal from the distribution of training set in our all experiments. For each timestep, we need twice of Number of Forward Evaluation (NFE), we optimize them by concatenating the conditional and unconditional signal along the batch dimension so that we only need one time of NFE in every timestep.

Evaluation details. We use the common package Clean-FID [30], torch-fidelity [28] for FID, IS calculation, respectively. For IS, we use the standard 10-split setting, we only report IS on ImageNet, as it might be not an appropriate metric for non object-centric datasets [3]. For checking point, we pick the checking point every 10 epochs by minimal FID between generated sample set and train set.

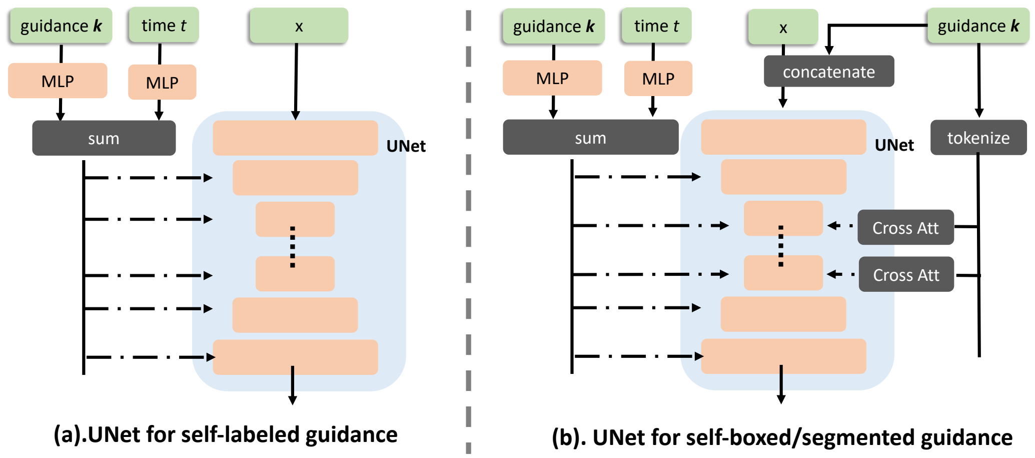


Figure 8: **The structure of UNet module.**

F.1 UNet structure

Guidance signal injection. We describe the detail of guidance signal injection in Figure 8. The injection of self-labeled guidance and self-boxed/segmented guidance is slightly different. The common part is by concatenation between timestep embedding and noisy input, the concatenated feature will be sent to every block of the UNet. For the self-boxed/segmented guidance, we not only conduct the information fusion as above, but also incorporate the spatial inductive-bias by concatenating it with input, the concatenated result will be fed into the UNet.

Timestep embedding. We embed the raw timestep information by two-layer MLP: FC(512, 128)$\rightarrow$SiLU$\rightarrow$FC(128, 128).

Guidance embedding. The guidance is in the form of one/multi-hot embedding $\mathbb{R}^K$, we feed it into two-layer MLP: FC(K, 256)$\rightarrow$SiLU$\rightarrow$FC(256, 256), then feed those guidance signal into the UNet following in Figure 8.

Cross-attention. In training for non object-centric dataset, we also tokenize the guidance signal to several tokens following Imagen Saharia et al. [38], we concatenate those tokens with image tokens (can be transposed to a token from typical feature map by $\mathbb{R}^{W\times H\times C} \rightarrow \mathbb{R}^{C\times WH}$), the cross-attention [36; 4] is conducted by $\mathtt{CA}(\mathtt{m}, \mathtt{concat}[\mathbf{k}, \mathtt{m}])$. Due to the quadratic complexity of transformer [23; 27], we only apply the cross-attention in lower-resolution feature maps.

F.2 Training Parameter

$3\times32\times32$ model,4GPU,ImageNet32

Base channels: 128
Channel multipliers: 1, 2, 4
Blocks per resolution: 2
Attention resolutions: 4
number of head: 8
Conditioning embedding dimension: 256
Conditioning embedding MLP layers: 2
Diffusion noise schedule: linear
Sampling timesteps: 256

Optimizer: AdamW
Learning rate: $3e-4$
Batch size: 128
EMA: 0.9999
Dropout: 0.0
Training hardware: 4 × A5000(24G)
Training Epochs: 100
Weight decay: 0.01

$3\times64\times64$ model, 4GPU, ImageNet64

Base channels: 128
Channel multipliers: 1, 2, 4
Blocks per resolution: 2
Attention resolutions: 4
number of head: 8
Conditioning embedding dimension: 256
Conditioning embedding MLP layers: 2
Diffusion noise schedule: linear
Sampling timesteps: 256

Optimizer: AdamW
Learning rate: $1e-4$
Batch size: 48
EMA: 0.9999
Dropout: 0.0
Training hardware: 4 × A5000(24G)
Training Epochs: 100
Weight decay: 0.01

$3\times64\times64$ model, 1GPU, Pascal VOC, COCO_20K, COCO-Stuff

Base channels: 128
Channel multipliers: 1, 2, 4
Blocks per resolution: 2
Attention resolutions: 4
Number of head: 8
Conditioning embedding dimension: 256
Conditioning embedding MLP layers: 2
Diffusion noise schedule: linear
Sampling timesteps: 256
Context dim: 32

Optimizer: AdamW
Learning rate: $1e-4$
Batch size: 80
EMA: 0.9999
Dropout: 0.0
Training hardware: 1 × A6000(45G)
Training Epochs: 800/800/400
Weight decay: 0.01
Context token number: 8

F.3 Dataset preparation

The preparation of unbalanced dataset. There are 50,000 images in the validation set of ImageNet with 1,000 classes (50 instances for each). We index the class from 0 to 999, for each class c_i, the instance of the class c_i is $\lfloor i \times 50/1000 \rfloor = \lfloor i/200 \rfloor$.

Pascal VOC. We use the standard split from [40]. It has 12,031 training images. As there is no validation set for Pascal VOC dataset, therefore, we only evaluate FID on train set. We sample 10,000 images and use 10,000 random-croped 64-sized train images as reference set for FID evaluation.

COCO_20K. We follow the split from [40; 44; 25]. COCO_20k is a subset of the COCO2014 trainval dataset, consisting of 19,817 randomly chosen images, used in unsupervised object discovery [40; 44]. We sample 10,000 images and use 10,000 random-croped 64-sized train images as reference set for FID evaluation.

COCO-Stuff. It has a train set of 49,629 images, validation set of 2,175 images, where the original classes are merged into 27 (15 stuff and 12 things) high-level categories. We use the dataset split following [15; 21; 10; 46], We sample 10,000 images and use 10,000 train/validation images as reference set for FID evaluation.

F.4 LOST, STEGO algorithms

LOST algorithm details. We conduct padding to make the original image can be patchified to be fed into the `ViT` architecture [13], and feed the original padded image into the `LOST` architecture using official source code [2]. `LOST` can also be utilized in a two-stage approach to provide multi-object, due to its complexity, we opt for only single-object discovery in this paper.

STEGO algorithm details. We follow the official source code [3], and apply padding to make the original image can be fed into the `ViT` architecture to extract the self-segmented guidance signal.

For COCO-Stuff dataset, we directly use the official pretrained weight. For Pascal VOC, we train `STEGO` ourselves using the official hyperparameters.

In `STEGO`'s pre-processing for the k-NN, the number of neighbors for k-NN is 7. The segmentation head of `STEGO` is composed of a two-layer `MLP` (with `ReLU` activation) and outputs a 70-dimension feature. The learning rate is $5e-4$, batch size is 64.

