Self-Guided Diffusion Model

Anonymous Author(s)
Affiliation
Address
email

Abstract

Diffusion models have demonstrated remarkable progress in image generation quality, especially when guidance is used to control the generative process. However, such 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 aim to 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 flexible guidance signals at various image granularities: from the level of holistic images to object boxes and even segmentation masks.

1 Introduction

The image fidelity of diffusion models is spectacularly enhanced by conditioning on class labels (19). 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 (19). Classifier-free guidance by (32) replaces the dedicated classifier with an 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 (66; 57), image layout (64), visual neighbors (3), and image features (7; 24). 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 (52). In this paper, we propose to remove the necessity of any ground-truth annotation for guidance diffusion models.

We are inspired by progress in self-supervised learning (15; 12), which encodes data, and especially images, into semantically meaningful latent vectors without using any label information. It usually does so by solving a pretext task (78; 25; 2; 29) on image-level to remove the necessity of ground-truth labels in discriminative tasks. This annotation-free paradigm enables the representation learning to scale to larger and more diverse (image) datasets (23). Recently, the holistic image-level self-supervision has been extended to more expressive dense representations, including bounding boxes, e.g., (68; 48) and pixel-precise segmentation masks, e.g., (27; 81). As some self-supervised learning methods even outperform supervised alternatives (29; 12), we hypothesize self-supervision may also provide a flexible and competitive, possibly even stronger guidance signal than ground-truth labeled guidance for diffusion models.

2 Self-Guided Diffusion

The equations describing conditional diffusion by classifier-free guidance (see Appendix A for the background) 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):

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

Under review at the NeurIPS 2022 Workshop on Score-Based Methods. Do not distribute.

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 (65) due to its superior image generation performance, following (33; 71; 58; 66).

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. See Appendix B for a description of the box-level and pixel-level guidance approaches.

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 (63), 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.

3 Experiments

In this section, we aim to answer the overarching question: Can we substitute ground-truth annotations with self-annotations? First, we consider the image-label setting, in which we examine what kind of self-labeling is required to improve image fidelity. Next, we look at image-bounding box pairs. Finally, we examine whether it is possible to gain fine-grained control without any image-segmentation pairs. As page limits, we only present self-labeled guidance here. For self-boxed/segmented guidance, please check Appendix C.

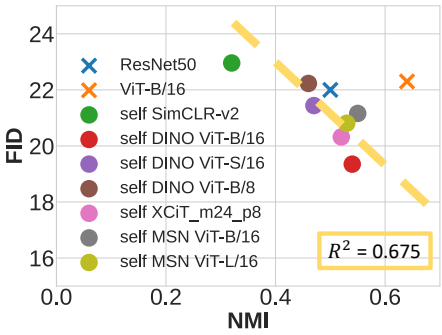
3.1 Self-labeled Guidance

We use ImageNet32/64 (18) to validate the efficacy of self-labeled guidance. For evaluation of sampling quality, we adopt Fréchet Inception Distance (FID) (31) and the Inception Score (IS) (67), following the common practice (19; 36; 9).

Choice of feature extraction function *g*. We first ablate the influence of the feature extraction function g used before clustering. We consider two supervised feature backbones: `ResNet50` (28) and `ViT-B/16` (21), and four self-supervised backbones: `SimCLR` (16), `MAE` (30), `MSN` (4) and `DINO` (12). To assure a fair comparison we use 10k clusters for all architectures. From the results in Figure 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

	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

(a)



(b)

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

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 suggests the label assignment for the guidance is not unique. Self-supervised guidance leads to more unbiased hierarchies in comparison to supervised guidance. To further 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 1 we plot the connection between NMI and FID. For the label-supervised functions the NMI is unrelated to FID, but for the self-supervised functions, NMI and FID are negatively correlated, which suggests NMI allows to select the feature extraction function g. 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 and IS results for DDPM and classifier-free guidance (for 20 epochs). From the result in Figure 2 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 even 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 2. The performance 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.

Comparisons on ImageNet32/64. We compare our self-labeled guidance with ground-truth labels guidance, which utilizes the technique of classifier-free guidance (32). 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 1. Similar to Dhariwal & Nichol (19), 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 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 condition signal due to better alignment with the visual similarity of the images.

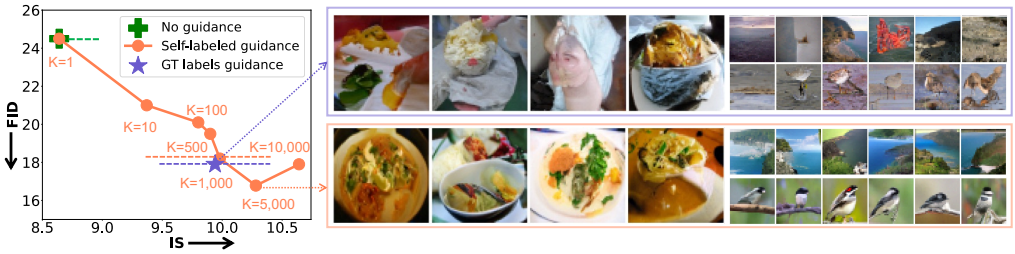


Figure 2: **Effect of number of clusters.** Self-labeled guidance outperforms DDPM without any guidance beyond a single cluster, is competitive with classifier-free guidance beyond 1,000 clusters and is even able to outperform guidance by ground-truth(GT) labels for 5,000 clusters. We visualize generated samples from ImageNet64 (middle) and ImageNet32 (right) for ground-truth labels guidance (top) and self-labeled guidance (bottom). More qualitative results in Appendix Figure 12.

		ImageNet32		ImageNet64	
Guidance	**Annotation-free?**	FID↓	IS ↑	FID↓	IS↑
Ground-truth labels (32)	✗	10.2	19.0	16.8	18.6
None (33)	✓	14.3	10.8	36.1	10.4
Self-labeled (***this paper***)	✓	**7.3**	**20.3**	**12.1**	**23.1**

Table 1: **Comparisons on ImageNet32/64.** Self-labelled guidance surpasses the no-guidance baseline by a large margin on both datasets, further outperforming the model trained using ground-truth class labels.

Varying guidance strength w. Next we consider the influence of the guidance strength w on our sampling results. Here we always run 100 epochs on ImageNet32. 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. For the standard ImageNet32 validation setting in Figure 3a, our method achieves a 17.8% improvement among the lower bounds 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 3b, be it that the overall FID results are slightly higher for both methods and the maximal performance improvement increases to 18.7%. We conjecture this is due to the unbalanced nature of the k-means algorithm (42), and clustering based on the statistics of the overall dataset can potentially lead to more robust performance in an unbalanced setting.

More results on self-boxed/segmented guidance. Please check the quantitative and qualitative results in C.

4 Conclusion

We have explored the potential of self-supervised representations to replace human annotations in generative models. We have shown significant improvements in image quality over models trained with ground-truth annotations, by using self-annotated condition signals. Furthermore, both our self-boxed and self-segmented approaches demonstrate that it is possible to acquire fine-grained control over the image content, without any ground-truth bounding boxes or segmentation masks. Due to limited computational resources, we have restricted our experiments to images of a maximum size of 64×64. For future work, it will be significant to verify our findings on larger images. Ultimately, our goal is to enable the benefits of conditioning and guidance for larger and more diverse datasets, and for new domains, wherein this work is a promising first step.

References

[1] Mohammadreza Armandpour, Ali Sadeghian, Chunyuan Li, and Mingyuan Zhou. Partition-guided gans. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[2] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. *International Conference on Learning Representations*, 2020.

[3] 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.

[4] 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.

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

[6] Jens Behrmann, Paul Vicol, Kuan-Chieh Wang, Roger Grosse, and Jörn-Henrik Jacobsen. Understanding and mitigating exploding inverses in invertible neural networks. In *International Conference on Artificial Intelligence and Statistics*, 2021.

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

[8] Florian Bordes, Randall Balestriero, and Pascal Vincent. High fidelity visualization of what your self-supervised representation knows about. *arXiv preprint arXiv:2112.09164*, 2021.

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

[10] 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.

[11] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 2020.

[12] 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.

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

[14] Ting Chen, Xiaohua Zhai, Marvin Ritter, Mario Lucic, and Neil Houlsby. Self-supervised gans via auxiliary rotation loss. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[15] 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.

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

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

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

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

[20] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

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

[22] Patrick Esser, Robin Rombach, and Bjorn Ommer. A disentangling invertible interpretation network for explaining latent representations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9223–9232, 2020.

[23] Shang-Hua Gao, Zhong-Yu Li, Ming-Hsuan Yang, Ming-Ming Cheng, Junwei Han, and Philip Torr. Large-scale unsupervised semantic segmentation. *arXiv preprint arXiv:2106.03149*, 2021.

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

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

[26] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

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

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

[29] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

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

[31] 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. *Advances in Neural Information Processing Systems*, 2017.

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

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

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

[35] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

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

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

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

[39] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in Neural Information Processing Systems*, 2018.

[40] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. *Advances in Neural Information Processing Systems*, 2016.

[41] Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):3964–3979, 2020.

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

[43] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014.

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

[45] 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. *Advances in Neural Information Processing Systems*, 2021.

[46] Mario Lučić, Michael Tschannen, Marvin Ritter, Xiaohua Zhai, Olivier Bachem, and Sylvain Gelly. High-fidelity image generation with fewer labels. In *International Conference on Machine Learning*, 2019.

[47] Puneet Mangla, Nupur Kumari, Mayank Singh, Balaji Krishnamurthy, and Vineeth N Balasubramanian. Data instance prior (disp) in generative adversarial networks. In *IEEE Winter Conf. on Applications of Computer Vision*, 2022.

