Contents

A. More results and details

A.1. Results on Churches-256

Implementation details. We directly use the official code of Latent Diffusion Model[1] [53], and reduce the base channel number from 192 to 128 and attention resolution from [32, 16, 8, 4] to [8, 4] to accelerate training. Note that these changes significantly reduce the number of parameters from 294M to 108M.

Qualitative and Quantitative result. We present more qualitative results in Figure 8. We use the FID metric for quantitative comparison. For non-guidance and our self-labeled guidance, we get an FID of 23.1 and 16.2 respectively. Our self-labeled guidance improves by almost 7 points for free.

A.2. Precision and Recall in ImageNet32/64 dataset

We show the extra results of ImageNet on precision and recall in Table 6. We follow the evaluation code of precision and recall from ICGAN [12], our self-labeled guidance also outperforms ground-truth labels in precision and remains competitive in the recall.

A.3. Correlation between NMI and FID in different feature backbones.

Normalized Mutual Information (NMI) can be used to assess the performance in self-supervised representation learning. It measures the similarity between the cluster assignments and the ground-truth labels. We examine whether there is a relation between the quality of the self-supervised method, as it is typically measured, and the FID resulting from the clusters induced by the self-supervised features. In Figure 9 we plot the NMI and FID for different self-supervised models. The models trained with ground-truth labels show no change in FID for different NMI values. In contrast, the self-supervised models exhibit a negative correlation between the NMI and FID, suggesting that NMI is also predictive of the model's usefulness in our setting. This indicates that future progress in self-supervised learning will also translate to improvements to self-labeled guidance.

A.4. Varying guidance strength w

We consider the influence of the guidance strength w on our sampling results. We mainly conduct this experiment in 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. We train both models for 100 epochs. For the standard ImageNet32 validation setting in Figure 10a, our method achieves a 17.8% improvement for the 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 10b, 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 [35], and clustering based on the statistics of the overall dataset can potentially lead to more robust performance in an unbalanced setting.

A.5. Cluster number ablation in self-boxed guidance

In Tab. 7, 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 $k = 100$.

A.6. Trend visualization of training loss and validation FID

We visualize the trend of training loss and validation FID in Figure 11.

B. 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, and 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, and COCO-Stuff in image size 64, and visualize them by bilinear upsampling to 256, following [37].

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 [46], torch-fidelity [44] 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]. For the checkpoint, we pick the checking point every 10 epochs by minimal FID between generated sample set and the train set.

B.1. UNet structure

Guidance signal injection. We describe the detail of guidance signal injection in Figure 12. 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

[1] https://github.com/CompVis/latent-diffusion

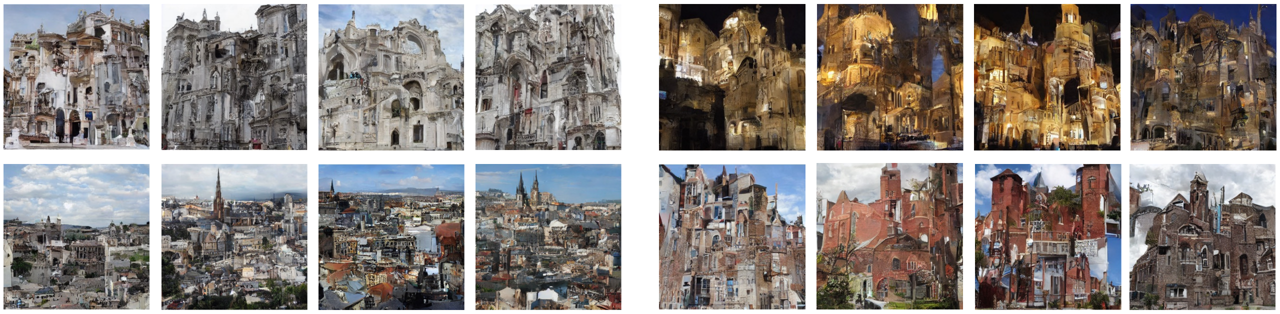


Figure 8. **Generated samples using self-labeled guidance on LSUN-Churches 256×256.** Each row corresponds to a different cluster. Clusters can capture concepts like nighttime, a far shot that includes the city, a close shot of the church, and the church's color.

Diffusion Method	**Annotation-free?**	**ImageNet32**				**ImageNet64**			
		FID↓	IS ↑	P ↑	R ↑	FID↓	IS↑	P↑	R ↑
Ground-truth labels guidance	✗	9.2	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 guidance	✓	**7.3**	**20.3**	**0.77**	**0.63**	**12.1**	**23.1**	**0.78**	**0.62**

Table 6. **Comparison with baseline on ImageNet32 and ImageNet64 dataset with FID, IS, Precision (P), Recall (R).**

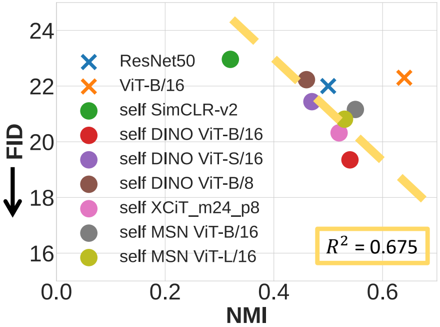


Figure 9. **Correlation between NMI and FID** on ImageNet32. The Normalized Mutual Information (NMI) is not related to FID for supervised backbones, while, for the self-supervised model, NMI and FID are negatively correlated.

Cluster number k	**FID ↓**
21	22.5
50	18.6
100	18.5

Table 7. **Cluster number ablation on Pascal VOC dataset for self-boxed guidance.**

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)→SiLU→FC(128, 128).

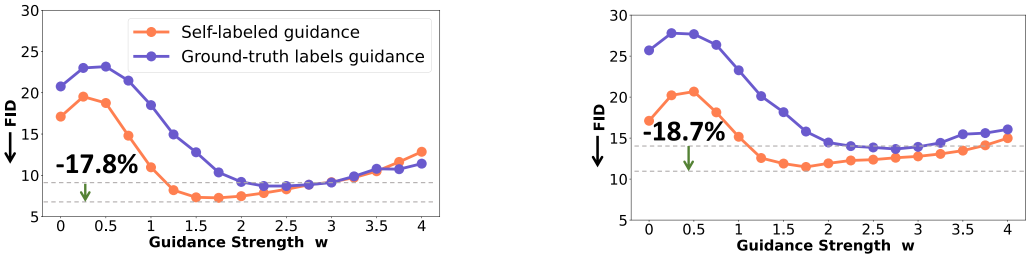
Guidance embedding. The guidance is in the form of one/multi-hot embedding $\mathbb{R}^K$, we feed it into two-layer MLP: FC(K, 256)→SiLU→FC(256, 256), then feed those guidance signal into the UNet following in Figure 12.

Cross-attention. In training for non object-centric dataset, we also tokenize the guidance signal to several tokens following Imagen [55], 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 [6, 53] is conducted by $\mathtt{CA}(\mathtt{m}, \mathtt{concat}[\mathbf{k}, \mathtt{m}])$. Due to the quadratic complexity of transformer [31, 38], we only apply the cross-attention in lower-resolution feature maps.

B.2. Training Parameter

B.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 $\lfloor i \times 50/1000 \rfloor = \lfloor i/200 \rfloor$.



(a) Setting I: ImageNet32 balanced

(b) Setting II: ImageNet32 unbalanced

Figure 10. **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 strengths. The difference between them is slightly more prominent for unbalanced data, we conjecture that this 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.

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: 0.01
Sampling timesteps: 256	

Table 8. **3×32×32 model, 4GPU, ImageNet32.**

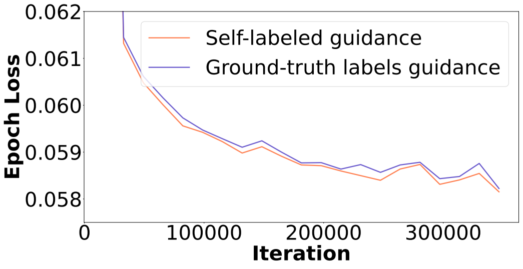


Figure 11. **Epoch loss trend.**

Pascal VOC. We use the standard split from [57]. It has 12,031 training images. As there is no validation set for Pascal VOC dataset, therefore, we only evaluate FID on the train set. We sample 10,000 images and use 10,000 random-cropped 64-sized train images as reference set for FID evaluation.

COCO_20K. We follow the split from [36, 57, 64]. COCO_20k is a subset of the COCO2014 trainval dataset, consisting of 19,817 randomly chosen images, used in unsupervised object discovery [57,64]. We sample 10,000 images and use 10,000 random-cropped 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 [15,22,29,70], We sample 10,000 images and use 10,000 train/validation images as reference set for FID evaluation.