G More qualitative results

[2] https://github.com/valeoai/LOST

[3] https://github.com/mhamilton723/STEGO

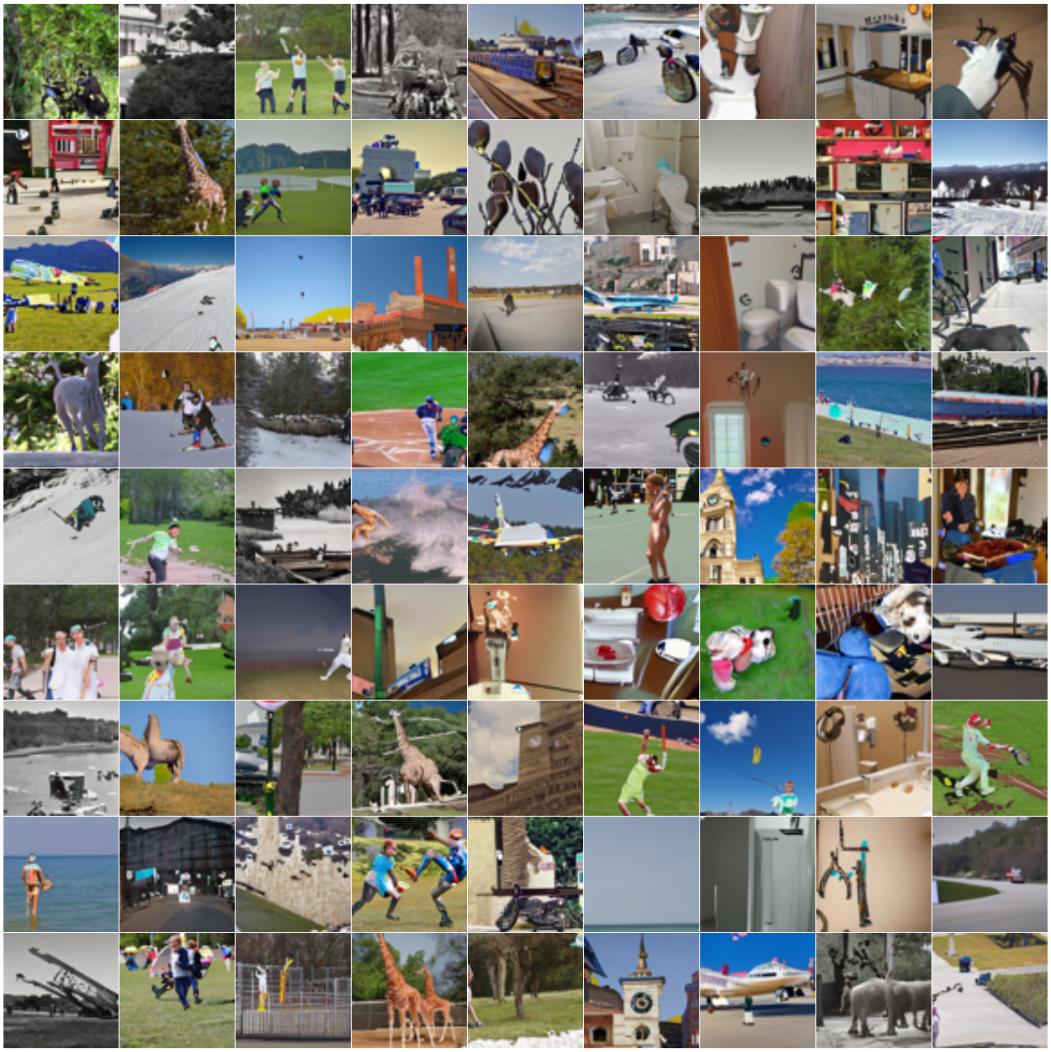


Figure 9: **Self-segmented guidance samples from COCO-Stuff.** Best viewed in color.

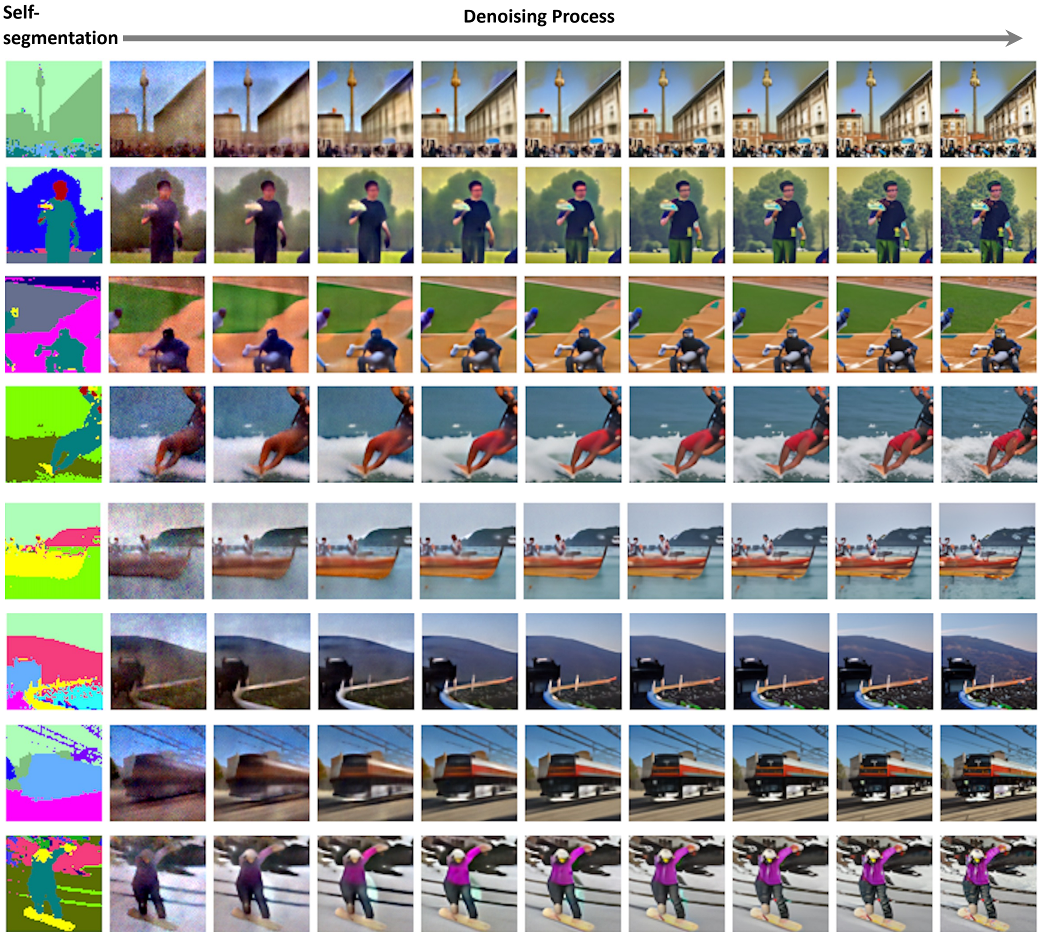


Figure 10: **Denoising process of self-segmented guidance samples (uncurated) from COCO-Stuff.** The first column is the self-segmented guidance mask from STEGO [15], The remaining columns are from the most noisy period to less noisy period. Best viewed in color.

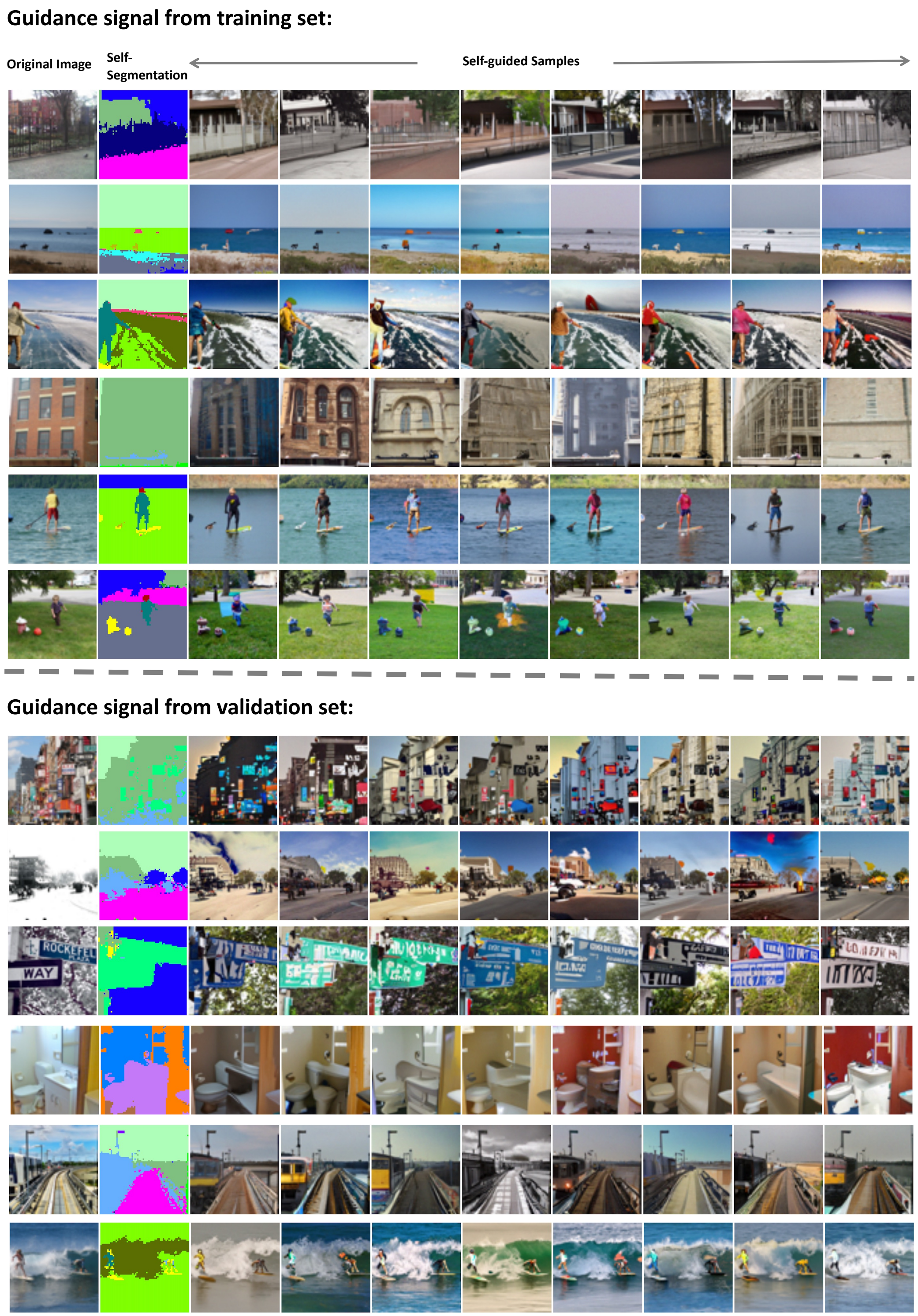

Guidance signal from training set:
Original Image
Self-Segmentation
Self-guided Samples
Guidance signal from validation set:
ROCKEFEL
WAY

Figure 11: **Self-segmented guidance samples (uncurated) from COCO-Stuff.** The first column is the real image where we attain the conditional mask. The second column is the self-segmented mask we obtain from `STEGO` [15], The remaining columns are the random samples conditioning on the same self-segmented mask. Best viewed in color.