[48] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8364–8375, 2022.

[49] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[50] Stanislav Morozov, Andrey Voynov, and Artem Babenko. On self-supervised image representations for gan evaluation. In *International Conference on Learning Representations*, 2020.

[51] 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.

[52] Andreas Panteli, Jonas Teuwen, Hugo Horlings, and Efstratios Gavves. Sparse-shot learning with exclusive cross-entropy for extremely many localisations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2813–2823, 2021.

[53] George Papamakarios, Eric T Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *J. Mach. Learn. Res.*, 22(57):1–64, 2021.

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

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

[56] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

[57] 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.

[58] 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.

[59] Ali Razavi, Aäron van den Oord, Ben Poole, and Oriol Vinyals. Preventing posterior collapse with delta-vaes. *arXiv preprint arXiv:1901.03416*, 2019.

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

[61] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*. PMLR, 2015.

[62] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*. PMLR, 2014.

[63] Jason Tyler Rolfe. Discrete variational autoencoders. *arXiv preprint arXiv:1609.02200*, 2016.

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

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

[66] 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. *arXiv preprint arXiv:2205.11487*, 2022.

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

[68] 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. *British Machine Vision Conference*, 2021.

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

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

[71] 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. URL `https://openreview.net/forum?id=PxTIG12RRHS`.

[72] Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. *Advances in Neural Information Processing Systems*, 2017.

[73] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[74] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

[75] Huy V Vo, Patrick Pérez, and Jean Ponce. Toward unsupervised, multi-object discovery in large-scale image collections. In *European Conference on Computer Vision*, pp. 779–795. Springer, 2020.

[76] Van Huy Vo, Elena Sizikova, Cordelia Schmid, Patrick Pérez, and Jean Ponce. Large-scale unsupervised object discovery. *Advances in Neural Information Processing Systems*, 34: 16764–16778, 2021.

[77] Ning Yu, Ke Li, Peng Zhou, Jitendra Malik, Larry Davis, and Mario Fritz. Inclusive gan: Improving data and minority coverage in generative models. In *European Conference on Computer Vision*, 2020.

[78] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[79] Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. Dense siamese network. *arXiv preprint arXiv:2203.11075*, 2022.

[80] Yufan Zhou, Ruiyi Zhang, Changyou Chen, Chunyuan Li, Chris Tensmeyer, Tong Yu, Jiuxiang Gu, Jinhui Xu, and Tong Sun. Lafite: Towards language-free training for text-to-image generation. *arXiv preprint arXiv:2111.13792*, 2021.

[81] Adrian Ziegler and Yuki M Asano. Self-supervised learning of object parts for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

A Background

Diffusion models. Diffusion models (69; 33) 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 $(\mathbf{x}_t - (1-\overline{\alpha}_t)\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t))/\sqrt{\overline{\alpha}_t} \approx \mathbf{x}_0$. (33) 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 (33) requires many neural function evaluations to get good quality samples. Instead, the faster Denoising Diffusion Implicit Models (DDIM) sampler (70) 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 (19) 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, (32) 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 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 (32) 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 Method: Self-boxed/segmented guidance

Self-boxed guidance. Box is an important intermediary in representation learning (60; 10), we aim to attain this signal by self-supervision to achieve self-boxed guidance. For self-boxed guidance, we represent each bounding box by 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: $\mathbf{x}_t := \texttt{concat}[\mathbf{x}_t, \mathbf{k}_s]$ Usually in self-supervised learning, the derived bounding box is class-agnostic (75; 76). 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 $t := \texttt{concat}[t, \mathbf{k}]$. To incorporate those guidance, in the end we reformulate the noise prediction function $\boldsymbol{\epsilon}_\theta$ as:

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

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 the object will be generated. Sometimes, we may need an even finer granularity, e.g., pixels. which we detail next.

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 via function mapping $\mathbf{k}_s{=}f_\psi(\cdot;\ \mathcal{D}) : \mathbb{R}^{W\times H\times C} \rightarrow \mathbb{R}^{W\times H\times K}$, 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 those 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: $\mathbb{R}^{W\times H\times K} \rightarrow \mathbb{R}^K$, and feed those image-level guidance $\hat{\mathbf{k}}$ in noise prediction function via concatenating them with 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 non-guided (33) and guided (32) approaches as detailed in Appendix A.

C More experiment results

C.1 Self-labeled Guidance Details

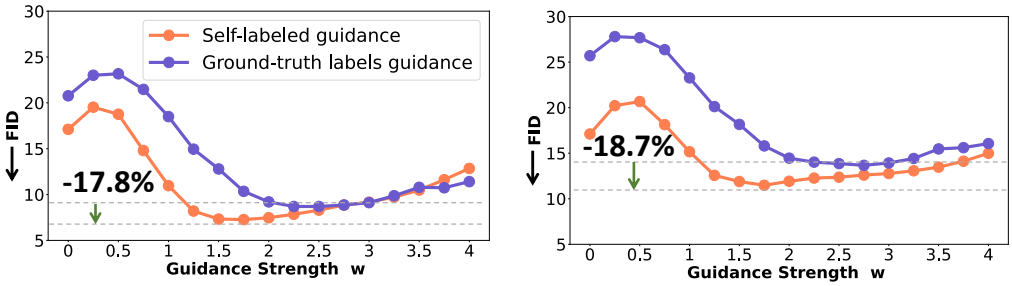
Evaluation metric. The Fréchet Inception Distance (FID) (31) provides a symmetric measure of the distance between two distributions in the feature space of Inception-V3 (73). It is able to evaluate both diversity and fidelity and has become the de facto metric for the evaluation of generative methods, e.g., (19; 36; 9; 66). We use FID as our main metric for the sampling quality.

Implementation details. We follow the implementation from DDPM (33), 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. We use DDIM (70) samplers with 250 steps, $\sigma_t{=}0$. For the learning rate, optimizer, and hyperparameters, please refer to the Appendix F.1. Code will be released.

C.2 Self-boxed Guidance

We use Pascal VOC and COCO_20K to validate the efficacy of self-boxed guidance. To obtain class-agnostic object bounding boxes, we use LOST (68) as our self-annotation function f. We train our diffusion model for 800 epochs with input image size 64×64. 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. See the Appendix F.3 for more details.

Benefit of self-boxed guidance. For the ground-truth guidance baseline, we condition on a class embedding in addition to the bounding box. Since there are now multiple objects per image, we 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 2, confirm that the multi-hot class embedding is indeed effective for non object-centric datasets, improving over the no-guidance model by a huge margin. This improvement comes at the cost of manually annotating multiple classes per image. Self-boxed guidance further improves upon this, by reducing the FID by an additional 5.1 and 3.3 points respectively without using any ground-truth annotation. In Figure 4, we see that our method can generate diverse and semantically aligned images.



(a) Setting I: ImageNet32 balanced

(b) Setting II: ImageNet32 unbalanced

Figure 3: **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 difference 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.

Guidance	Annotation-free?	Pascal VOC	COCO_20K
Ground-truth labels (32)	✗	23.5	19.3
None (33)	✓	58.6	42.5
Self-boxed (***this paper***)	✓	**18.4**	**16.0**

Table 2: **Comparison on Pascal VOC and COCO_20K.** Self-boxed guidance outperforms the ground-truth labels baseline by as much as 20% in terms of FID(↓) on both datasets.

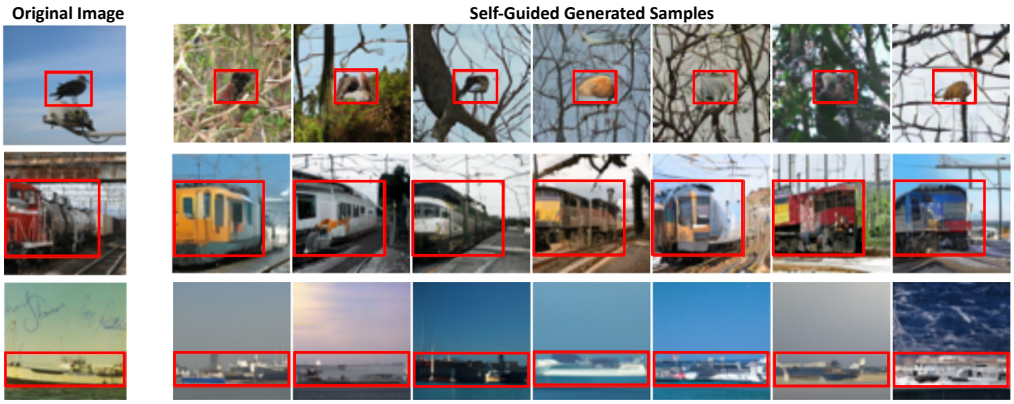


Figure 4: **Same self-boxed guidance signal on Pascal VOC.** Each column is sampled using different random noise. Our method generates visually diverse images that are semantically consistent. Example bounding box predictions by `LOST` can be found in Appendix Figure 23.

Guidance	Annotation-free?	Pascal VOC	COCO-Stuff	
			Train	Val
Ground-truth labels (32)	✗	23.5	16.3	20.5
None (33)	✓	58.6	29.1	34.1
Self-segmented (***this paper***)	✓	**18.2**	**12.5**	**17.7**
Ground-truth masks	✗	12.5	8.1	11.2

Table 3: **Comparison on Pascal VOC and COCO-Stuff.** Any form of guidance results in significant improvement in FID(↓) over the no-guidance model. Self-segmented guidance improves over ground-truth multi-hot labels guidance, narrowing the gap to ground-truth masks guidance.