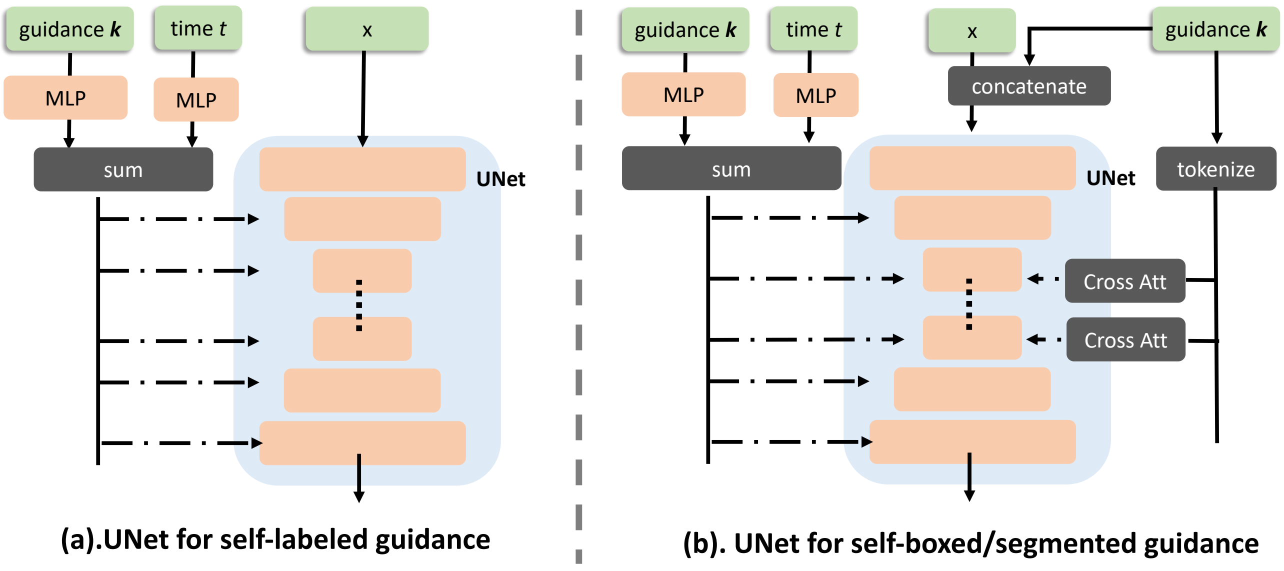
B.4. LOST, STEGO algorithms

LOST algorithm details. We conduct padding to make the original image can be patchified to be fed into the `ViT` architecture [18], 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.

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

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



(a).UNet for self-labeled guidance

(b). UNet for self-boxed/segmented guidance

Figure 12. **The structure of UNet module.**

Base channels: 128	Optimizer: AdamW
Channel multipliers: 1, 2, 4	Learning rate: $1e-4$
Blocks per resolution: 2	Batch size: 48
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: 0.01
Sampling timesteps: 256	

Table 9. **$3\times64\times64$ model, 4GPU, ImageNet64.**

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 $5e-4$, the batch size is 64.

C. Qualitative results

C.1. Assigning semantic descriptions in self-labeled/segmented guidance

In order to control the semantic content of a sample using self-guidance we can assign descriptions to each self-supervised cluster by manually checking a few example images per cluster. This is much more scalable since the total number of training images available are multiple orders of magnitude greater than the number of clusters. Furthermore, images in the same self-supervised cluster are highly semantically coherent and humans can easily describe their shared abstract concept [34].

In Figure 14 we show examples of self-labeled guidance that highlight the semantic coherence of samples guided by the same cluster id. In Figure 13 we show how this approach is also extendable to self-segmented guidance.

C.2. More qualitative results

Base channels: 128	Optimizer: AdamW
Channel multipliers: 1, 2, 4	Learning rate: $1e-4$
Blocks per resolution: 2	Batch size: 80
Attention resolutions: 4	EMA: 0.9999
Number of head: 8	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: 0.01
Sampling timesteps: 256	Context token number: 8
Context dim: 32	

Table 10. **3×64×64 model, 1GPU, Pascal VOC, COCO_20K, COCO-Stuff.**

Base channels: 128	Optimizer: AdamW
Channel multipliers: 1, 2,2,3, 4	Learning rate: $5e-5$
Blocks per resolution: 2	Batch size: 48
Attention resolutions: 4,8	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 Steps: 600k
Diffusion noise schedule: linear	Weight decay: 0.01
Sampling timesteps: 200	

Table 11. **3×256×256 model, 4GPU, Churches-256.**

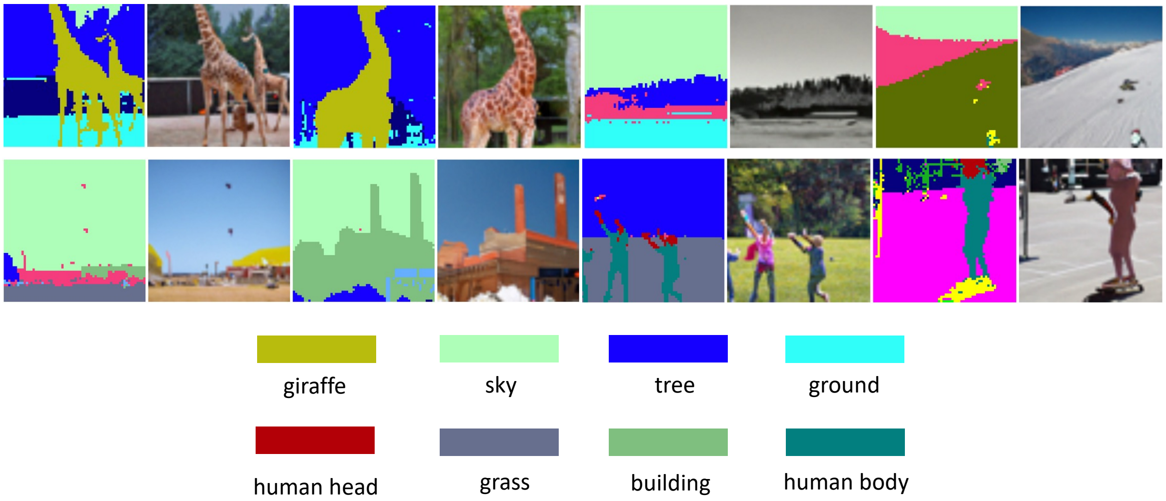


Figure 13. **Self-segmented guidance samples from COCO-Stuff companies with segmentation mask from STEGO [22].** The color map is shared among the overall dataset. The semantic description is deduced based on a few images. Best viewed in color.

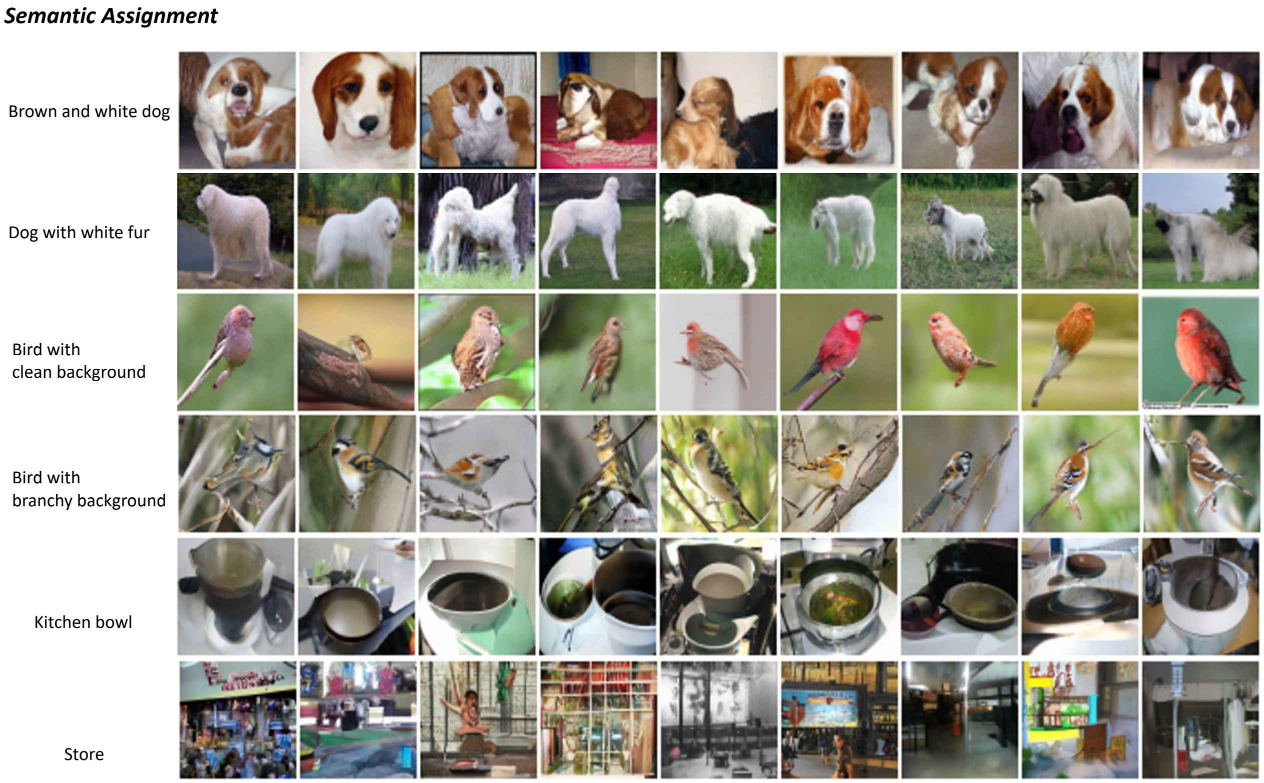


Figure 14. **Self-labeled guidance samples conditioning on the same guidance from ImageNet64.** We assign a cluster description based on a few sample images. Best viewed in color.