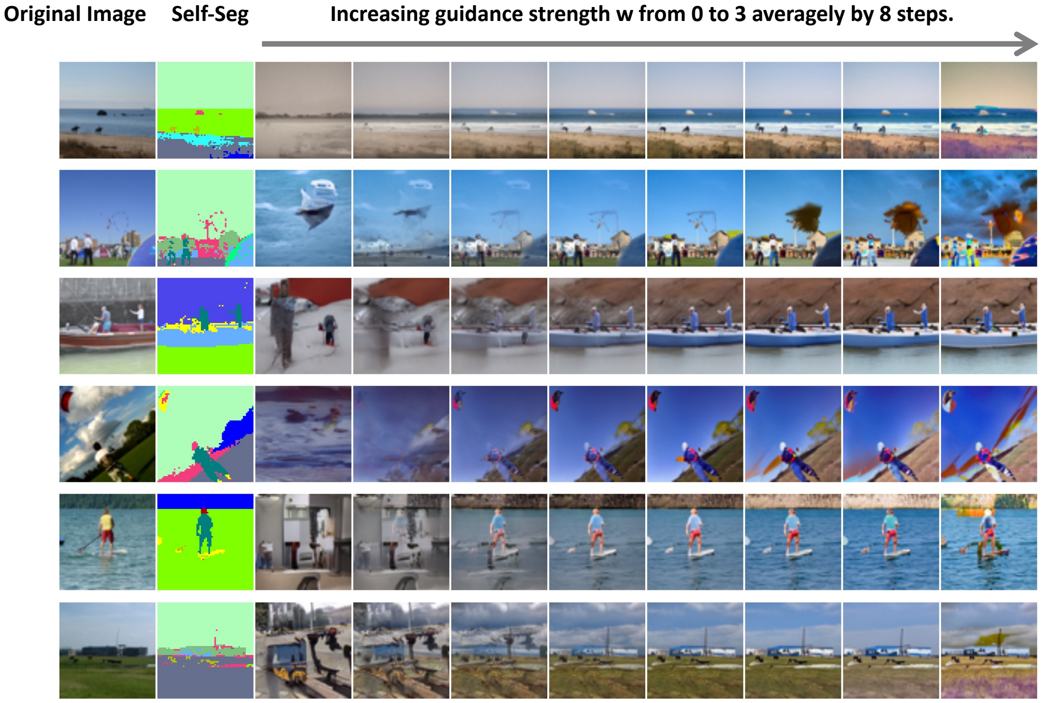

Original Image
Self-Seg
Increasing guidance strength w from 0 to 3 averagely by 8 steps.

Figure 12: **Self-segmented guidance samples from Pascal VOC.** The first column is the real image where we attain the conditional mask. The second column is the self-segmented mask we obtain from `STEGO` [15]. The remaining columns are the visualization when we averagely increase guidance strength w from 0 to 3 by 8 steps. Best viewed in color.

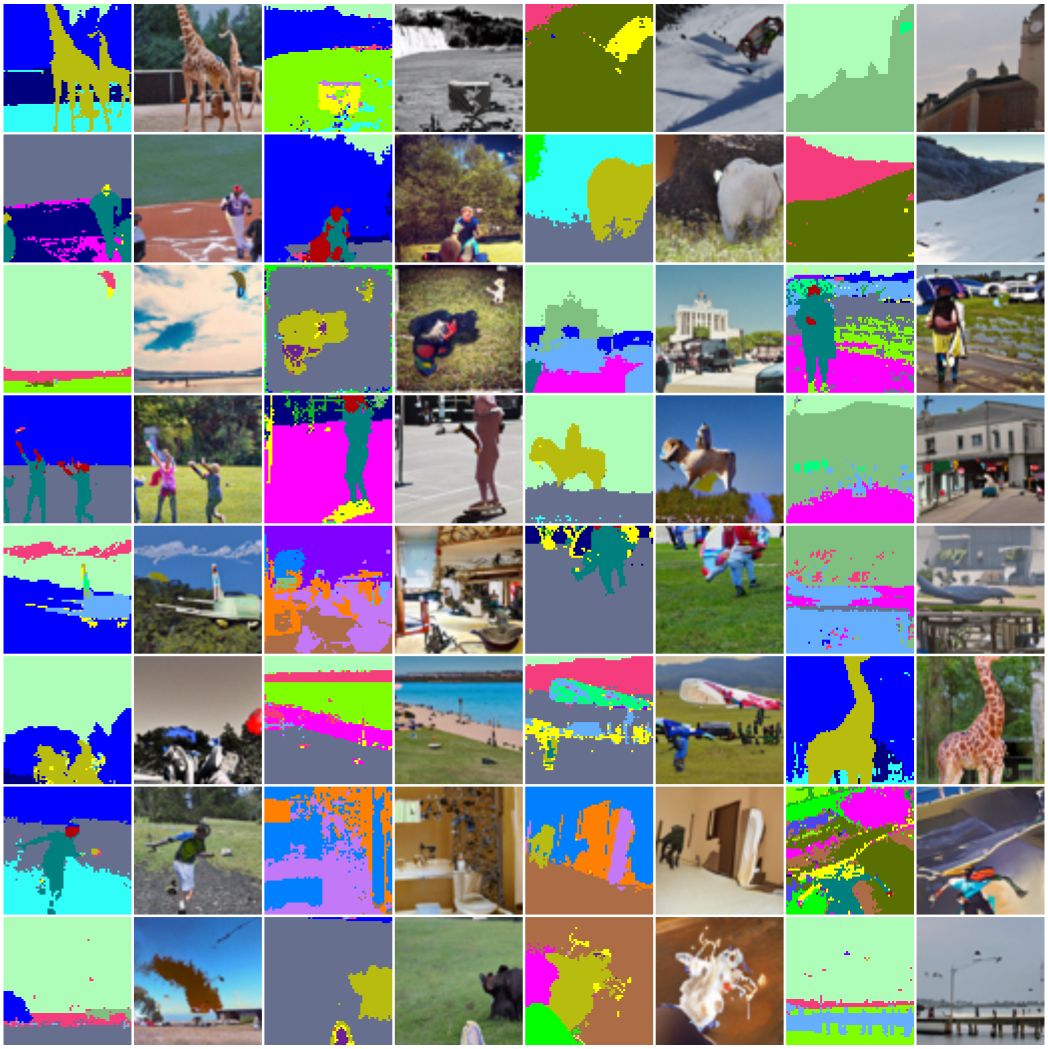


Figure 13: **Self-segmented guidance samples (uncurated)** from COCO-Stuff companies with segmentation mask from STEGO [15]. The color map is shared among the overall dataset. Best viewed in color.