C.3 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 (27; 34; 17; 79), with a train set of 49,629 images and a validation set of 2,175 images. Classes are merged to form 27 (15 stuff and 12 things) categories. For self-segmented guidance we apply STEGO Hamilton et al. (27) as our self-annotation function f. 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. We set the cluster number to 27 for COCO-Stuff, and 21 for Pascal VOC, following STEGO. More details on the dataset and experimental setup can be found in Appendix F.

Benefit of self-segmented guidance. 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 3 demonstrate that our self-segmented guidance still outperforms the ground-truth labels guidance baseline on all both datasets. The comparison between ground-truth labels and segmentation masks reveals an improvement in image quality when using the more granular 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 5. These examples further highlight a robustness property against noise in the segmentation masks, which our method acquires naturally due to training with noisy segmentations.

D Related Work

Conditional Generative Models. Earlier works on generative adversarial networks (GANs) have already observed improvements in image quality by conditioning on ground-truth labels (49; 9; 13). Recently, conditional diffusion models have seen similar improvements, while also offering a great amount of controllability via classifier-free guidance by training on images paired with textual descriptions, semantic segmentations, or other modalities. This work also aims to realize the benefits of conditioning and guidance, but in contrast, our self-guided diffusion approach achieves this goal solely from images without any manual annotations. Zhou et al. (80) train a GAN for text-to-image generation without any image-text pairs, by leveraging the CLIP (56) model that was pre-trained on a large collection of paired data. In this work, we do not suppose any paired data for the self-annotation models and rely purely on images. Additionally, image layouts are difficult to express with text, thus our self-boxed and self-segmented methods are complementary to text conditioning. Retrieval-augmented Diffusion (7) and KNN-diffusion (3) are two recent methods to explore using nearest neighbor as guidance signal in diffusion model, similar to our work, these methods rely on conditional guidance from an unsupervised pattern, while we focus more on the exploration of overall distribution on the dataset, and further attempt to provide more diverse *spatial* guidance, apart from the image-level signal as they do. Also, Blattmann et al. (7) relies on the feature backbone that needs to be pretrained with image-text pairs, while we can obtain those pairs free via self-supervised learning. Instance-Conditioned GANs (IC-GAN) (13) makes a step to construct a GAN conditioning on instance features, instead, we provide a more fine-grained guidance signal i.e., *(image, box, pixel)* for diffusion model from a self-supervised principle.

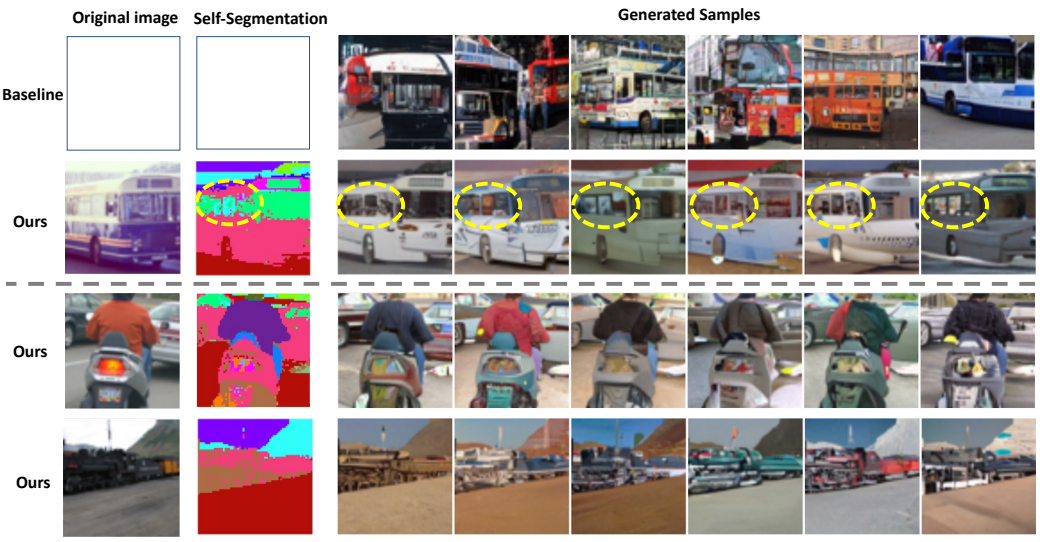


Figure 5: **Same self-segmented guidance signal in Pascal VOC.** Each column is sampled using different random noise. The first two rows indicate our self-segmented guidance can provide more fine-grained guidance than ground-truth labels guidance in the generation of bus. The last two rows shows that our method can create examples that can be well aligned with the self-segmented guidance signal. Specially, from the yellow dotted box, we can easily see that the diffusion model is robust enough to sample textually harmonious images, even when the self-segmented mask is noisy. Colormaps are shared among overall dataset. More examples in Figure 10. Example segmentation mask result by STEGO can be found in Appendix Figure 25.

Self-supervised learning in Generative Models. Self-supervised learning (11; 16; 2; 12) has shown great potential for representation learning in many downstream tasks. As a consequence, it is also commonly explored in generative model for evaluation and analysis (8; 50; 22), conditioning (13; 8; 47), stabilizing training (14), reducing label (46), avoiding mode collapse (1), etc. In contrast, we focus on utilizing self-supervised learning to totally remove the necessity of annotation in diffusion models. Our work carries over the benefits of self-supervised methods to the generative domain, providing flexible guidance signals to diffusion models at various image granularities.

E Extra Discussions

Generative model. Generative model is a widely-used task in deep learning. There are several typical methods to tackle it: Generative Adversarial Network(GAN) (26; 9; 36), Variational Auto Encoder(38; 62), Normalization Flow(61; 20; 39; 40), and Diffusion Models(74; 69; 33; 19; 66). VAE usually fails to capture the key statistics and tends to mode collapse (53; 59). Normalization Flow's design capacity is limited due to its strong requirement for reversibility(41), also, it can encounter instability problem in order to keep the reversibility(6). GAN usually can generate very high fidelity images (9; 36), while its intrinsic probability model property makes it not explainable and unpromising coverage rate (72; 77).

Guidance techniques in diffusion. There are many typical guidance techniques in diffusion models, the main of which are composed of Classifier guidance(19) and Classifier-free guidance(32). Classifier guidance offers an explicit guidance that needs to be trained by an extra classifier with noisy data. To mitigate the problem of extra training, classifier-free guidance, instead, injects the condition information by randomly dropping the condition information during training, by that way, the network is empowered to provide both the conditional and unconditional sampling ability in a single model. Both methods can be interpreted as a way to guide the sampling in an explicit and implicit way. All of them need the image-label pairs, we propose to replace them in an unsupervised pattern. Also, clasifier-free guidance's focus is to trade-off between fidelity and diversity, our focus is to replace the label in a self-supervised principle.

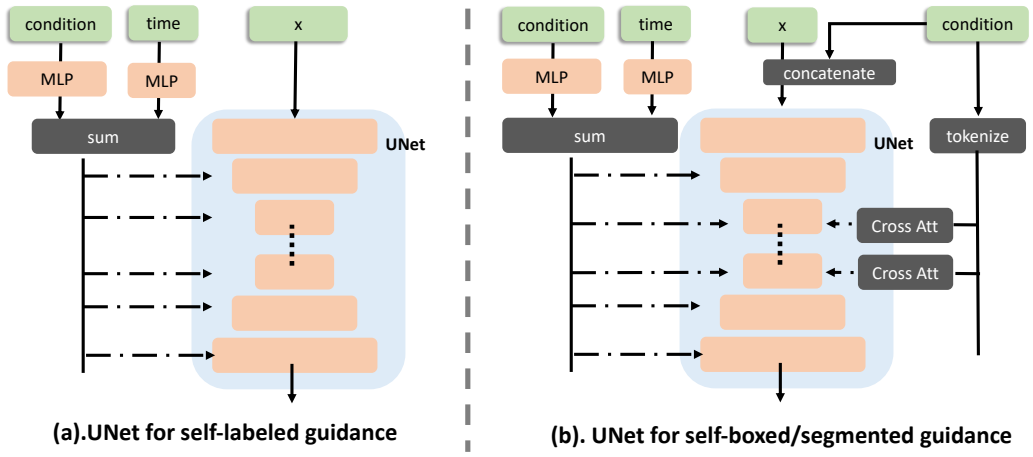


Figure 6: **The detail structure of UNet module.**

F Training details and hyperparameters

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 the Pascal VOC, COCO_20K, COCO-Stuff dataset in image size 64, and visualize them by bilinear up sampling to 256, following (44).

For non object-centric dataset, during training, we randomly crop with an image size of 224 on the original image, then further resize to the size 64 to feed it into the diffusion model.

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 (54), torch-fidelity (51) 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 (5). Parmar et al. (54) revealed that sampling method is vital in the evaluation of generative model, we always use the same bilinear sampling method to make the comparison fair.

For checking point, we pick the checking point by minimal FID between training data and validation data.

For the reference set of FID calculation on complex dataset e.g. Pascal VOC, COCO-Stuff. We randomly crop the original image to 224, and resize the cropped image from 224 to target size 64. By this way, we can generate more diverse reference set owing to the randomness in cropping, we empirically observe more stable evaluation results.

Image-level guidance signal for self-boxed guidance For self-boxed guidance, we need the box-level guidance information and image-level guidance information. Here we detail how we attain the image-level guidance. **1).** We randomly crop the image size 320 to obtain the overall image that represents enough content of the original images. **2).** We resize it from 320 to 224 to make it compatible for `ViT` architecture, and extract the feature via the backbone. **3).** Each image is assigned to a cluster id by k-means clustering on those extracted features.

F.1 UNet structure details

Condition Injection. We describe the condition injection in Figure 6. The injection of self-labeled guidance and self-boxed/segmented guidance is slighted different. The common part is by concatenation between timestep embedding and noisy input x: $\texttt{concat}[\mathbf{k}, t]$, 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 x, the concatenated result will be fed into the UNet as input.