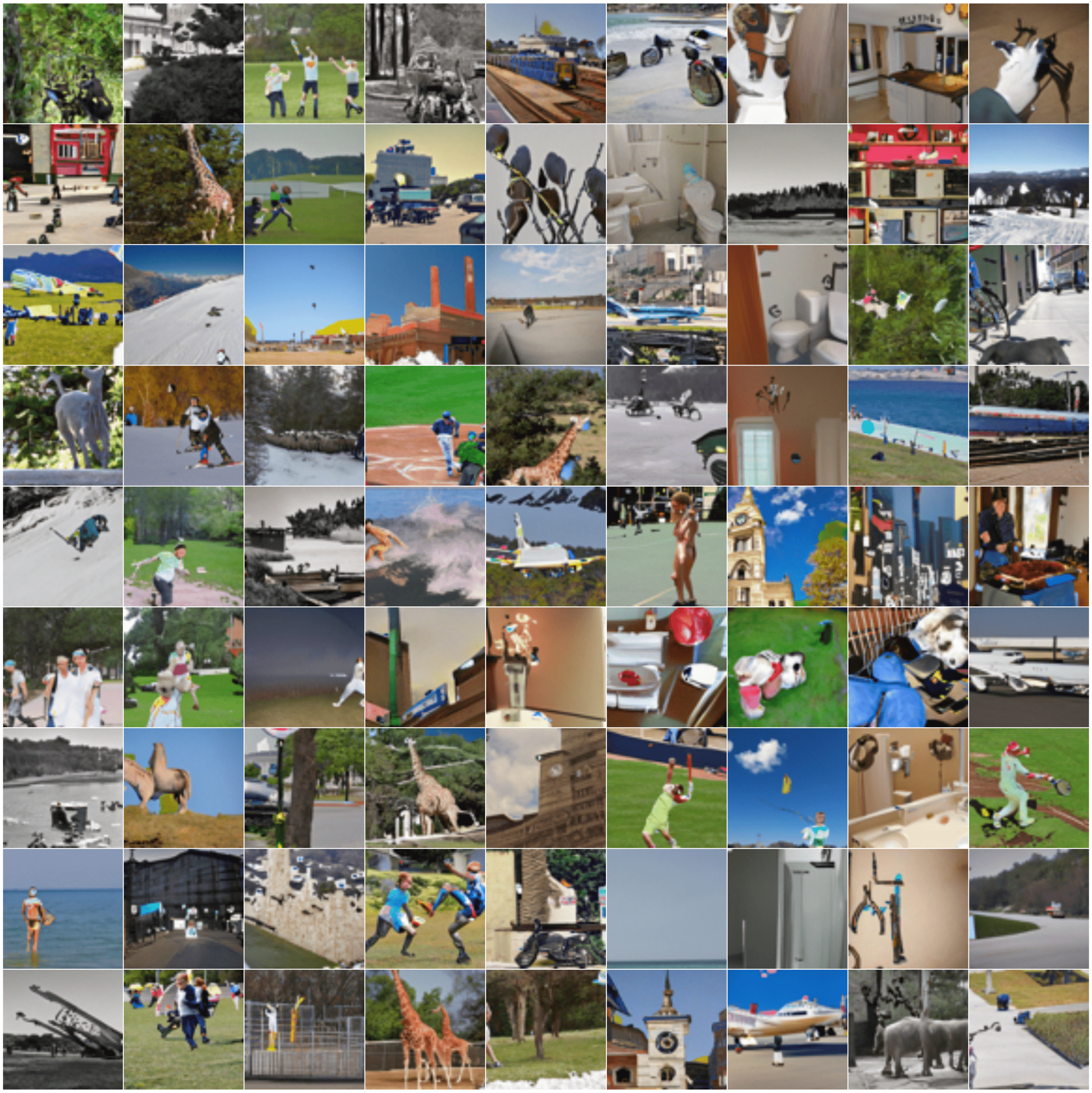


Figure 15. **Self-segmented guidance samples from COCO-Stuff.** Best viewed in color.

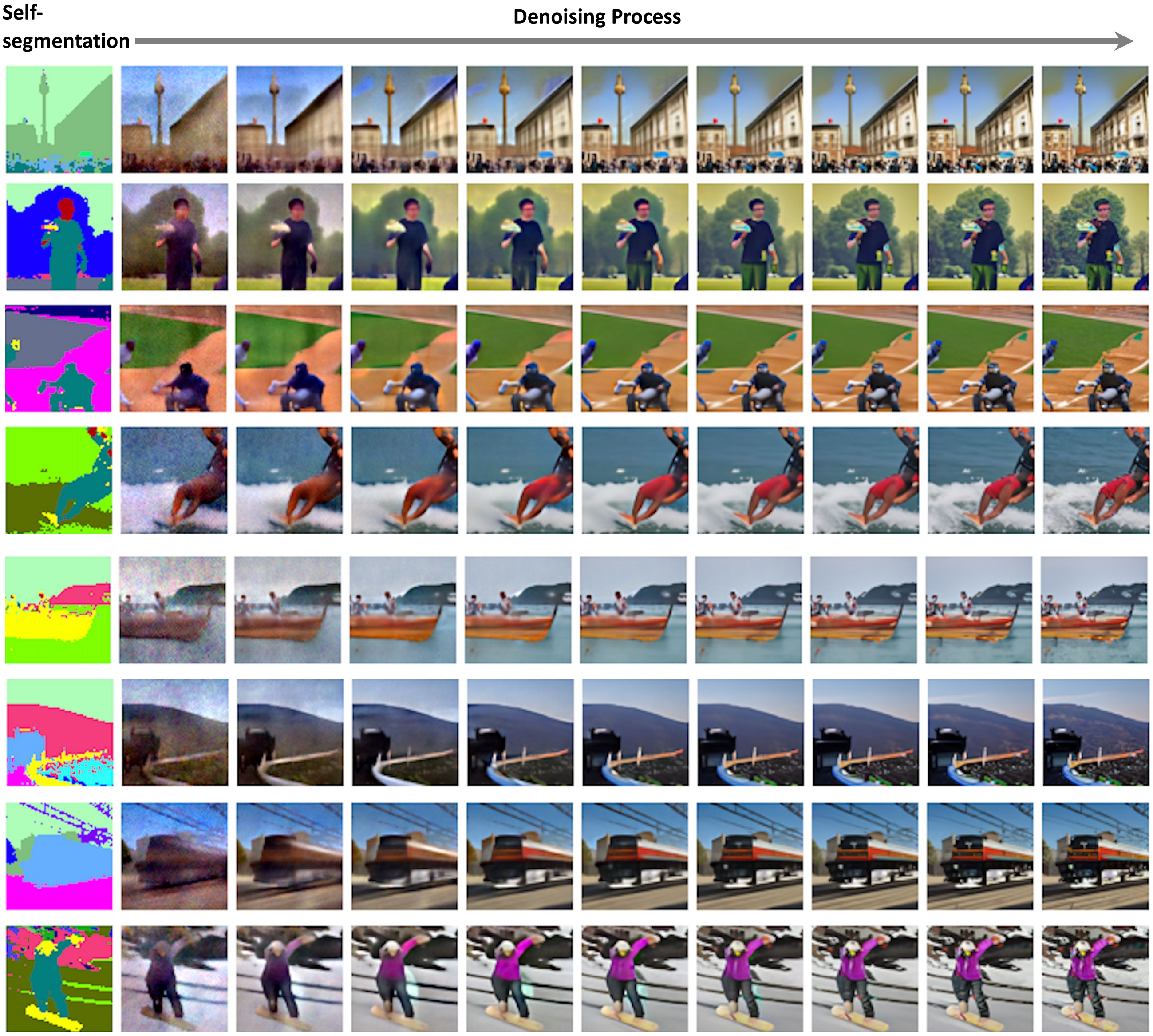


Figure 16. **Denoising process of self-segmented guidance samples (uncurated) from COCO-Stuff.** The first column is the self-segmented guidance mask from STEGO [22], The remaining columns are from the noisiest period to the less noisy period. Best viewed in color.