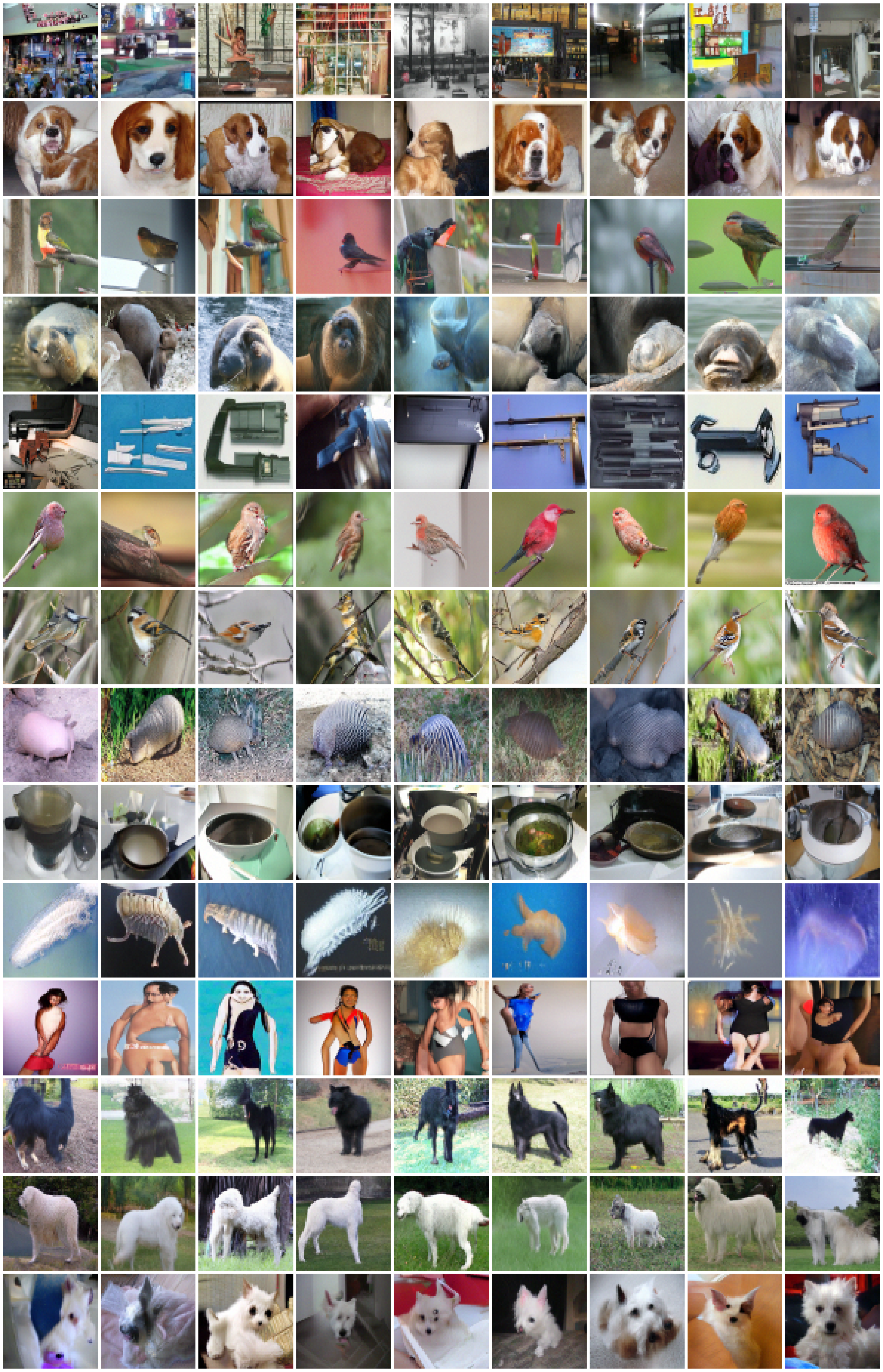


Figure 14: **Self-labeled guidance samples (uncurated) conditioning on the same guidance from ImageNet64.** Best viewed in color.

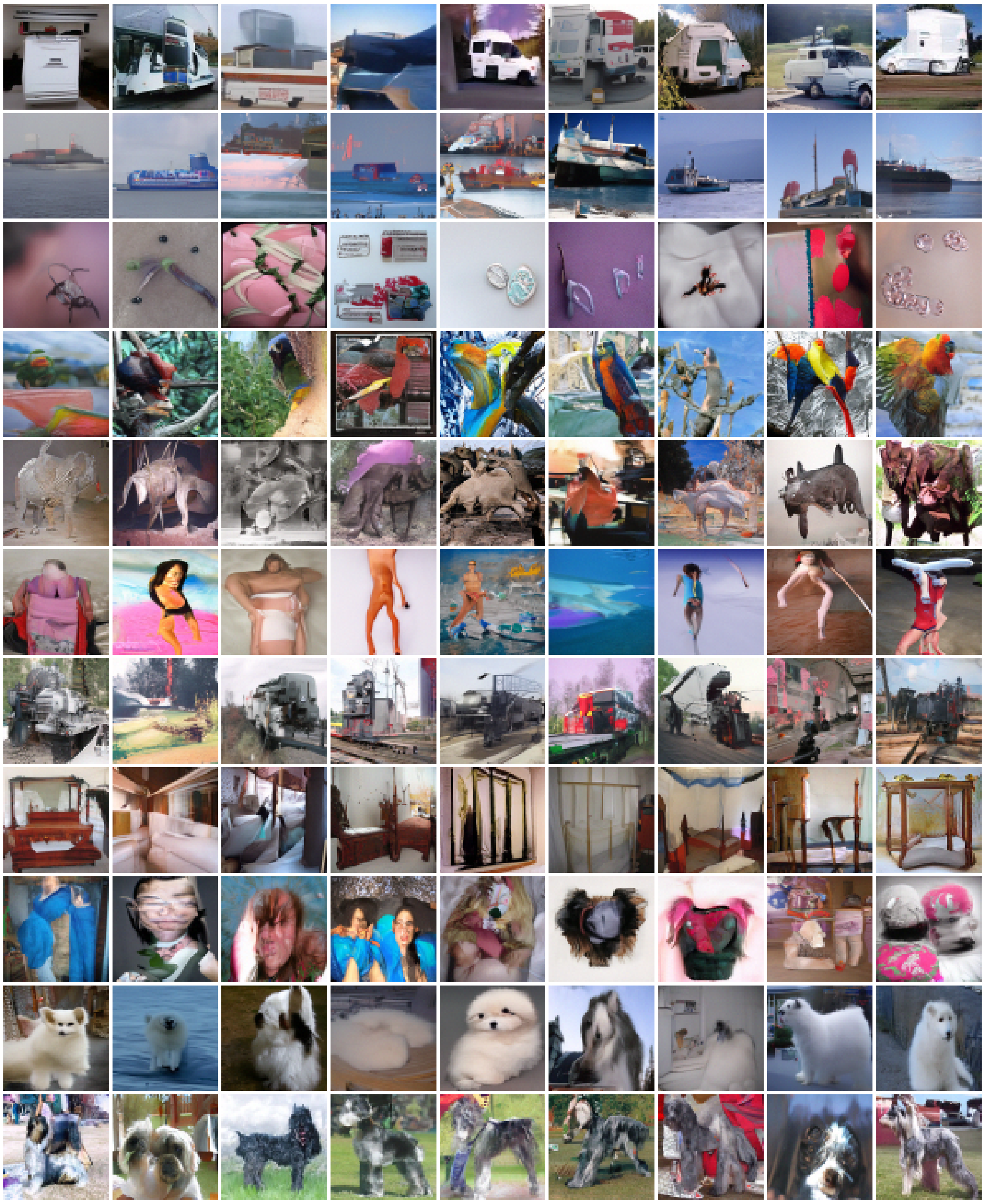


Figure 15: **Ground-truth labels guidance samples (uncurated) conditioning on the same guidance signal from ImageNet64.** Best viewed in color.