Cross-attention. In training for non object-centric dataset, we also tokenize the condition signal to several tokens following Imagen Saharia et al. (66), 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 (64; 7) is conducted by $\texttt{CA}(\texttt{m}, \texttt{concat}[\mathbf{k}, \texttt{m}])$. Due to the quadratic complexity of transformer (37; 45), we only apply the cross-attention in lower-resolution feature maps.

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

Condition embedding. The condition is in the form of one/multi-hot embedding $\mathbb{R}^K$, we feed it into two-layer MLP: FC(K, 256)→SiLU→FC(256, 256), then feed those guidance signal into the UNet following in Figure 6.

F.2 UNet hyperparameters

3×32×32 model,4GPU,ImageNet32

Base channels: 128	Optimizer: AdamW
Channel multipliers: 1, 2, 4	Learning rate: $3e-4$
Blocks per resolution: 2	Batch size: 128
Attention resolutions: 4	EMA: 0.9999
number of head: 8	Dropout: 0.0
Conditioning embedding dimension: 256	Training hardware: 4 × A5000(24G)
Conditioning embedding MLP layers: 2	Training Epochs: 100
Diffusion noise schedule: linear	Weight decay: 0.0
Sampling timesteps: 256	

3×64×64 model, 4GPU, ImageNet64

Base channels: 128	Optimizer: AdamW
Channel multipliers: 1, 2, 3, 4	Learning rate: $1e-4$
Blocks per resolution: 2	Batch size: 48
Attention resolutions: 4,8	EMA: 0.9999
number of head: 32	Dropout: 0.0
Conditioning embedding dimension: 256	Training hardware: 4 × A5000(24G)
Conditioning embedding MLP layers: 2	Training Epochs: 100
Diffusion noise schedule: linear	Weight decay: 0.0
Sampling timesteps: 256	context token number: 8
context dim: 32	

3×64×64 model, 1GPU, Pascal VOC, COCO_20K, COCO-Stuff

Base channels: 128	Optimizer: AdamW
Channel multipliers: 1, 2, 3, 4	Learning rate: $1e-4$
Blocks per resolution: 2	Batch size: 80
Attention resolutions: 4,8	EMA: 0.9999
number of head: 32	Dropout: 0.0
Conditioning embedding dimension: 256	Training hardware: 1 × A6000(45G)
Conditioning embedding MLP layers: 2	Training Epochs: 800/800/400
Diffusion noise schedule: linear	Weight decay: 0.0
Sampling timesteps: 256	context token number: 8
context dim: 32	

F.3 Dataset preparation

The preparation of imbalanced 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. For self-boxed guidance, for Pascal VOC, we use the standard split from (68). 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. We empirically observed that using more reference set e.g, 50,000 images don't make too much difference.

COCO_20K. For self-boxed guidance, COCO_20k, we follow the split from (68; 75; 43). COCO_20k is a subset of the COCO2014 trainval dataset, consisting of 19,817 randomly chosen images, used in unsupervised object discovery (68; 75). We sample 10,000 images and use 10,000 random-croped 64-sized train images as reference set for FID evaluation. We empirically observed that using more reference set e.g, 50,000 images don't make too much difference.

COCO-Stuff. COCO-Stuff has a train set of 49,629 images, validation set of 2,175 images, where the original classes are merged to form 27 (15 stuff and 12 things) high-level categories. We use the dataset split following (27; 34; 17; 79), we also report validation FID in COCO-Stuff dataset, we randomly crop $(224, 224)$ on those 2,175 images, and resize them to our target size 64. We sample 10,000 images and use 10,000 train/validation images as reference set for FID evaluation. We empirically observed that using more reference set e.g, 50,000 images don't make too much difference.

F.4 More details about LOST, STEGO algorithms

LOST algorithm details. In detail, we conduct padding to make the original image can be patchified to be fed into the `ViT` architecture (21), and feed the original padded image into the `LOST` architecture using official source code [1]. We empirically found a slightly over-clustered cluster number $k = 100$ performs best. `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 here.

STEGO algorithm details. For `STEGO` (27), we use the backbone `ViT-B` following the official source code [2], and we 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 KNN, the number of neighbors for KNN 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.

F.5 Clustering algorithm details

We use efficient k-means clustering implementation from `FAISS` (35). We conduct only k-means clustering on training set of the dataset.

F.6 Feature backbone experiment details

For feature preparation of `ViT`-based backbones, as the `ViT`-structure requires that the image can be patchified by size 8 or 16, we resize the images up to 224, then feed them into different backbones before feature extraction. We exactly follow the standard data transformation following respective backbones.

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

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

G More qualitative results

G.1 Uncurated sample visualization from non object-centric dataset

We show uncurated visualization of self-segmented guidance for COCO-Stuff dataset in Figure 8 and Figure 7.

We also show the denoising process of our self-segmented diffusion in COCO-stuff dataset in Figure 9.

We show more self-segmented guidance samples on COCO-Stuff dataset in fig. 10.

We show the visualization of gradually increasing guidance strength w in Figure 11.

G.2 Uncurated sample visualization from ImageNet32/64

We show uncurated visualization for ImageNet32 in Figure 15, ImageNet64 in Figure 14.

We also show uncurated sampling results for ground-truth labels guidance, and self-labeled guidance in Figure 13, Figure 12 respectively.

G.3 The KNN Query visualization

We wish to understand how well the distribution of the real images are intertwined with the distribution of the samples. We conduct a KNN query experiment of ImageNet32 in Figure 16.

G.4 Denoising process visualization

We show the denoising process visualization of ImageNet64 in Figure 17, Pascal VOC in Figure 18.

G.5 Interpolation between different conditions

We show the interpolation result of ImageNet64 in Figure 20.

G.6 Cluster visualization

We show the cluster visualization of real images for ImageNet32 in Figure 21, Pascal VOC in Figure 22.

G.7 Intermediate result of LOST, STEGO algorithm

We show the intermediate result of `LOST` in Figure 23, Figure 24, and `STEGO` in Figure 25.

Figure 7: **Self-segmented guidance samples(uncurated)** from COCO-Stuff dataset. Best viewed in color.

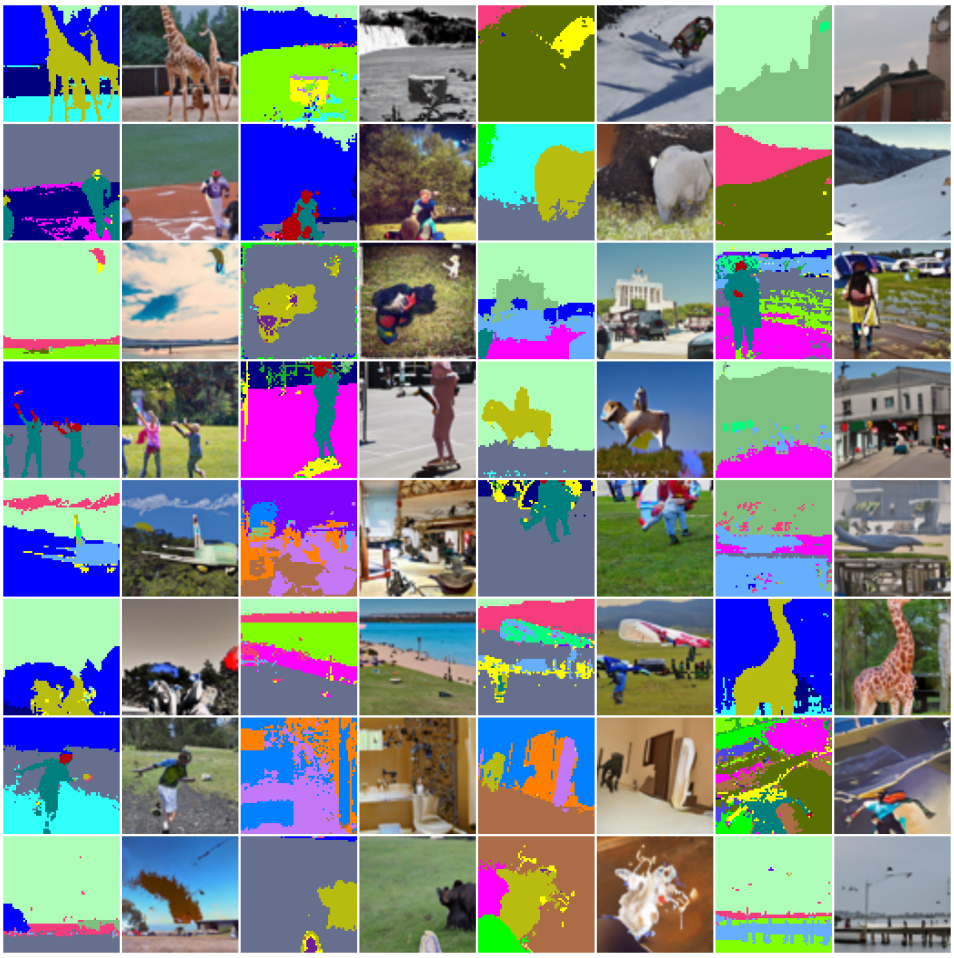


Figure 8: **Self-segmented guidance samples(uncurated)** from COCO-Stuff dataset companies with segmentation mask from STEGO (27). Color map are shared among overall dataset. Best viewed in color.