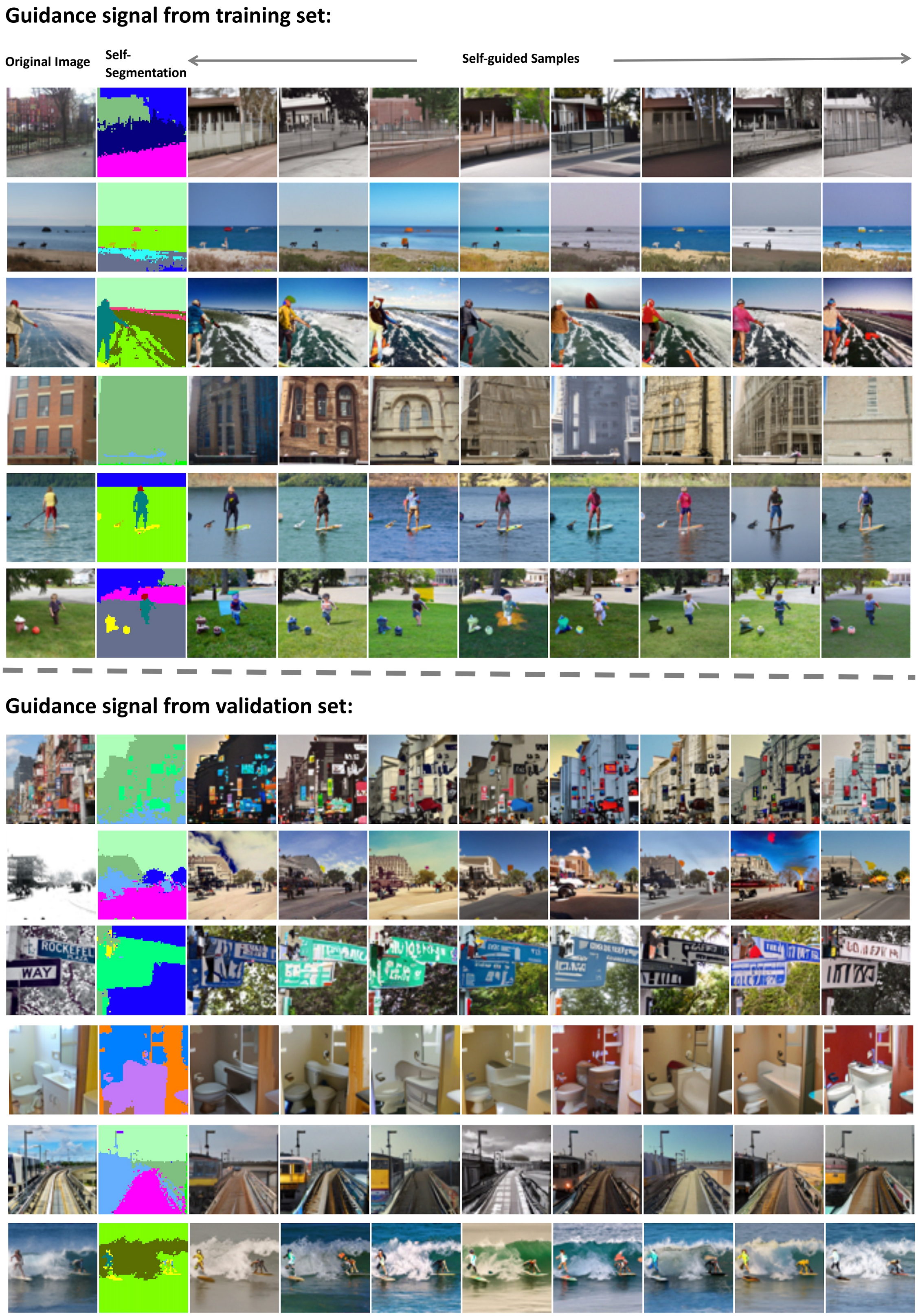

Guidance signal from training set:
Original Image
Self-Segmentation
Self-guided Samples
Guidance signal from validation set:
ROCKEFEL
WAY

Figure 17. **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 [22], The remaining columns are the random samples conditioning on the same self-segmented mask. Best viewed in color.

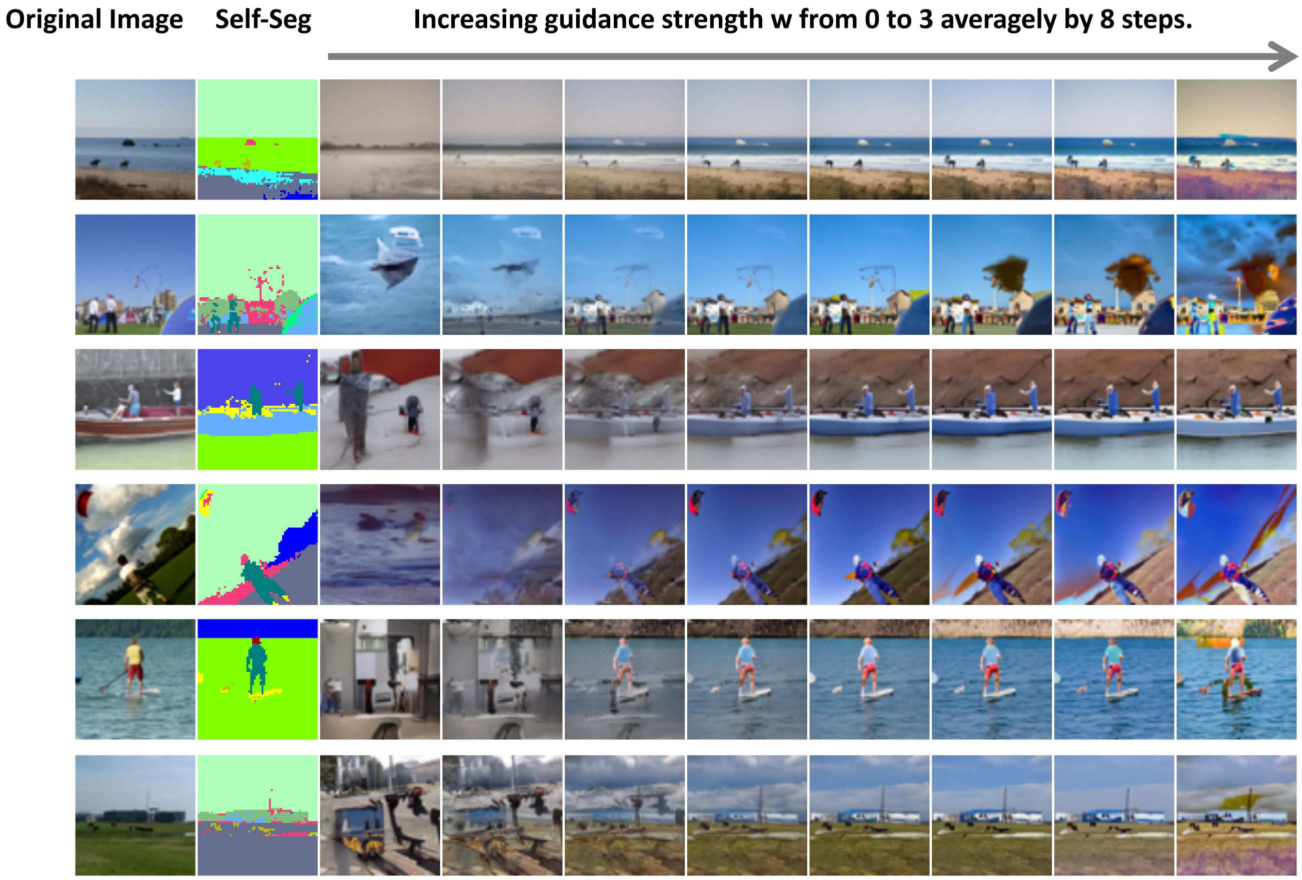

Original Image
Self-Seg
Increasing guidance strength w from 0 to 3 averagely by 8 steps.

Figure 18. **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 [22]. The remaining columns are the visualization when we averagely increase guidance strength w from 0 to 3 by 8 steps. Best viewed in color.

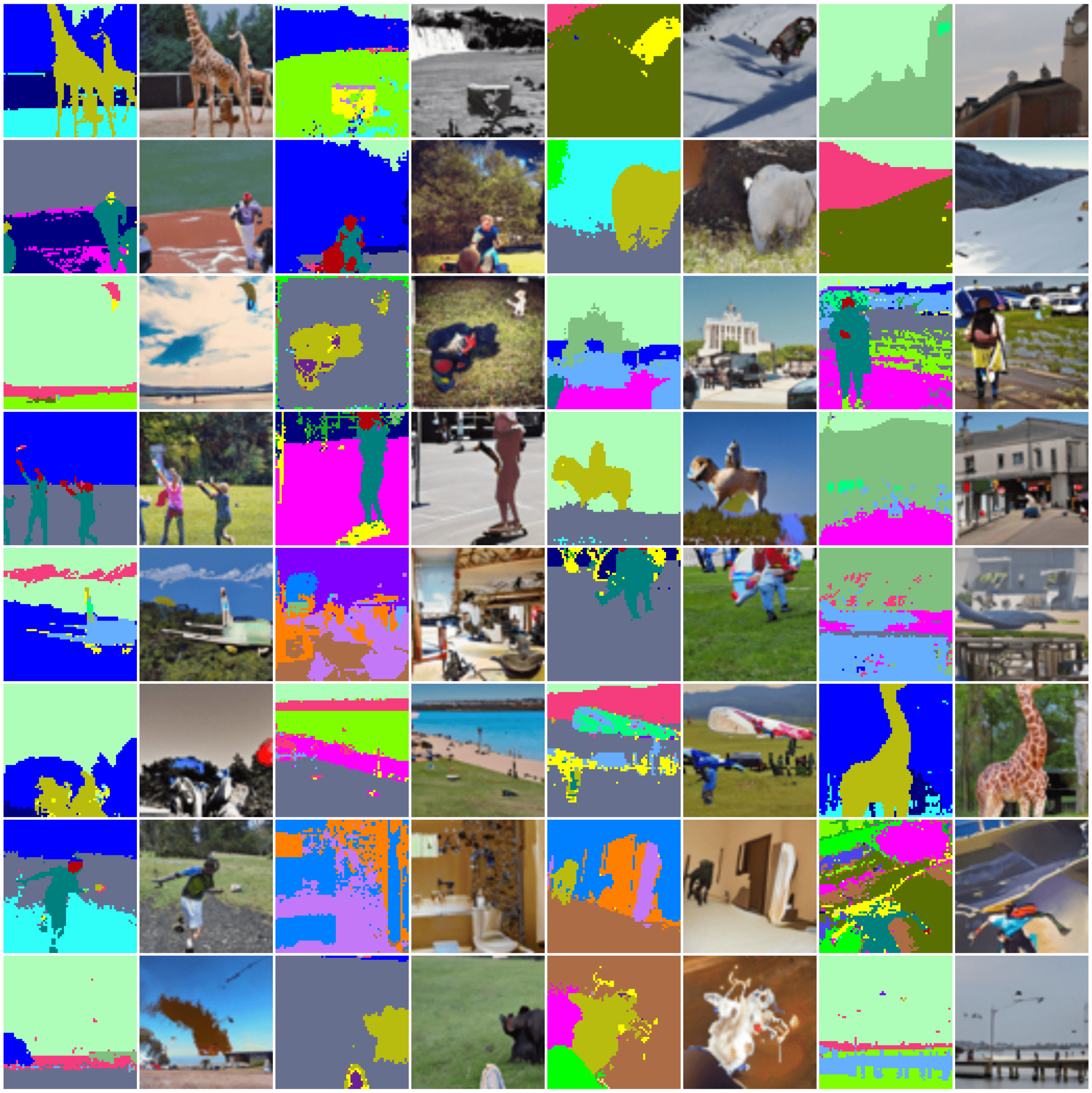


Figure 19. **Self-segmented guidance samples (uncurated)** from COCO-Stuff companies with segmentation mask from `STEGO` [22]. The color map is shared among the overall dataset. Best viewed in color.