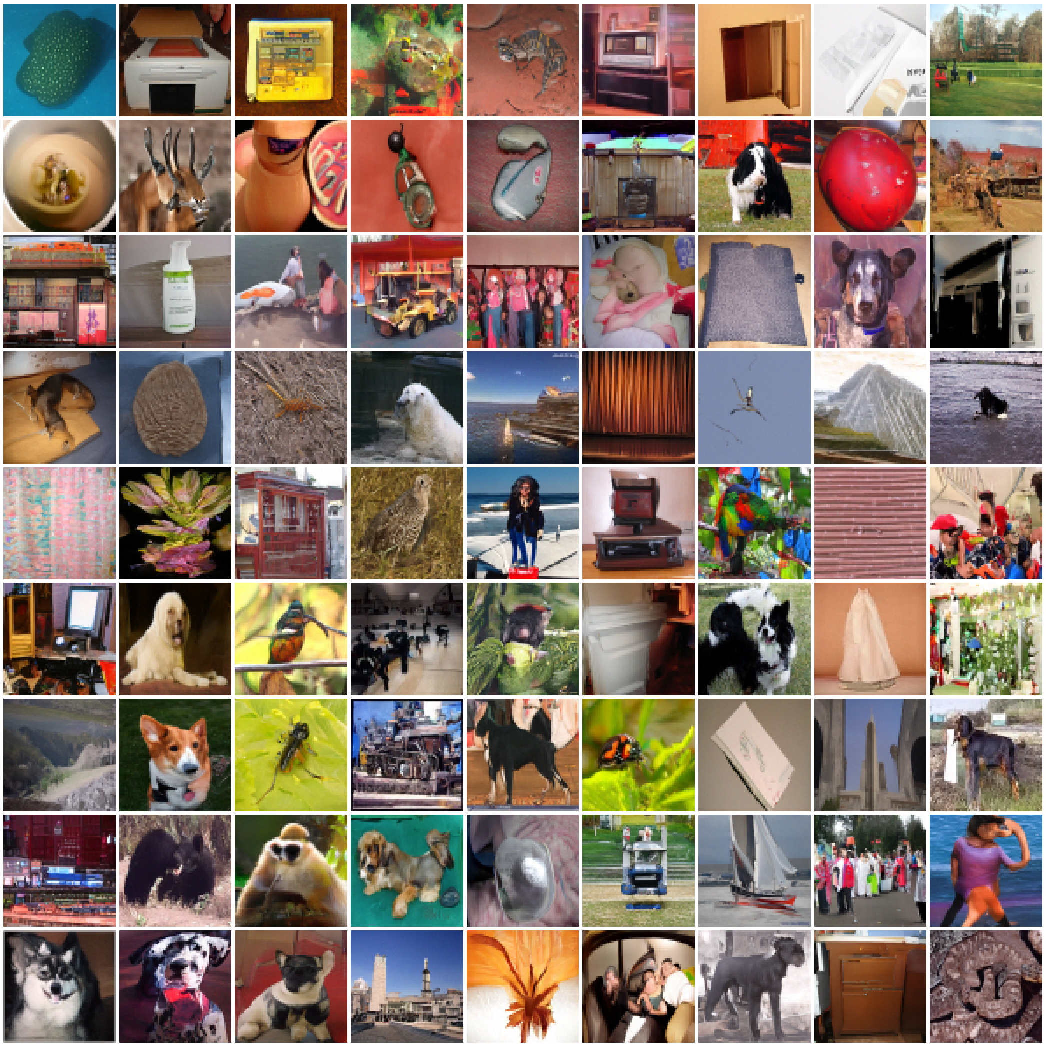


Figure 16: **Self-labeled guidance samples (uncurated) from ImageNet64.** Best viewed in color.

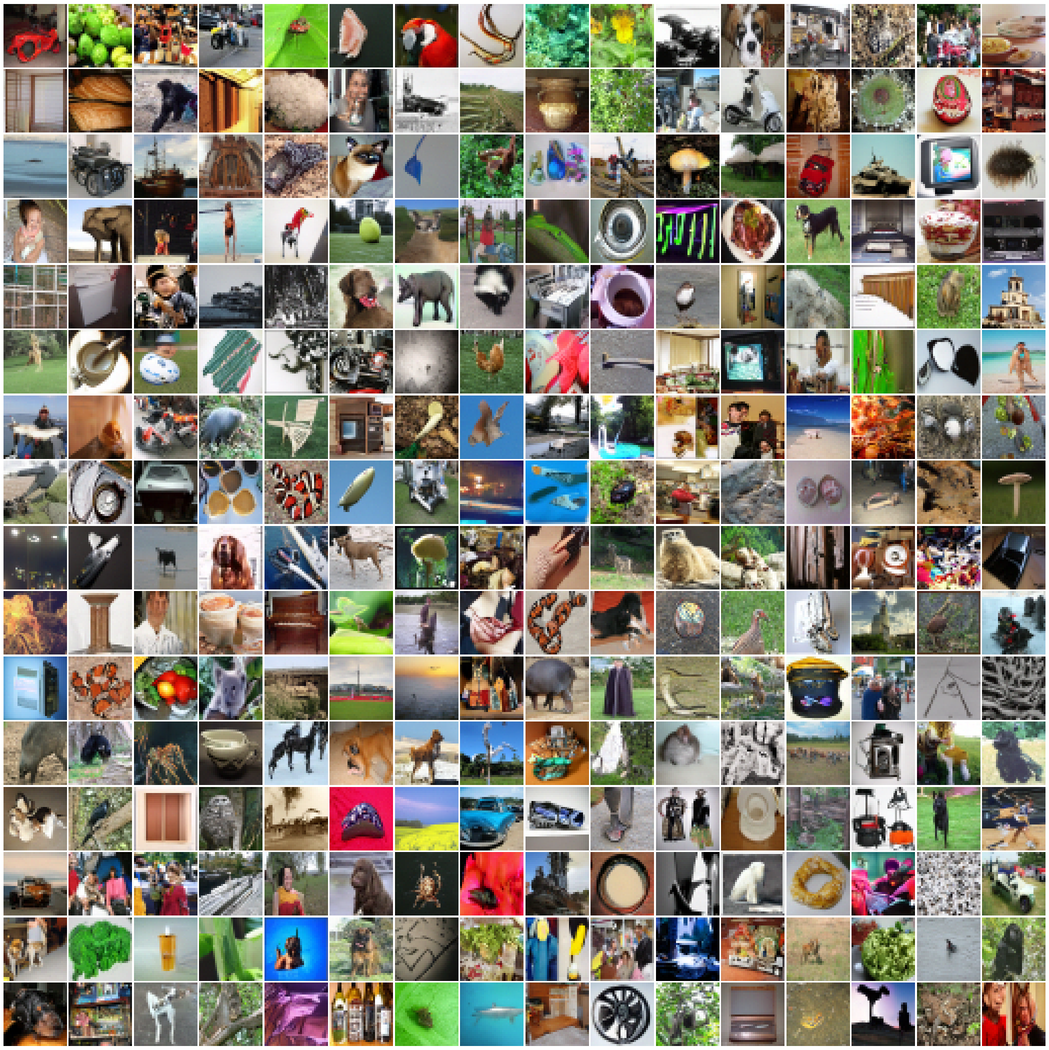
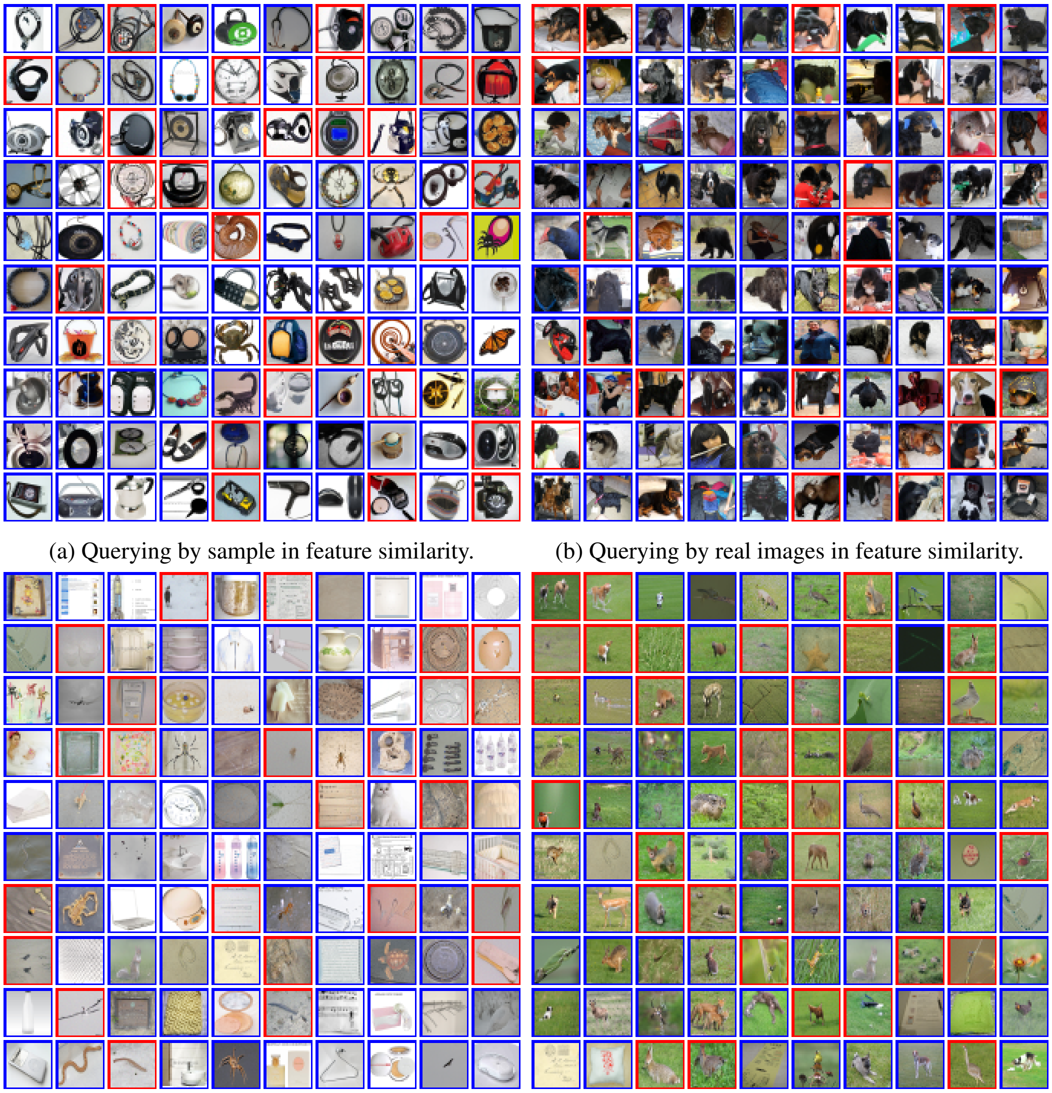


Figure 17: **Self-labeled guidance samples (uncurated) from ImageNet32.** Best viewed in color.