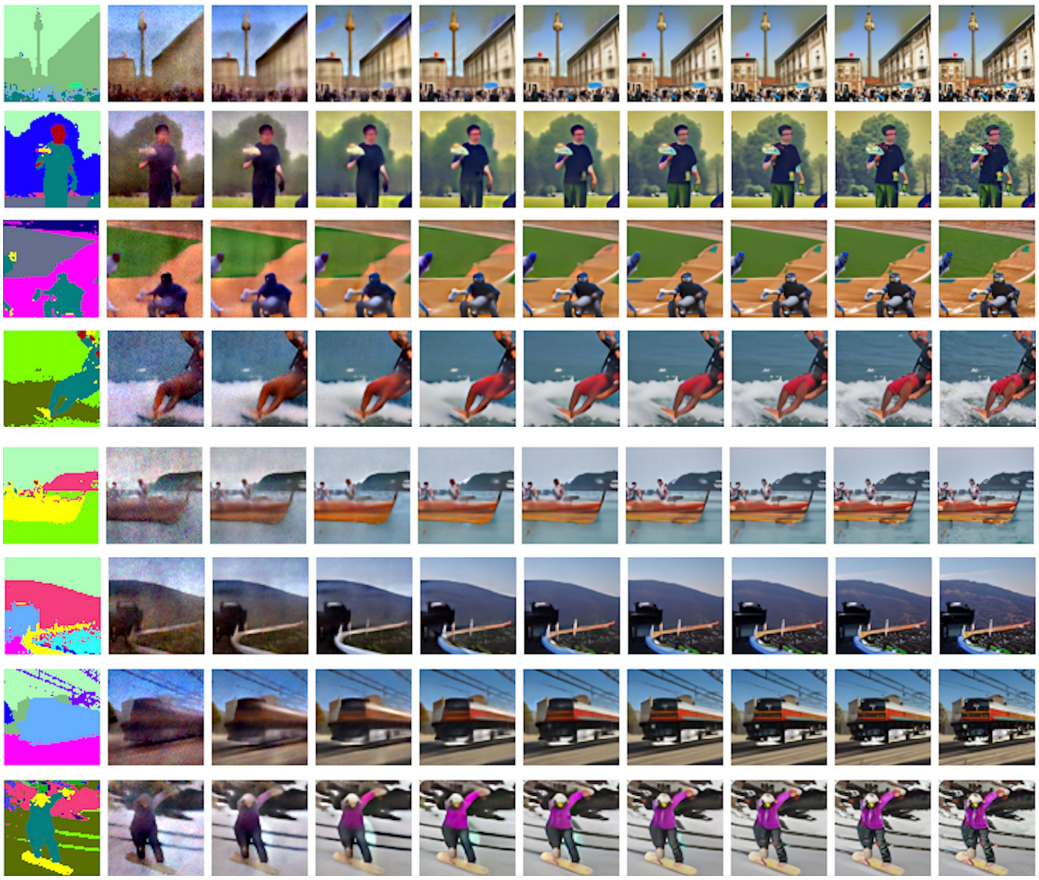


Figure 9: **Denoising process of self-segmented guidance samples(uncurated)** from COCO-Stuff dataset. First column is the condition mask from STEGO (27), The remaining columns are from the most noisy period to less noisy period. Best viewed in color.

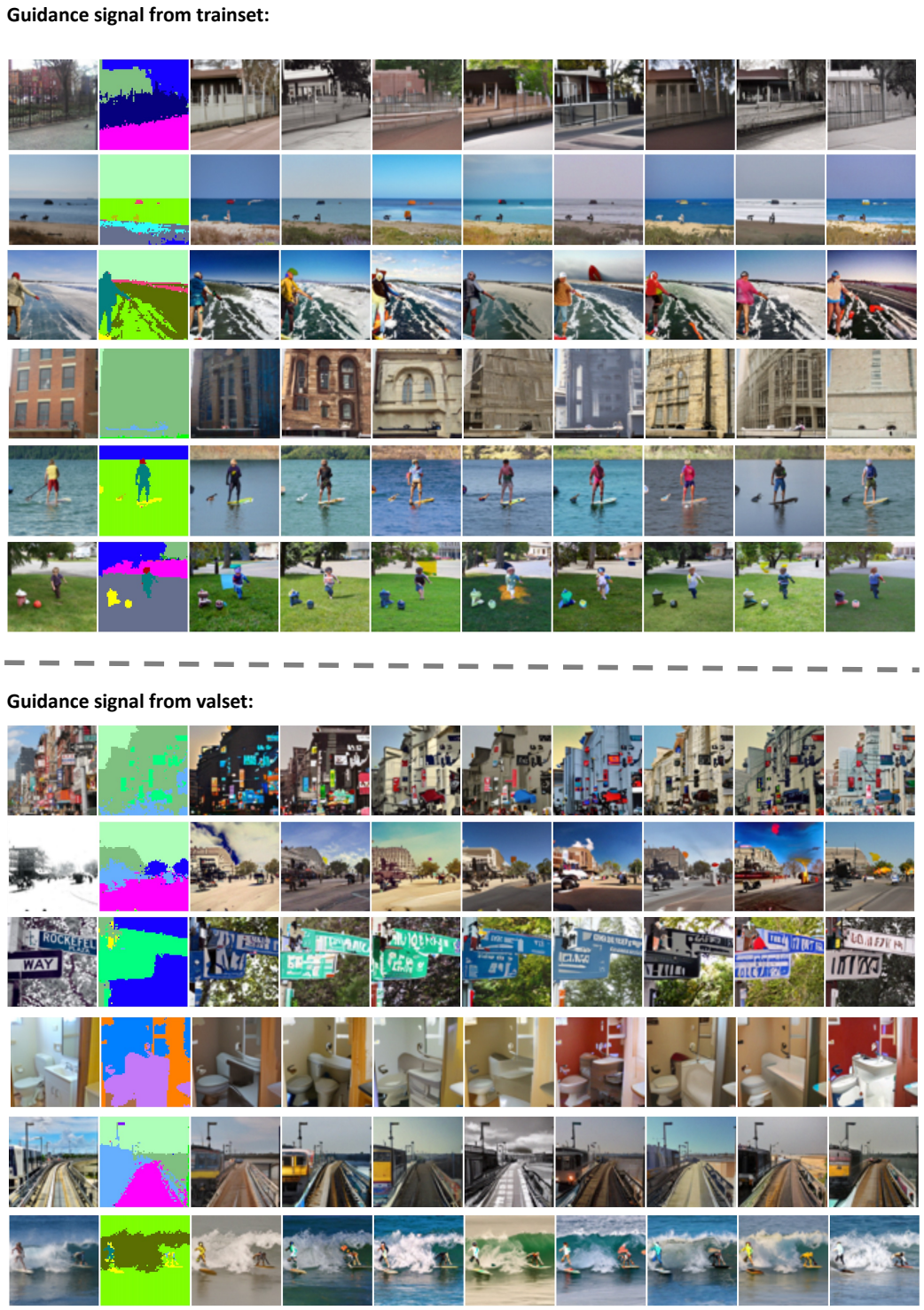


Figure 10: **Self-segmented guidance samples(uncurated)** from COCO-Stuff dataset. First column is the real image where we attain the conditional mask. Second column is the seg-segmented mask we obtain from STEGO (27), The remaining columns are the random samples conditioning on the same self-segmented mask. Best viewed in color.

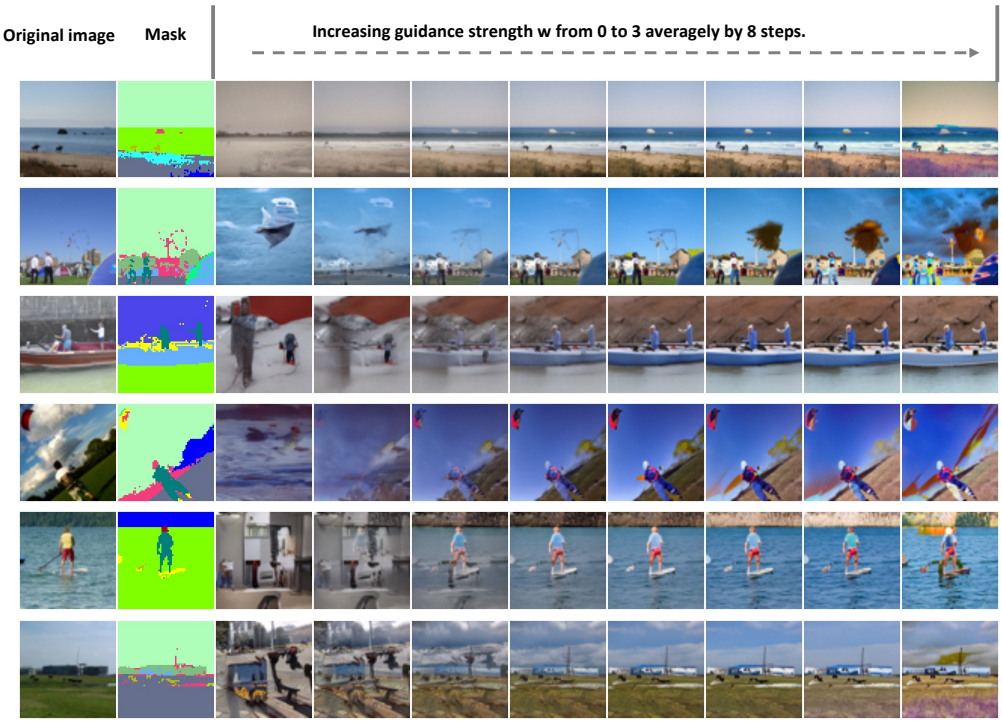


Figure 11: **Self-segmented guidance samples** from Pascal VOC dataset. First column is the real image where we attain the conditional mask. Second column is the seg-segmented mask we obtain from STEGO (27), The remaining columns are the visualization when we averagely increase guidance strength w from 0 to 3 by 8 steps. Best viewed in color.