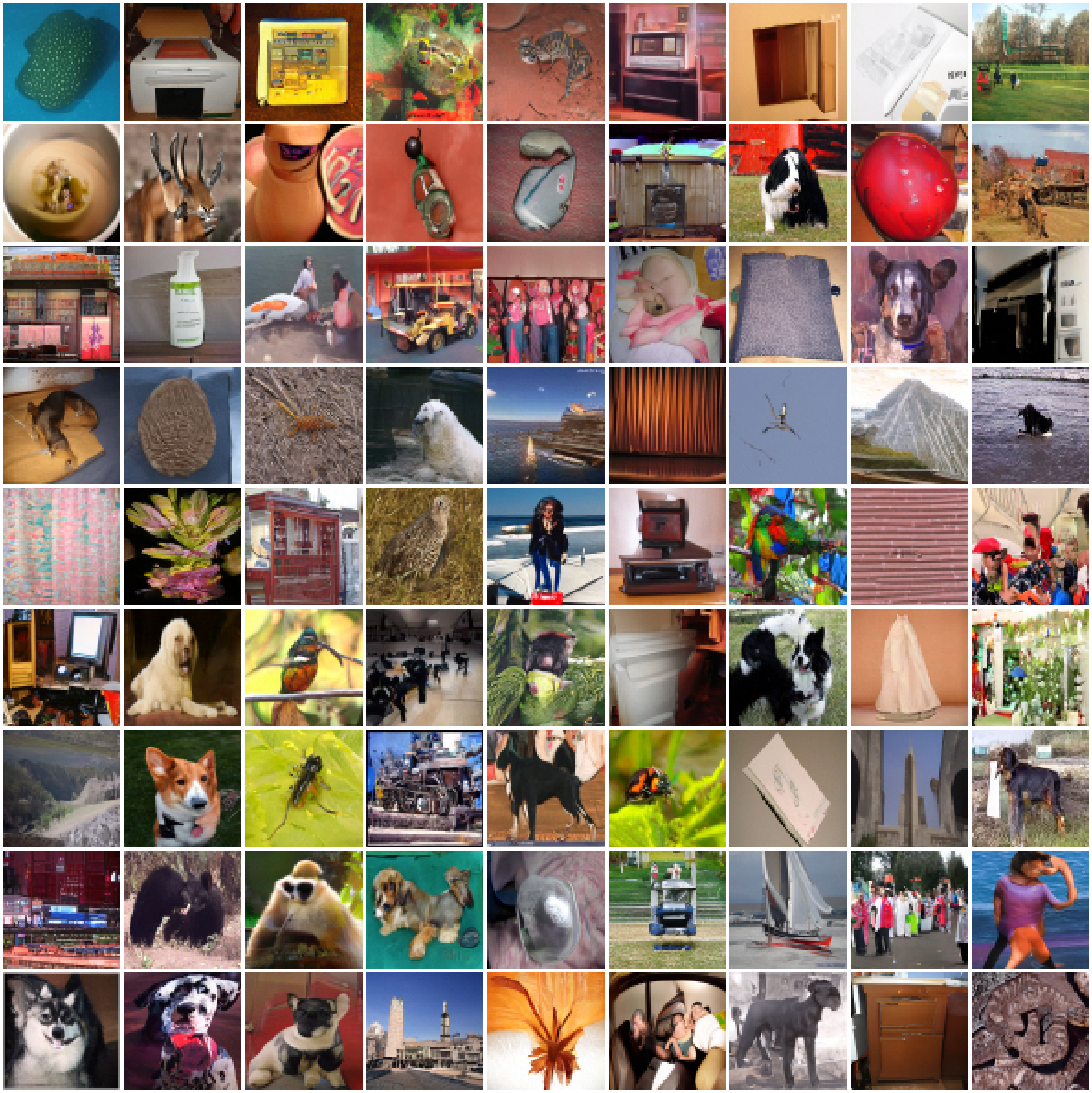


Figure 20. **Self-labeled guidance samples (uncurated) from ImageNet64.** Best viewed in color.

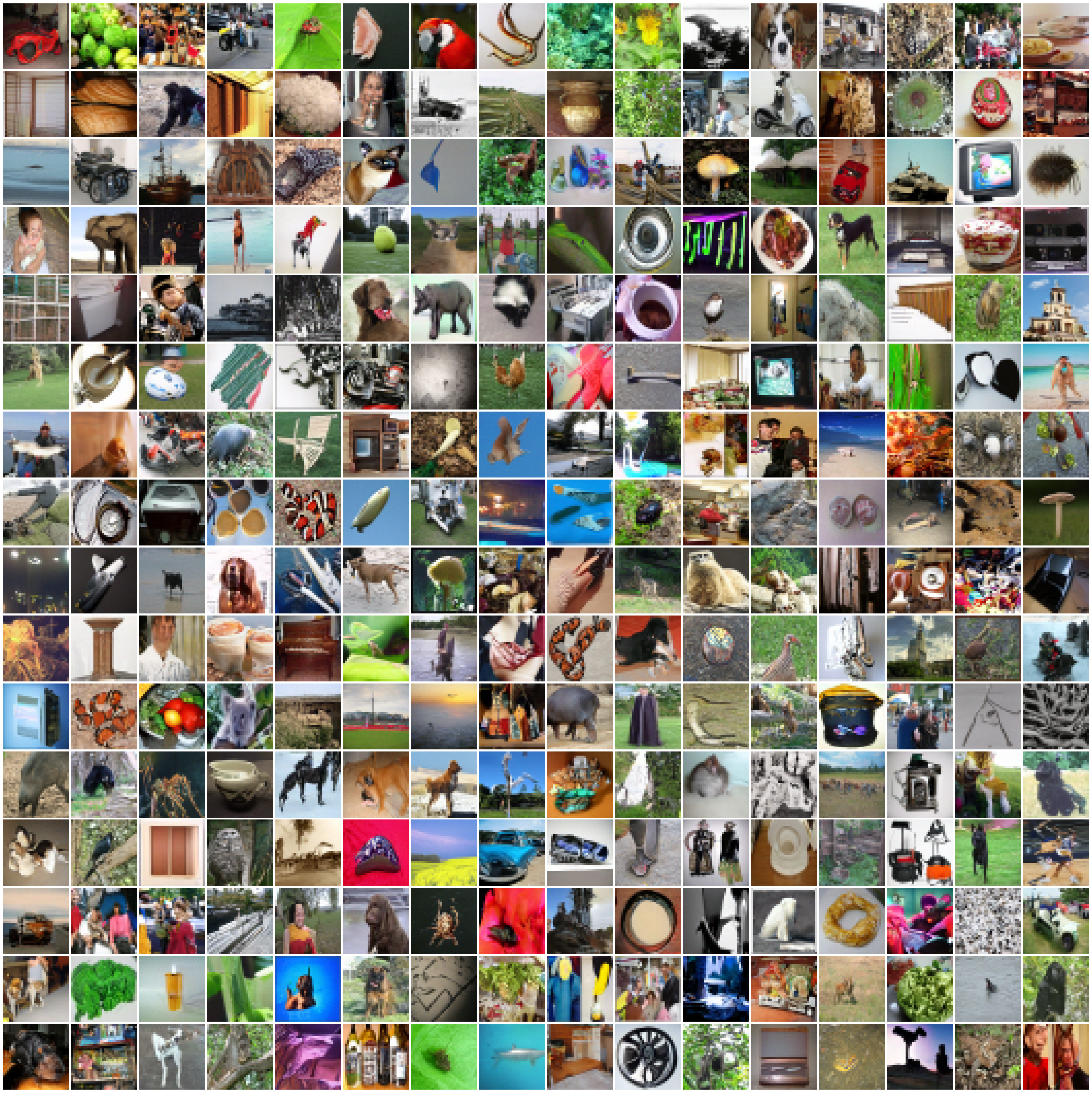
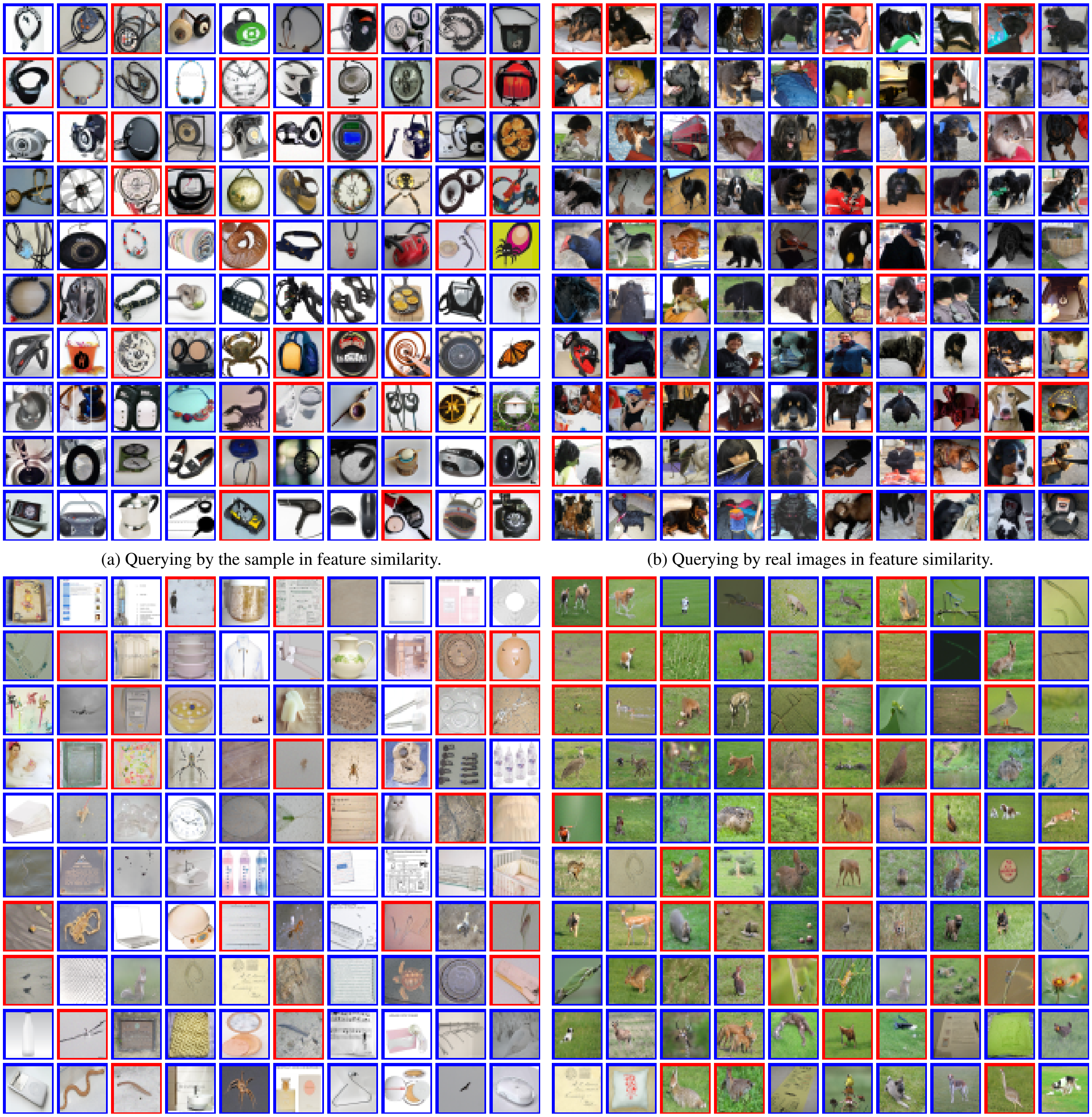