(a) Querying by sample in feature similarity.

(b) Querying by real images in feature similarity.

(c) Querying by sample in pixel similarity.

(d) Querying by real images in pixel similarity.

Figure 18: k**-NN query result visualization.** Blue means samples, red means real images. Images are ordered from left to right, top to down, by SimCLR [9] feature similarity or pixel similarity. Sampled images are sampled by DDIM [42] with 250 steps. Guidance strength w is 2. Firstly, we construct a gallery which is composed of equivalent number of sampled and real images, then we ablate two experiments by querying using sampled image or real images in feature space and image space. **Conclusion:**We can easily see, regardless of the feature space or image space, the k-NN query results are always highly semantic similar, and they show the diffusion model is not only to memorize the training data/real images, but also can generalize well to synthesize novel images.

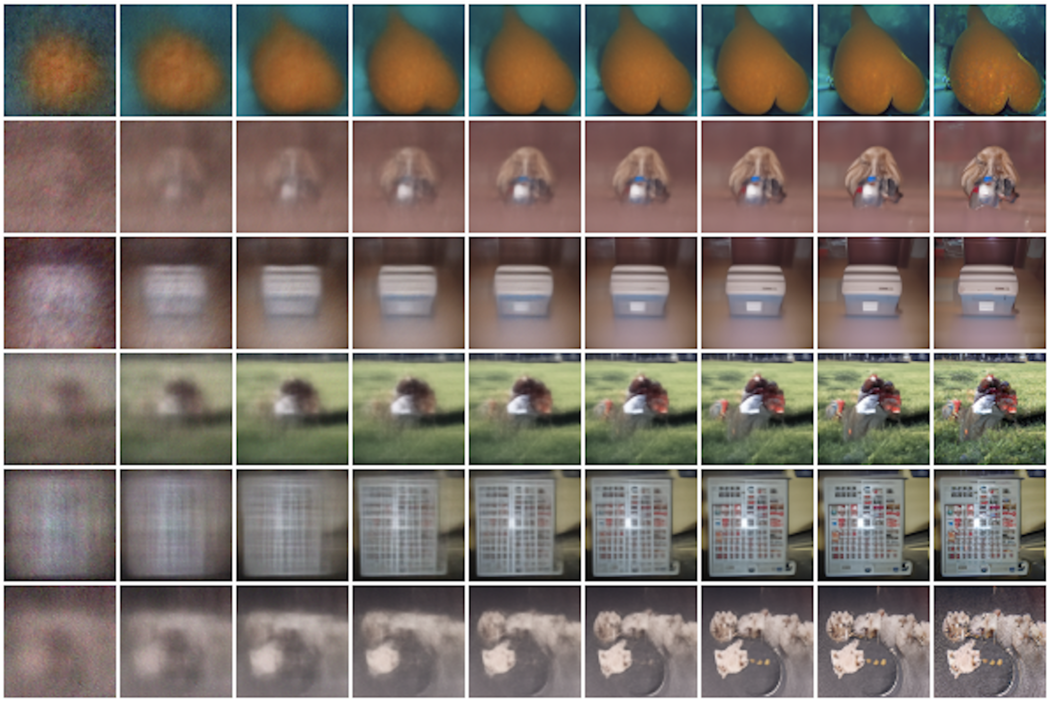


Figure 19: **Denoising process for ImageNet64.**

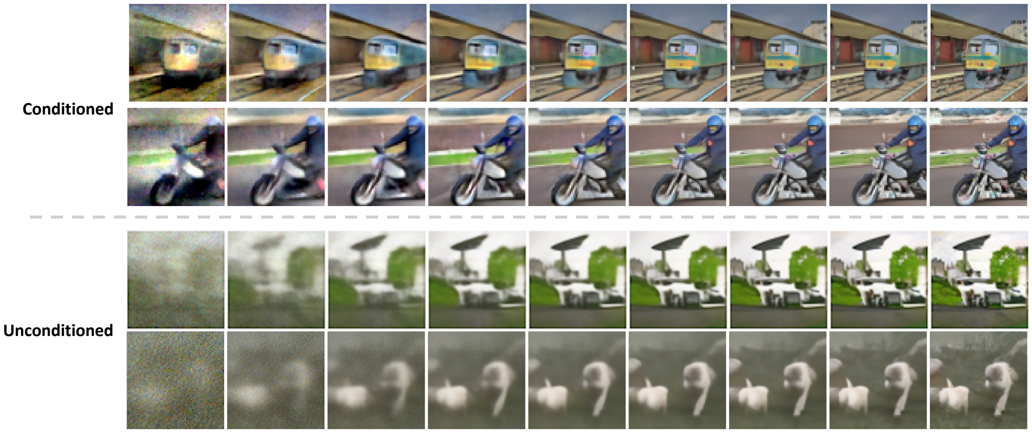


Figure 20: **Denoising process for Pascal VOC.** The first two rows are sampled from guidance strength $w = 2$ using our self-segmented guidance, the last two rows are sampled from guidance strength $w = 0$. By conditioning on our self-segmented guidance, the denoising process becomes easier and faster, this efficient denoising aligns with the observation from [31].

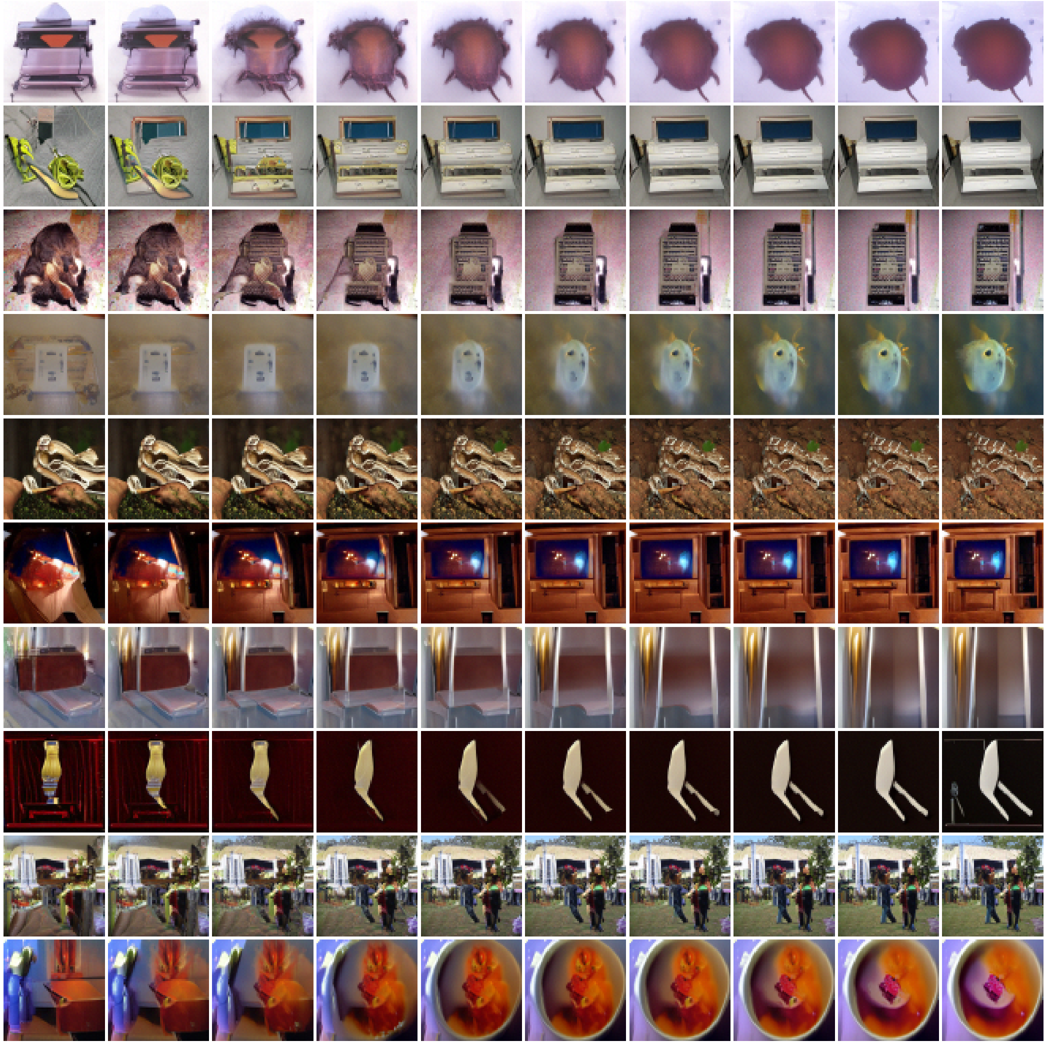


Figure 21: **Sample visualization with guidance strength w from 0 to 3 in ImageNet64.** Best viewed in color.