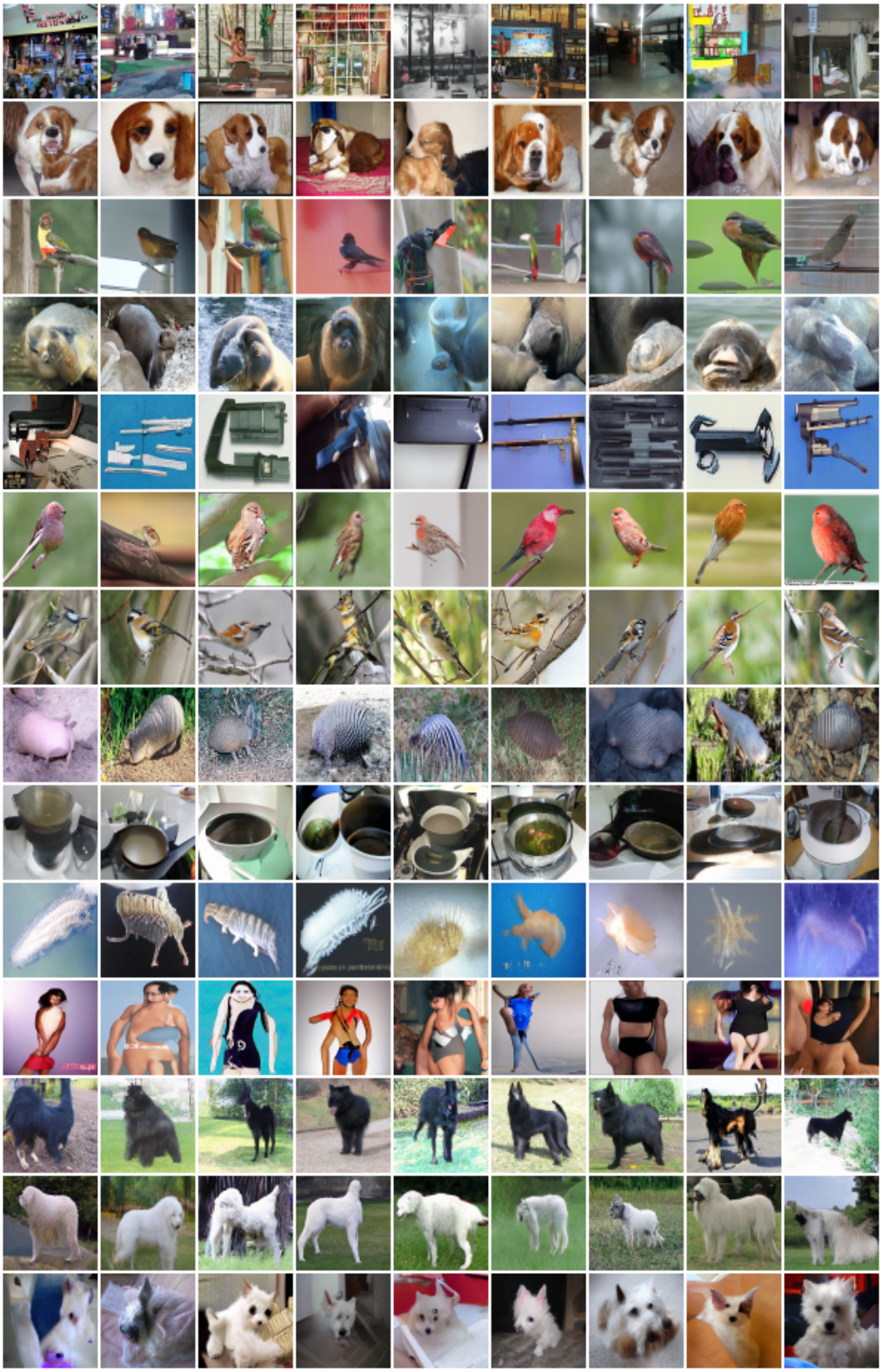


Figure 12: **Self-labeled guidance samples(uncurated) conditioning on the same guidance signal from ImageNet64.** Best viewed in color.

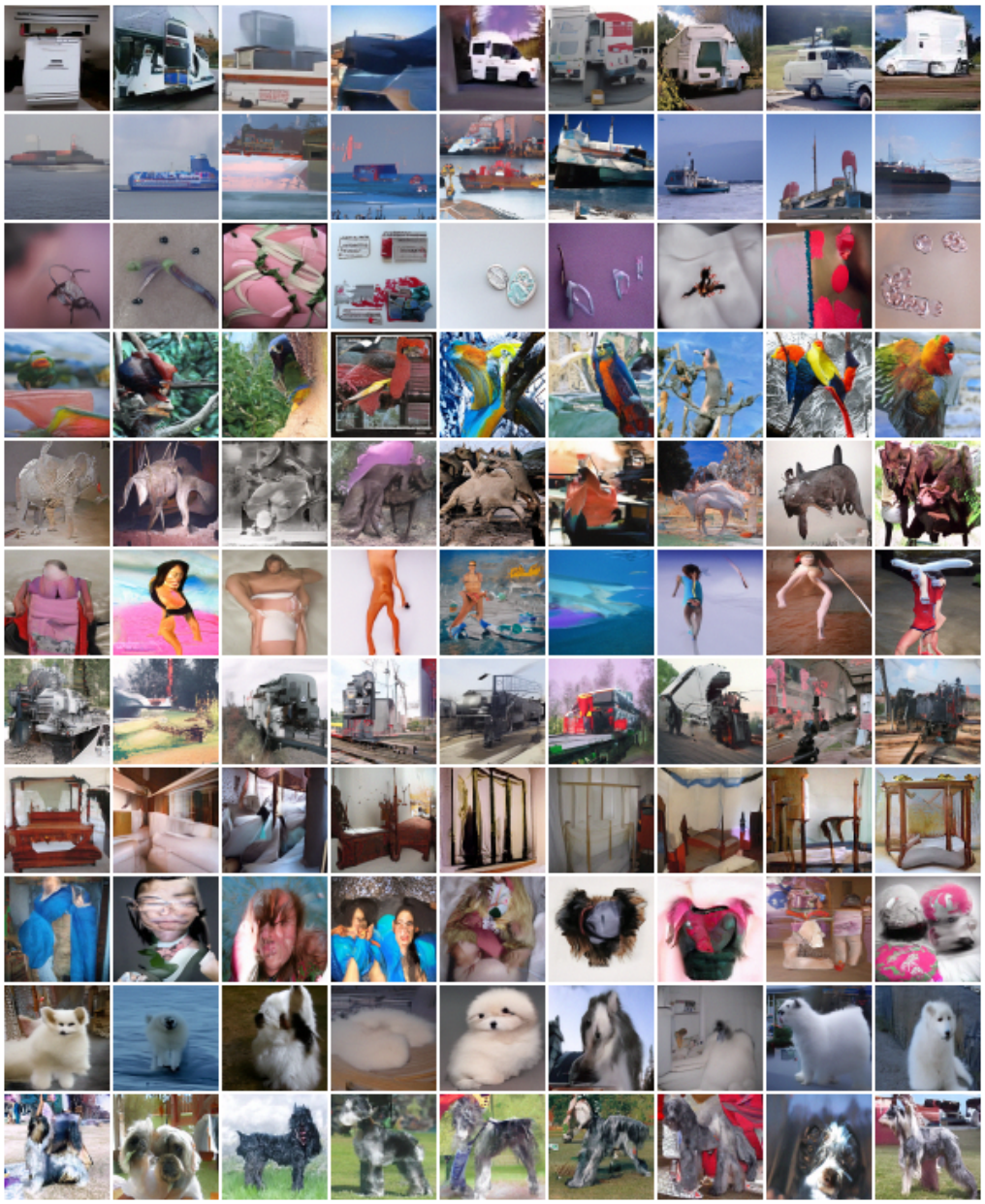


Figure 13: **Ground-truth labels guidance samples(uncurated) conditioning on the same guidance signal from ImageNet64.** Best viewed in color.