Figure 21. **Self-labeled guidance samples (uncurated) from ImageNet32.** Best viewed in color.



(a) Querying by the sample in feature similarity.

(b) Querying by real images in feature similarity.

(c) Querying by the sample in pixel similarity.

(d) Querying by real images in pixel similarity.

Figure 22. k**-NN query result visualization.** Blue means samples, red means real images. Images are ordered from left to right, top to down, by SimCLR [13] feature similarity or pixel similarity. Sampled images are sampled by DDIM [59] with 250 steps. Guidance strength w is 2. Firstly, we construct a gallery that is composed of an equivalent number of sampled and real images, then we ablate two experiments by querying using sampled images 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.

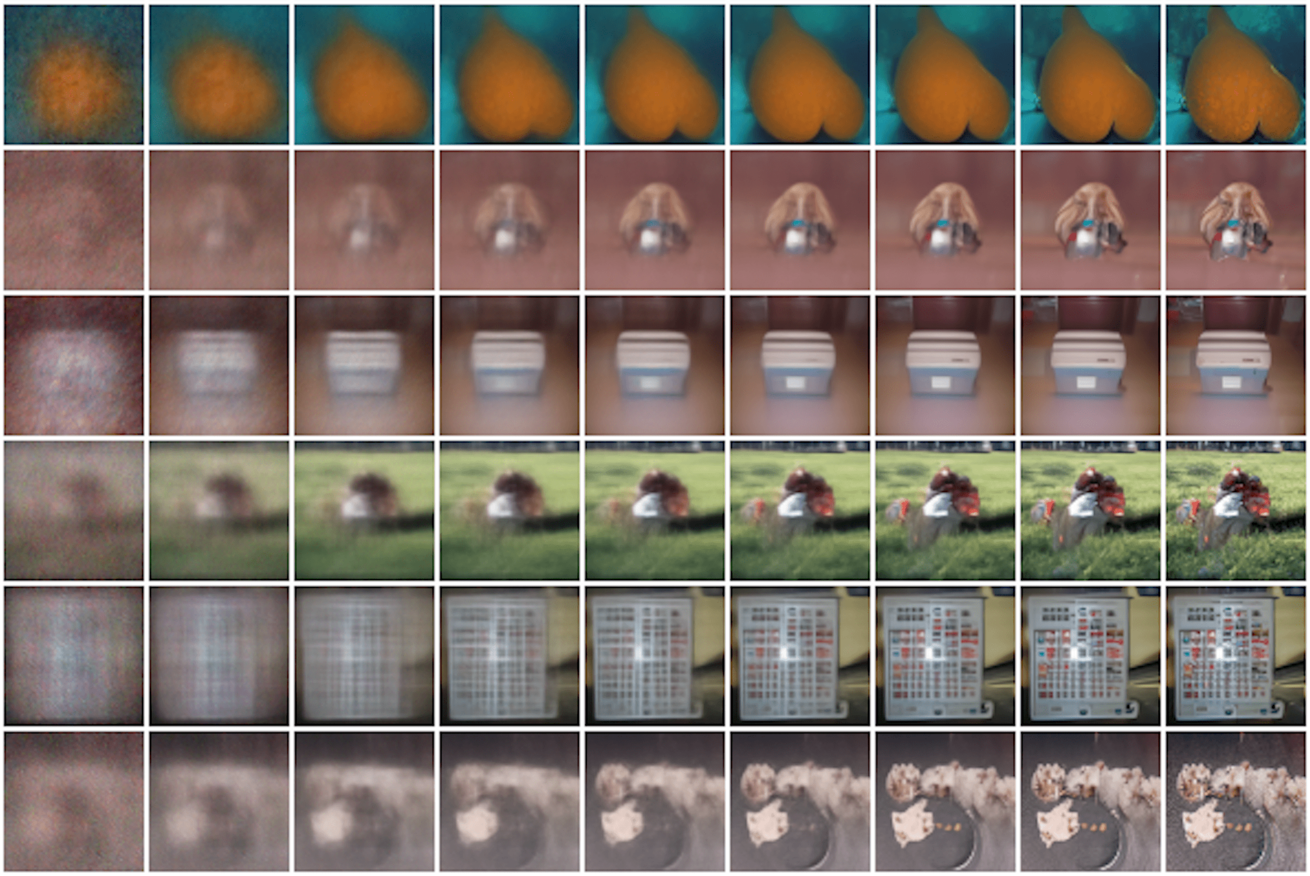


Figure 23. **Denoising process for ImageNet64.**

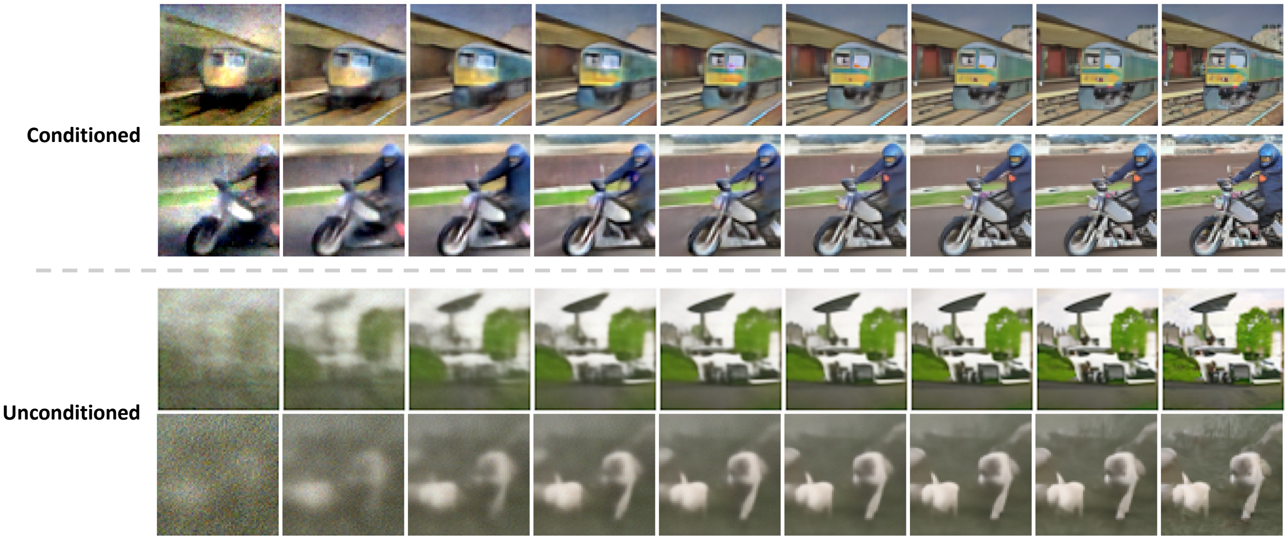


Figure 24. **Denoising process for Pascal VOC.** The first two rows are sampled from guidance strength $w = 2$ using our self-segmented guidance, the last two rows are sampled from guidance strength $w = 0$. By conditioning on our self-segmented guidance, the denoising process becomes easier and faster, this efficient denoising aligns with the observation from [47].