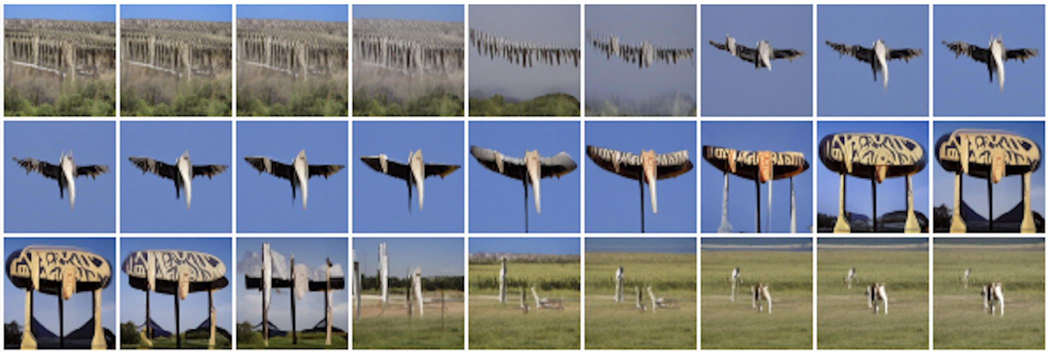
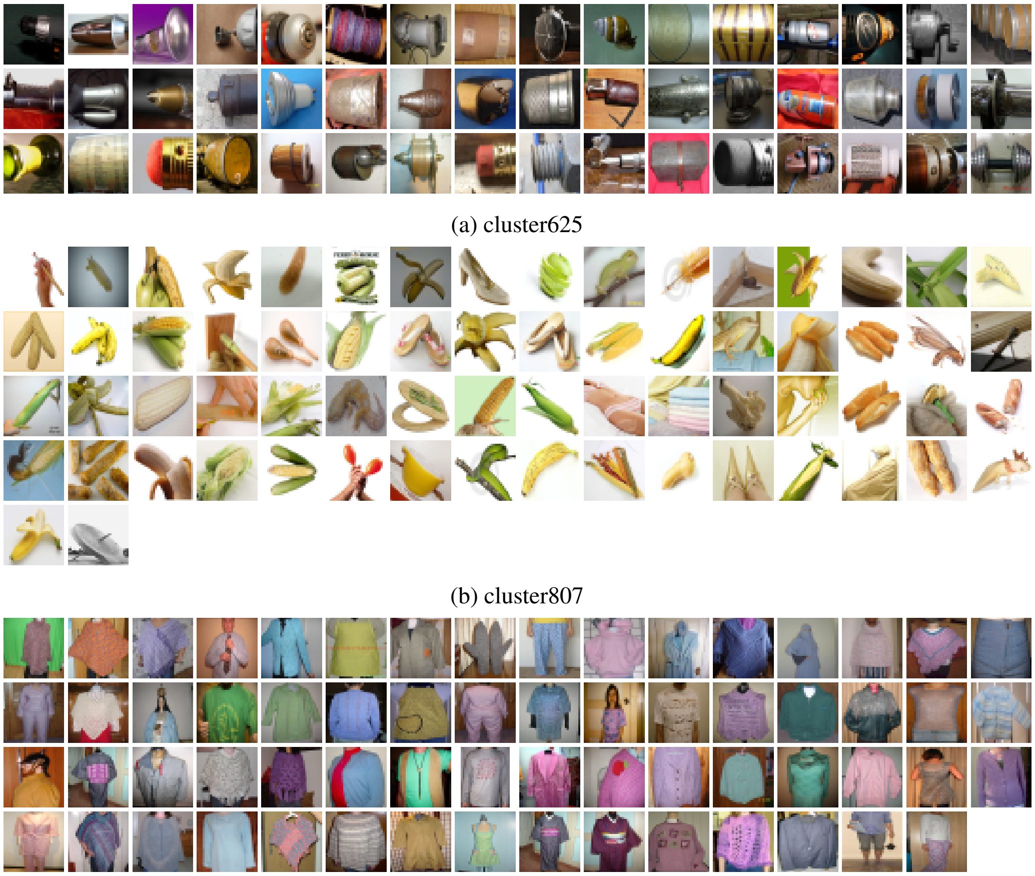


Figure 22: **Sphere interpolation between two random self-labeled guidance signals on ImageNet64.** The sphere interpolation follows the DDIM [42]. Best viewed in color.

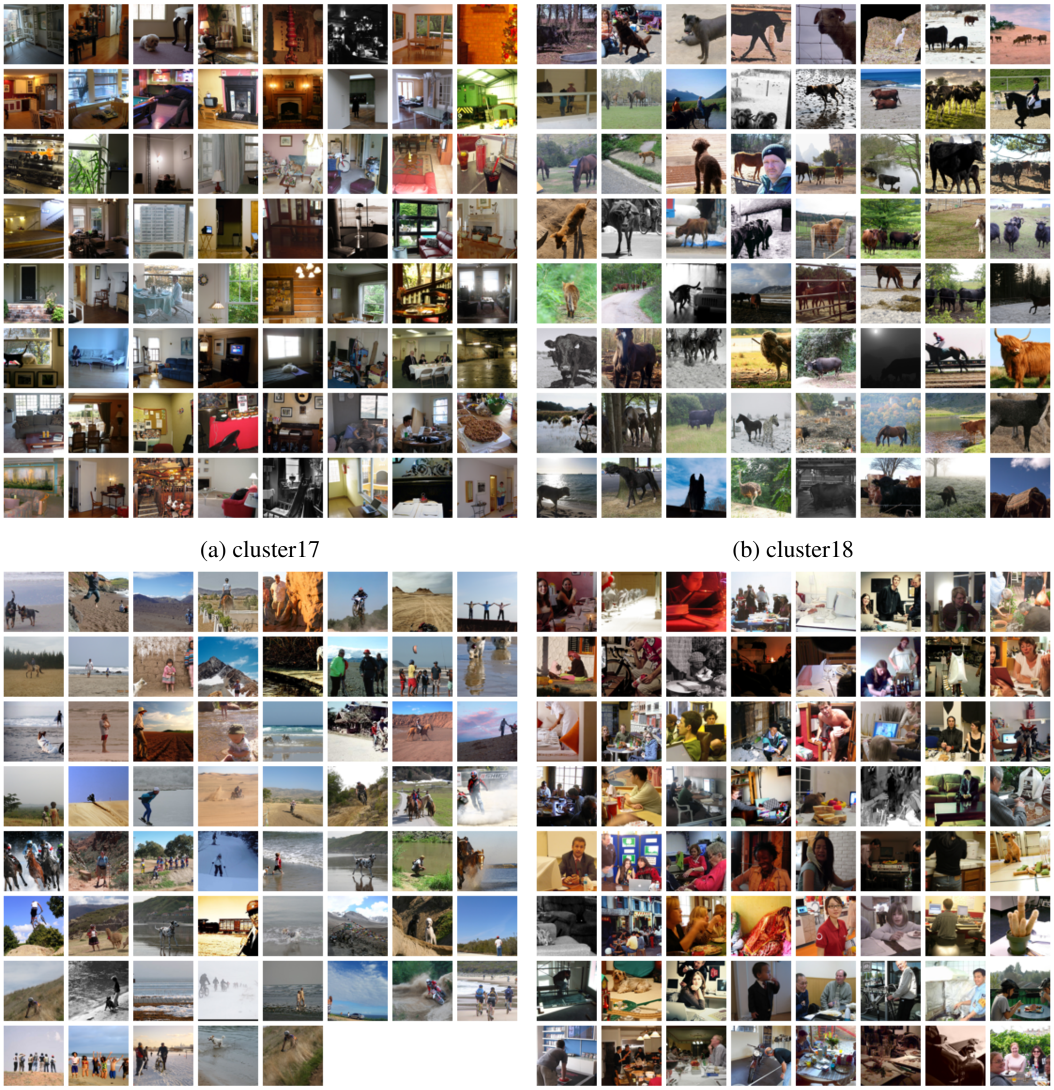


(a) cluster625

(b) cluster807

(c) cluster890

Figure 23: **Cluster visualization of real images in ImageNet32 after k-means.**



(a) cluster17

(b) cluster18

(c) cluster45

(d) cluster50

Figure 24: **Cluster visualization of real images in Pascal VOC after k-means.** Best viewed by zooming in.