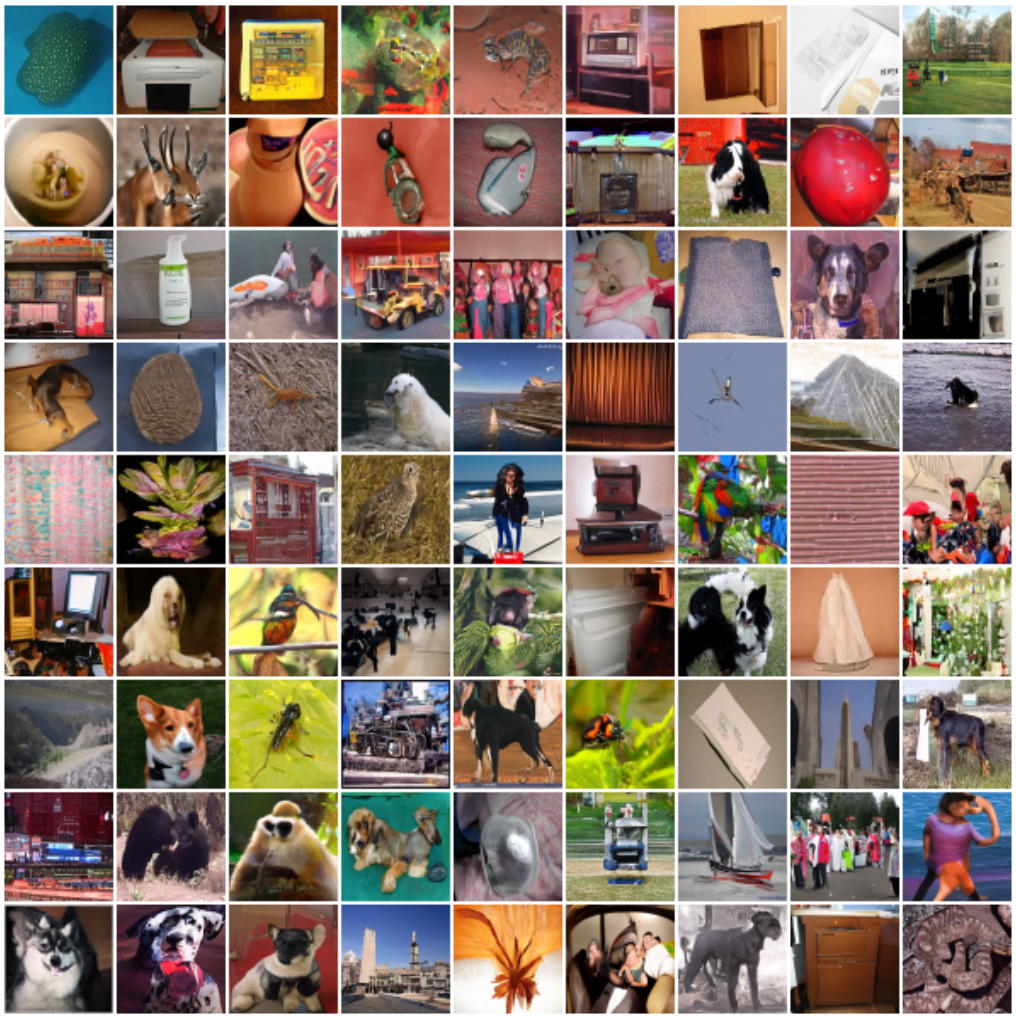


Figure 14: **Self-labeled guidance samples(uncurated) from ImageNet64.** Best viewed in color.

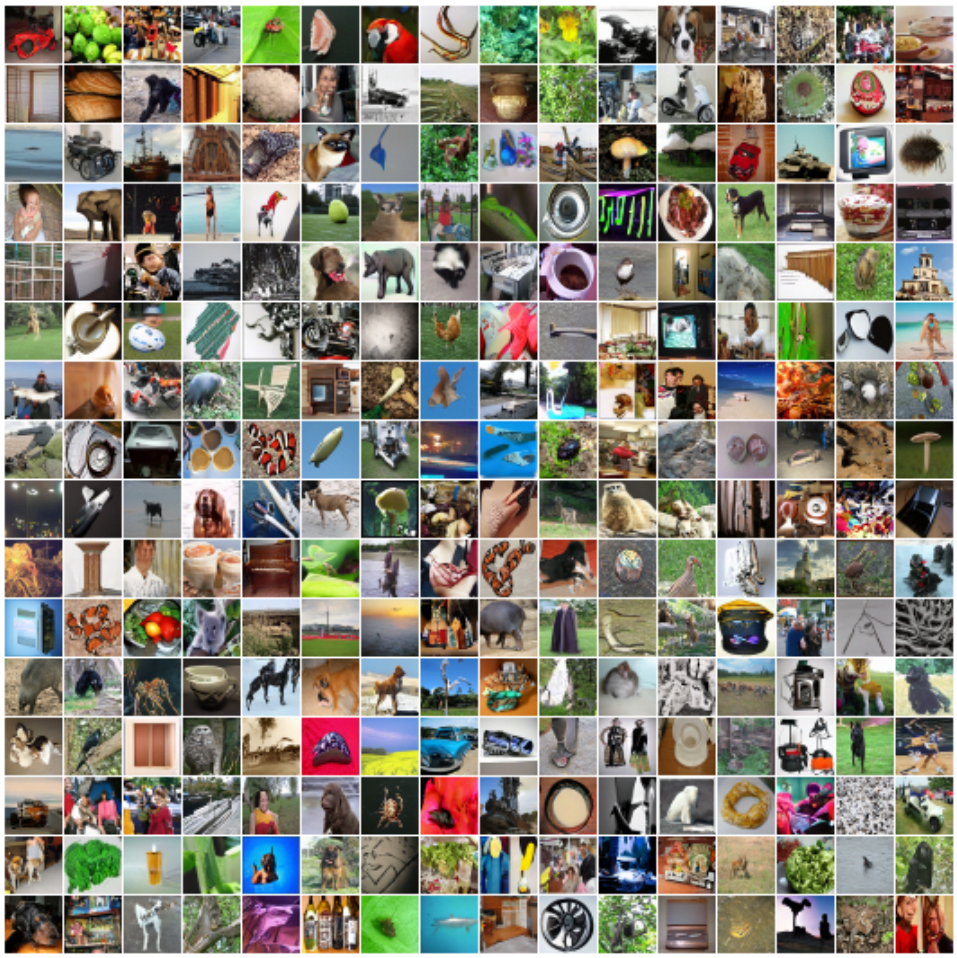
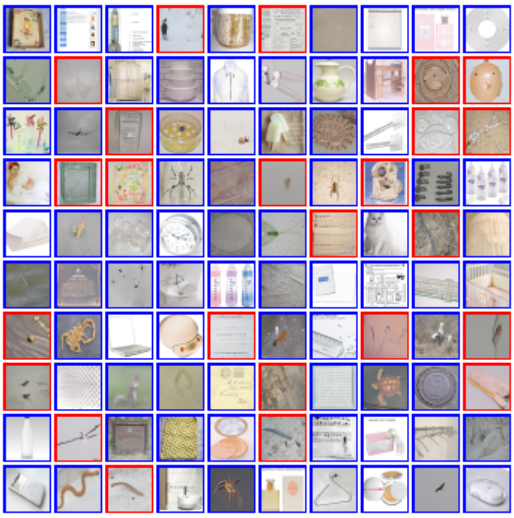


Figure 15: **Self-labeled guidance samples(uncurated) from ImageNet32.** Best viewed in color.

(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 16: **KNN query result visualization.** Blue means samples, red means real images. Images are ordered from left to right, top to down, by SimCLR (16) feature similarity or pixel similarity. Sampled images are sampled by DDIM (70) 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. We can easily see, regardless of the feature space or image space, the KNN 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 17: **Denoising process for ImageNet64.**

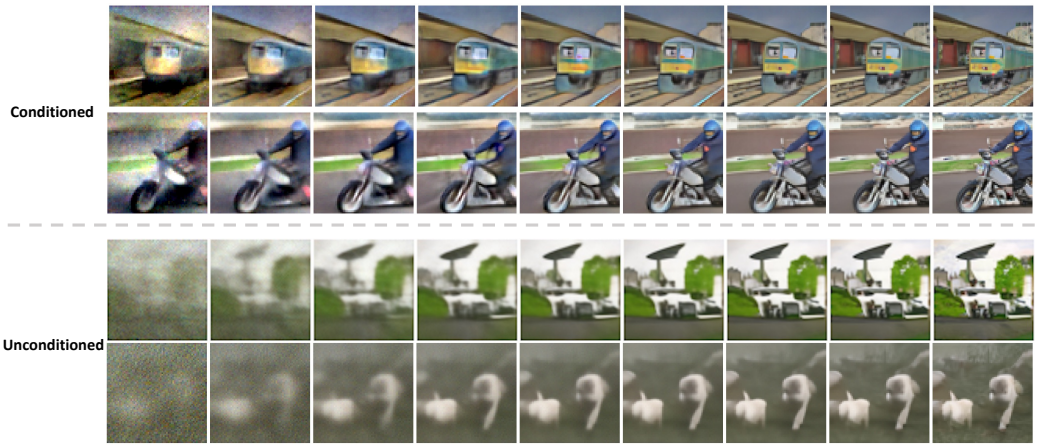
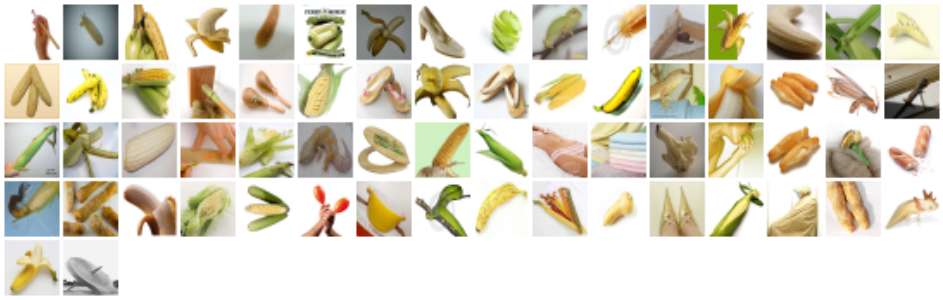


Figure 18: **Denoising process for Pascal VOC dataset.** The first two rows are sampled from guidance strength $w = 2$ using our self-segmented guidance, the last two rows are sampled from guidance strength $w = 0$. By conditioning on our self-segmented guidance, the denoising becomes easier and faster, this efficient denoising aligns with the observation from (55).

Figure 19: **Sample visualization with guidance strength w from 0 to 3 in ImageNet64.** Best viewed in color.

Figure 20: **Sphere interpolation between two random self-labeled guidance signals on ImageNet64.** Best viewed in color.

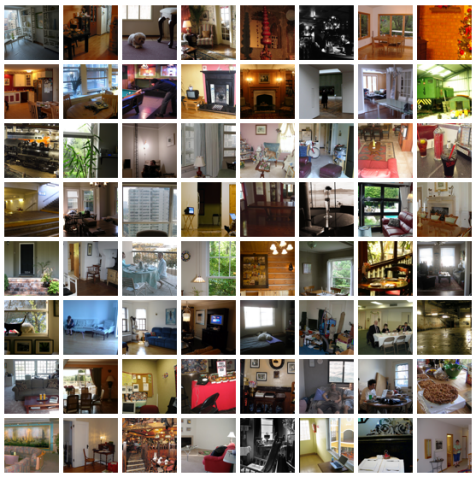
(a) cluster625.