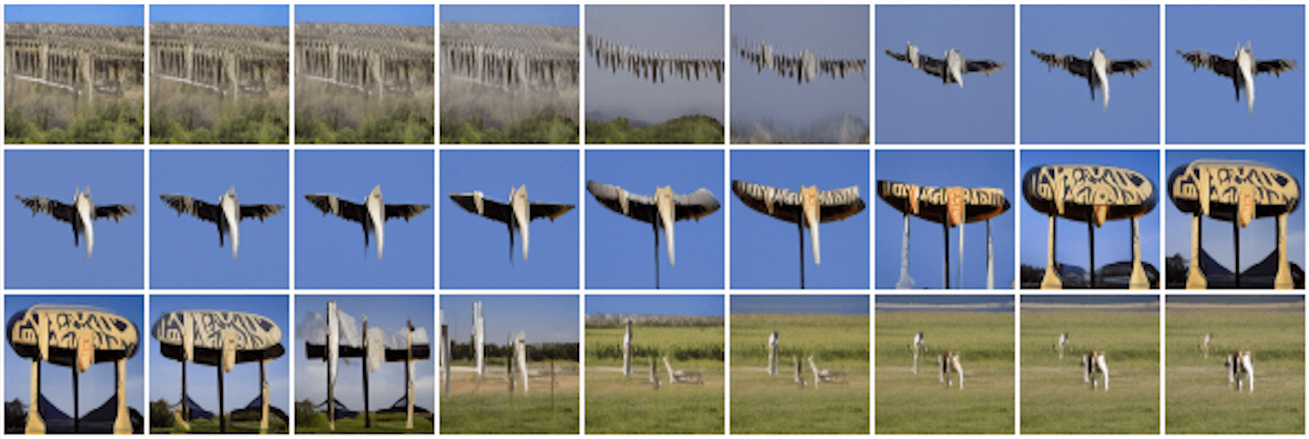
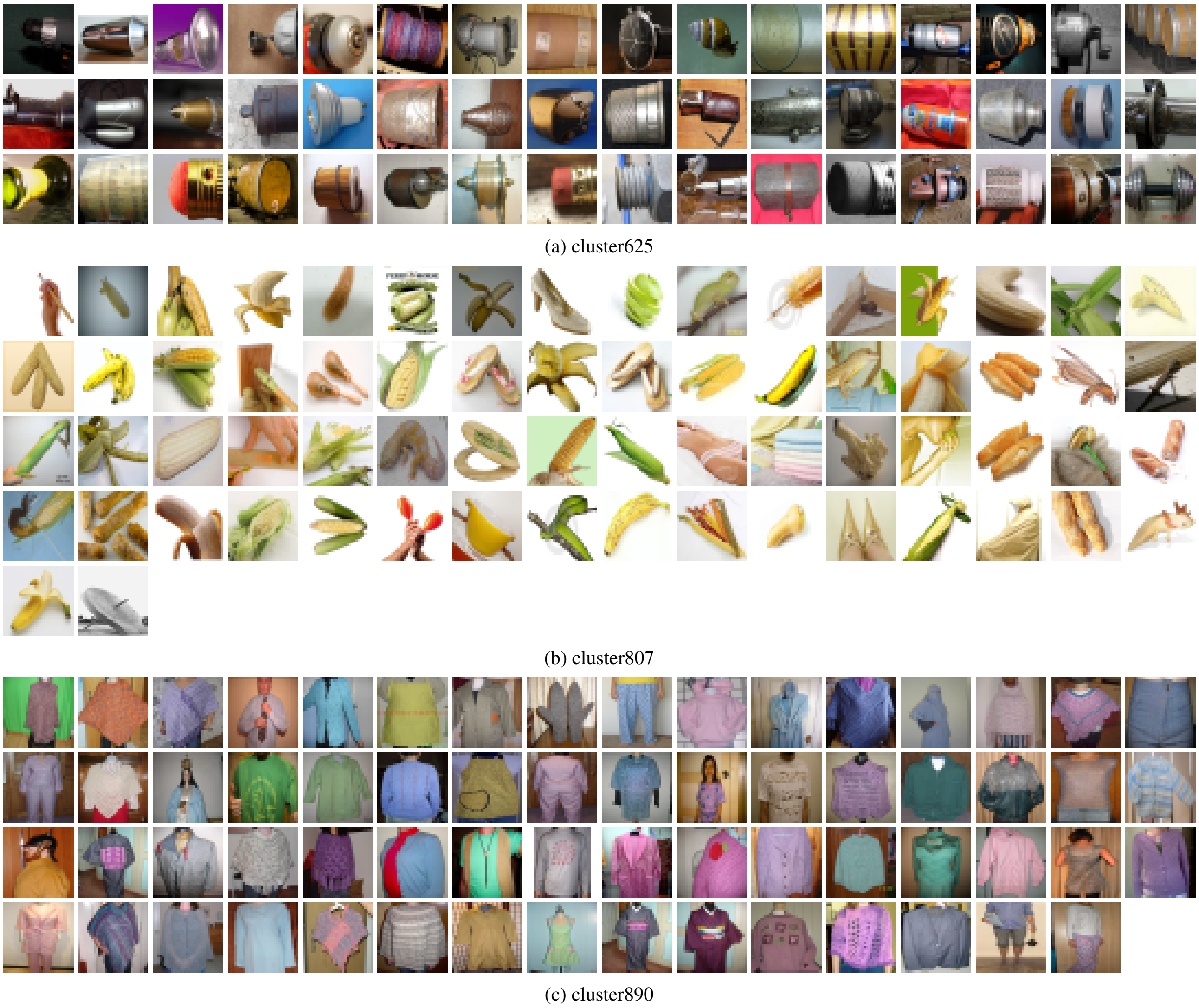


Figure 25. **Sphere interpolation between two random self-labeled guidance signals on ImageNet64.** The sphere interpolation follows the DDIM [59]. Best viewed in color.