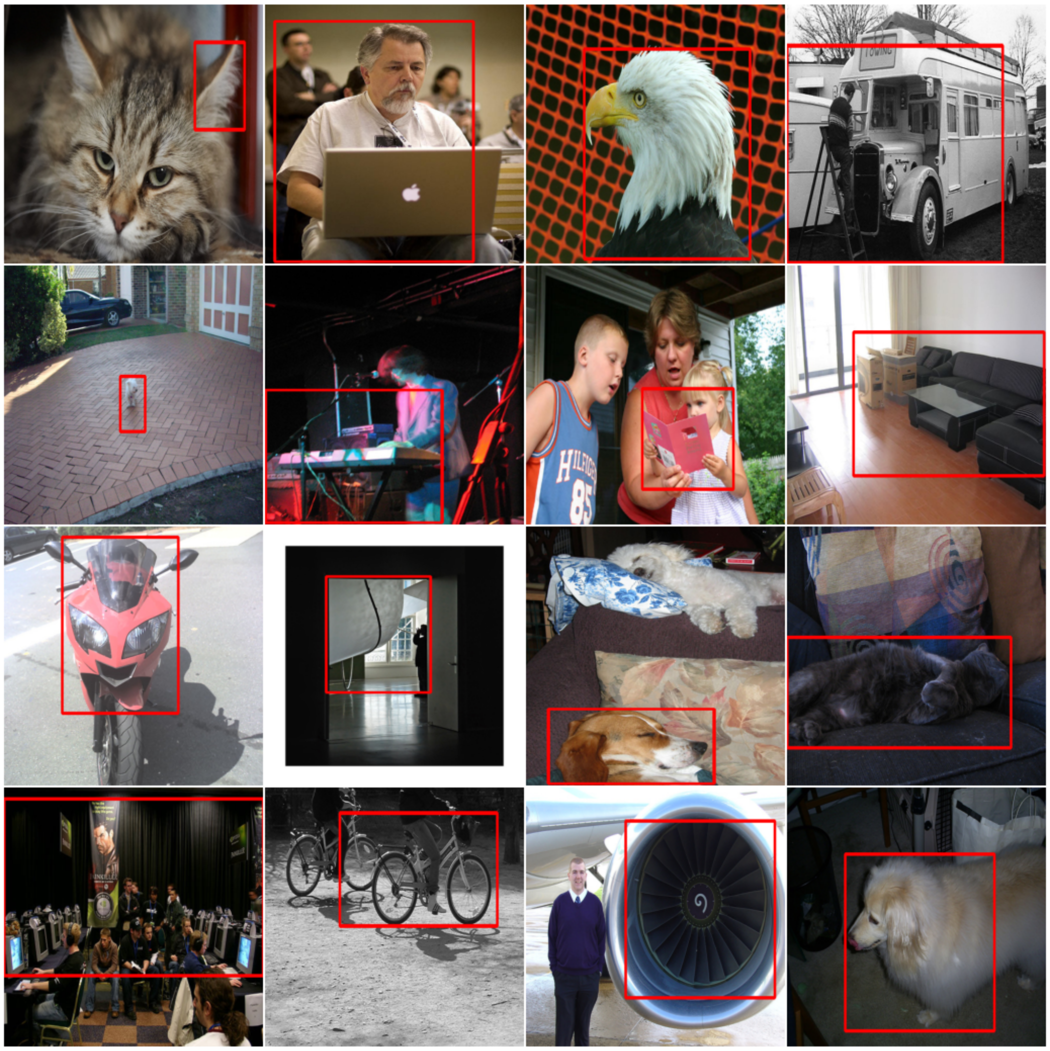


Figure 25: **Bounding box result from LOST on Pascal VOC**. As `LOST` [40] is an unsupervised-learning method, some flaws in the generated box are expected. Images are resized squarely for better visualization.

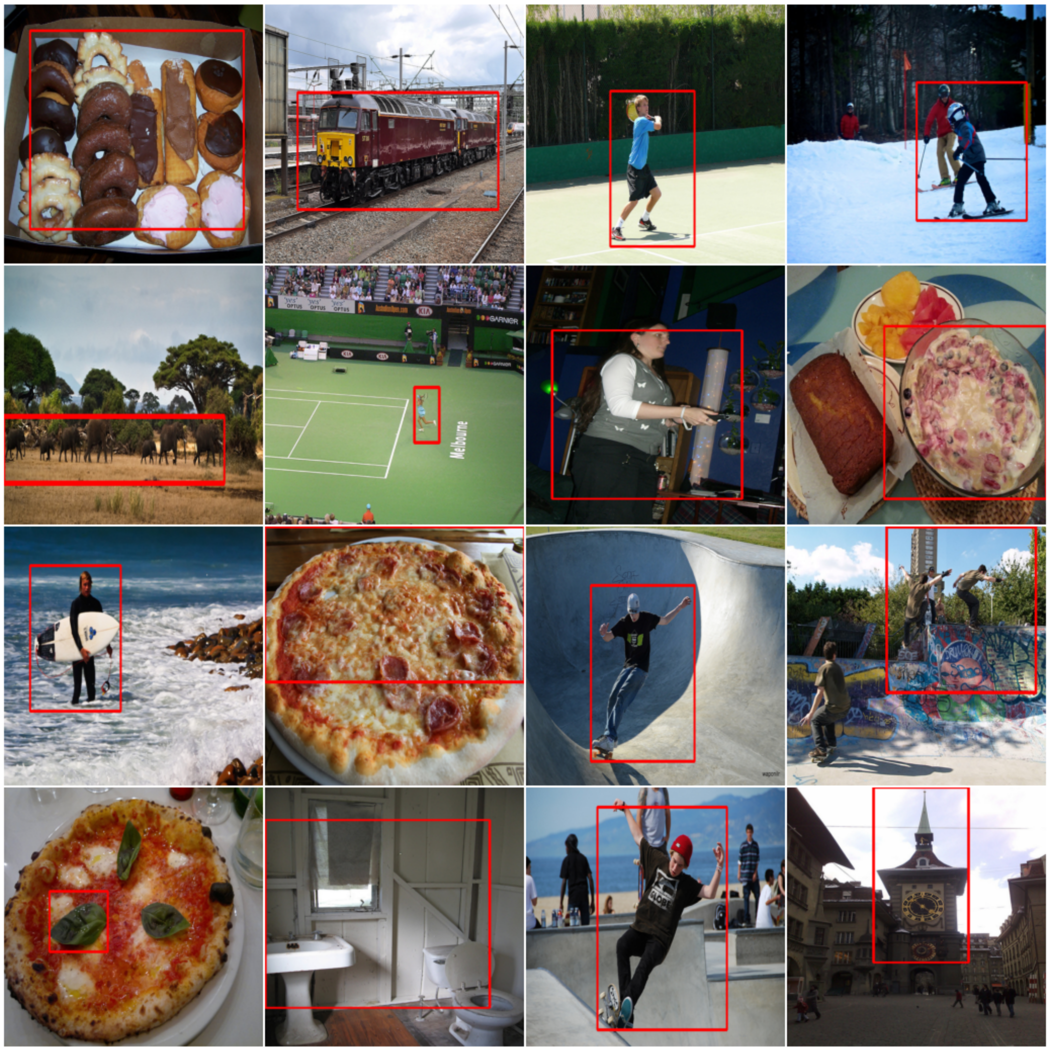


Figure 26: **Bounding box result from LOST [40] on COCO_20K.** Images are resized squarely for better visualization.

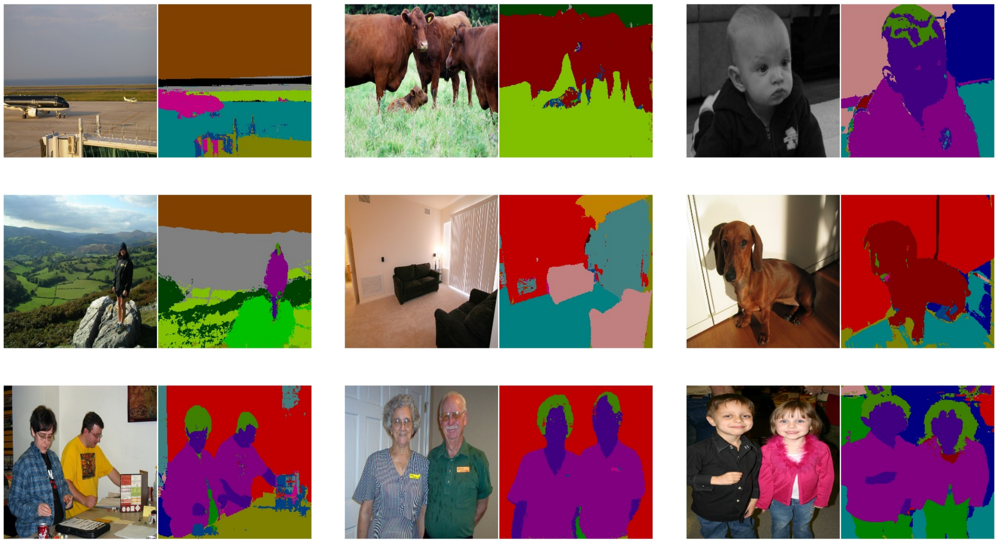


Figure 27: **Segmentation mask result from STEGO on Pascal VOC dataset**. Cluster number k is 21. Images are resized squarely for better visualization. The color map is shared among the overall dataset. Best viewed in color.