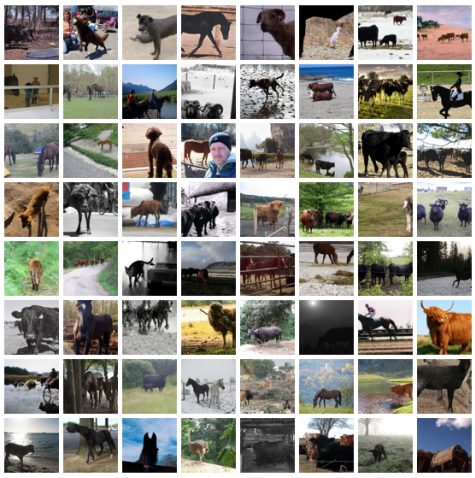
(b) cluster807

(c) cluster890

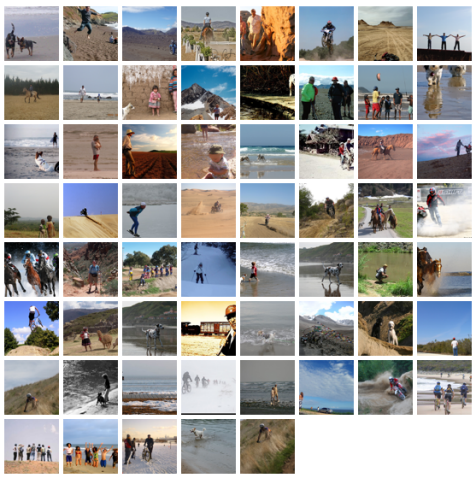
Figure 21: **Real images in ImageNet32 dataset that belong to the same cluster after k-means.**



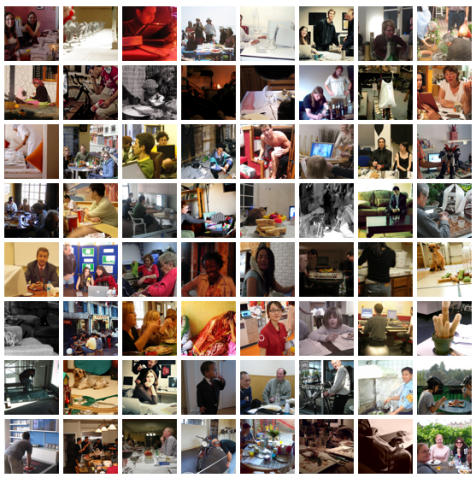
(a) cluster17



(b) cluster18



(c) cluster45



(d) cluster50

Figure 22: **Real images in Pascal VOC dataset that belong to the same cluster after k-means** Best viewed by zooming in.

Figure 23: **Bounding box result from LOST on Pascal VOC dataset**. As LOST (68) is an unsupervised-learning method, some flaws in the generated box are expected. Images are resized squarely for better visualization.

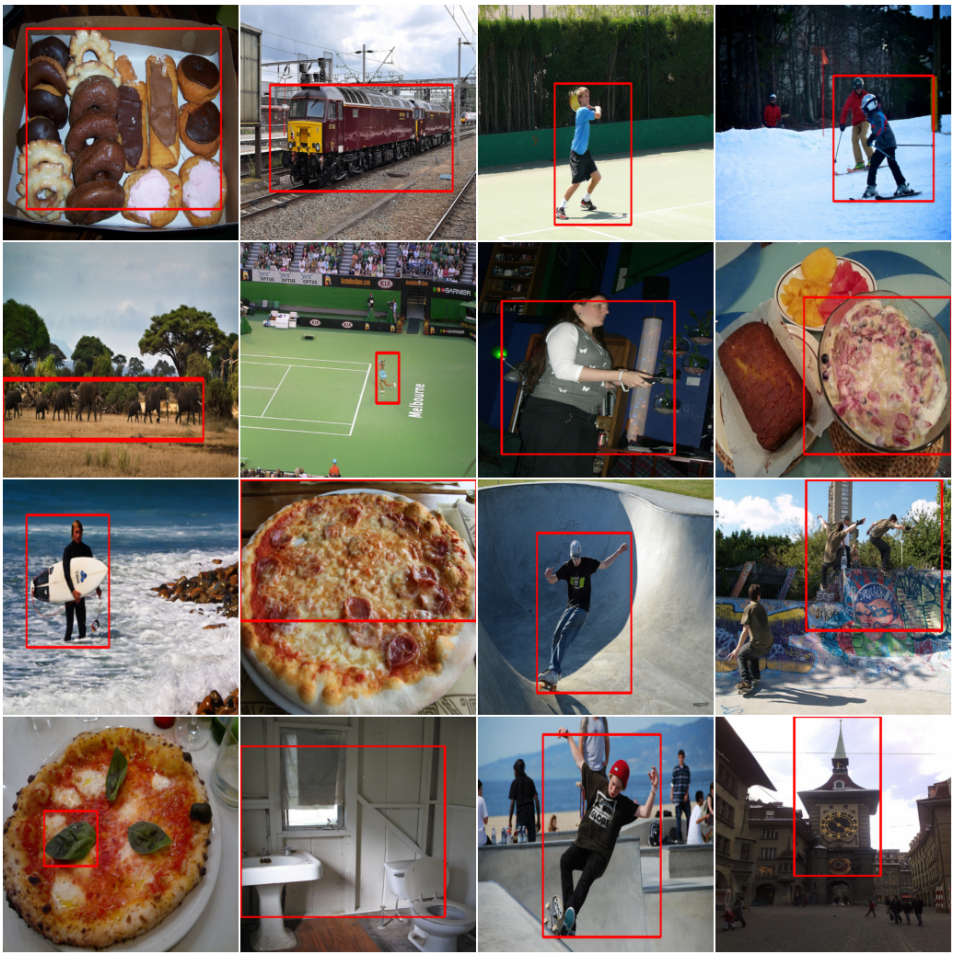


Figure 24: **Bounding box result from `LOST` (68) on COCO_20K dataset.** Images are resized squarely for better visualization.

Figure 25: **Segmentation mask result from STEGO on Pascal VOC dataset**. Cluster number k is 21. Images are resized squarely for better visualization. Color map are shared among overall dataset. Best viewed in color.

Table 4: **Extra cluster k helps?**

Cluster	Feat	FID	IS
k=10k	ViT/s16	18.2	10.4
w/ k=1k	ViT/s16	14.9	10.9
w/ k=10k	ViT/s16	15.5	11.1
w/ k=10k	ViT/b16	14.9	11.6

Table 5: **Greyscale feature helps?**

Method	FID	IS
w/o greyscale	18.2	10.4
w/greyscale	16.2	11.3

Table 6: **Clustering on PCA helps?**

Method	FID	IS
w/o PCA	18.2	10.4
w/ PCA	14.4	12.1

Table 7: **Label information helps?**

Method	FID	IS
w/o label	18.2	10.4
w/ label	17.1	11.6

H More quantitative results

H.1 Extra ablations

We explore whether we can ensemble the performance by adding extra cluster embedding, greyscaled-image cluster embedding, PCA clustering embedding in Table Table 4, Table 5, Table 6, respectively. We always observe some gain when deploying those information together with our baseline self-labeled guidance with $k = 10{,}000$. Interestingly, when we conduct the similar experiments with baseline of self-labeled guidance $k = 5{,}000$, we cannot observe those obvious gains, sometimes they can even deteriorate the result in comparison with baseline. This suggests that: 1). $k = 5,000$ is indeed a sweet spot for ImageNet dataset, those extra emsembling operation can't reap abundant gain. 2). This can be helpful when we don't know the sweet spot, we can easily conduct a similar ensemble to further dig out the potential gains. It's worth to mention that our self-labeled diffusion enables us to explore those choices to further boost the performance, which could not be easily provided by ground-truth label guidance.

Label information can further improve the self-labeled cluster information. In Table 7, we concatenate the label information(one-hot embedding) and cluster information(multi-hot embedding) together and feed them into diffusion model, we observe the best performance. This suggests that our cluster information can be borrowed by free as a plugin guidance information to further improve the performance. Also, the clustering method might not only cluster on the semantic-level of human-annotated label, but also focus some more fine-grained features.

H.2 Precision and Recall in ImageNet32/64 dataset

We show the related results in Table 8. We follow the evaluation code of Precision and Recall from ICGAN (13), our self-labeled guidance also outperforms the baseline by a large margin in precision and remains competitive in recall.

Guidance	ImageNet32				ImageNet64			
	FID↓	IS ↑	P ↑	R ↑	FID↓	IS↑	P	R
None (33)	14.3	10.8	0.49	0.61	43.1	10.38	0.59	0.60
ground-truth labels	10.2	19.0	0.71	0.62	16.8	18.6	0.71	0.62
self-labelled (*this paper*)	**7.3**	**20.3**	**0.77**	**0.63**	**12.1**	**23.1**	**0.78**	0.62

Table 8: **Comparison with baseline on ImageNet32 and ImageNet64 dataset.** Our self-labelled guidance outperform the baseline by a large margin in both datasets.

cluster number k	**FID** $\downarrow$
21	22.5
50	18.5
100	18.5

Table 9: **Choice of cluster number on Pascal VOC dataset for self-boxed guidance.**

H.3 Ablation about cluster number in self-boxed guidance

In table 9, 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 $k = 100$.