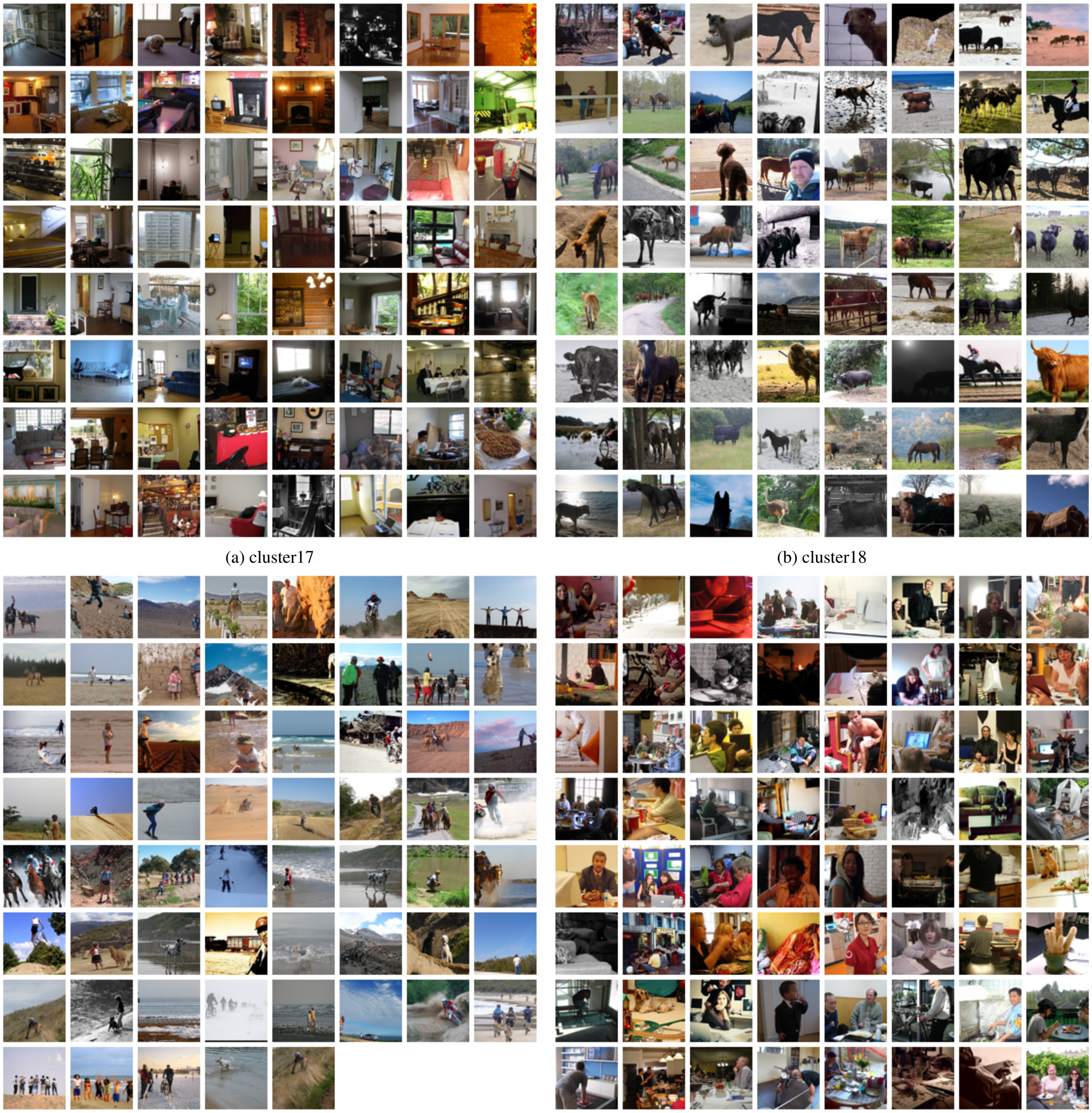


(a) cluster625

(b) cluster807

(c) cluster890

Figure 26. **Cluster visualization of real images in ImageNet32 after k-means.**



(a) cluster17

(b) cluster18

(c) cluster45

(d) cluster50

Figure 27. **Cluster visualization of real images in Pascal VOC after k-means.** Best viewed by zooming in.

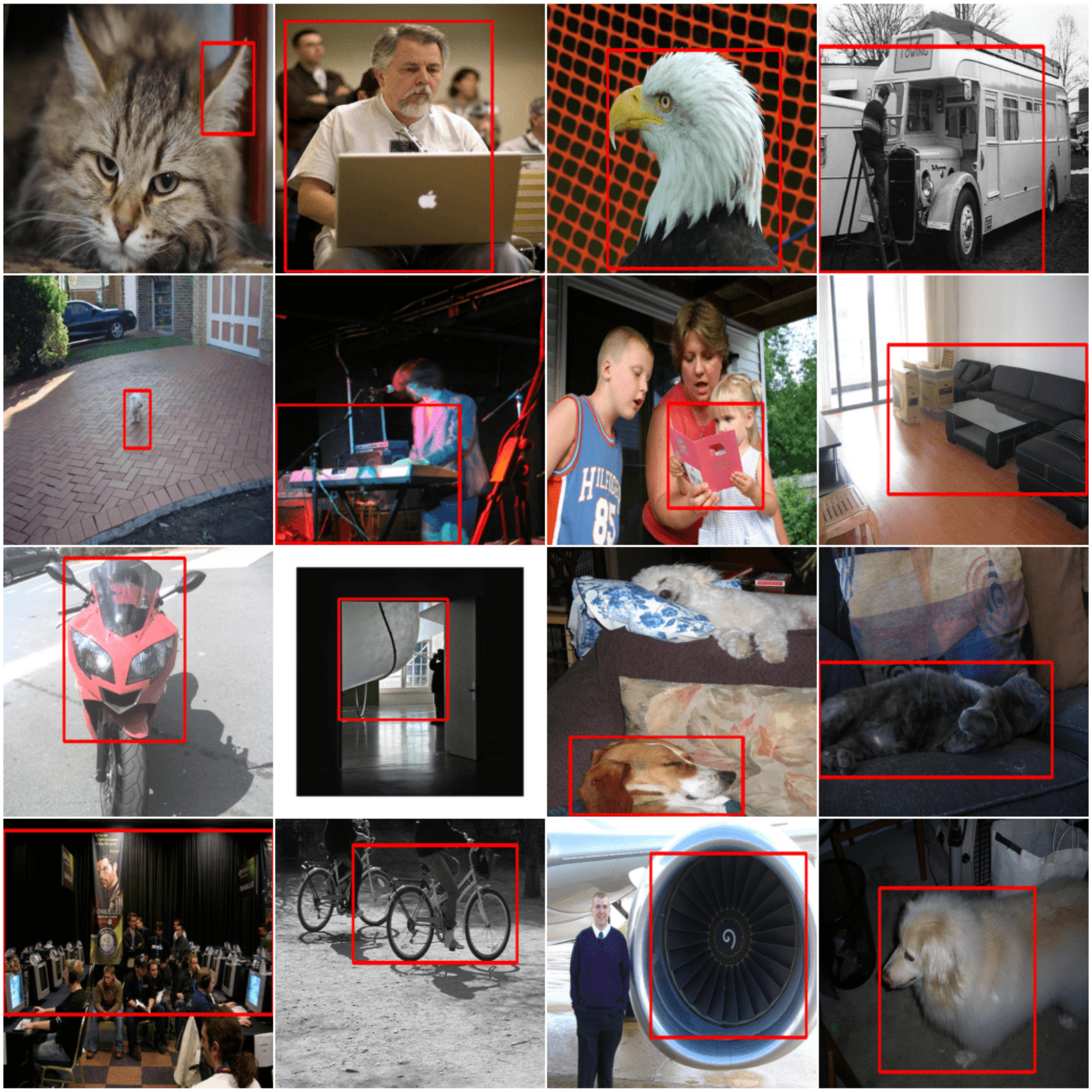


Figure 28. **Bounding box result from LOST on Pascal VOC**. As `LOST` [57] is an unsupervised-learning method, some flaws in the generated box are expected. Images are resized squarely for better visualization.

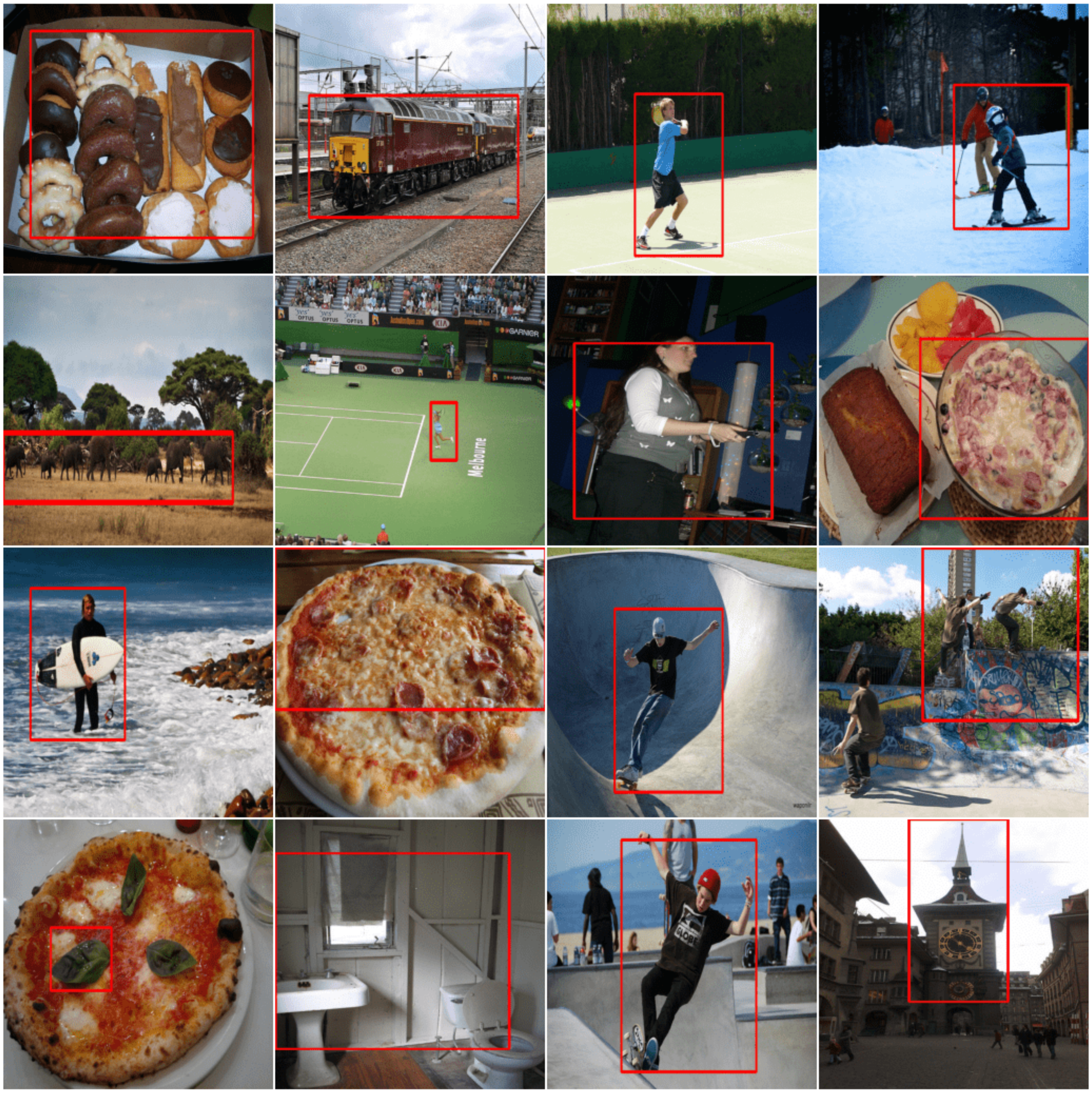


Figure 29. **Bounding box result from LOST [57] on COCO_20K.** Images are resized squarely for better visualization.

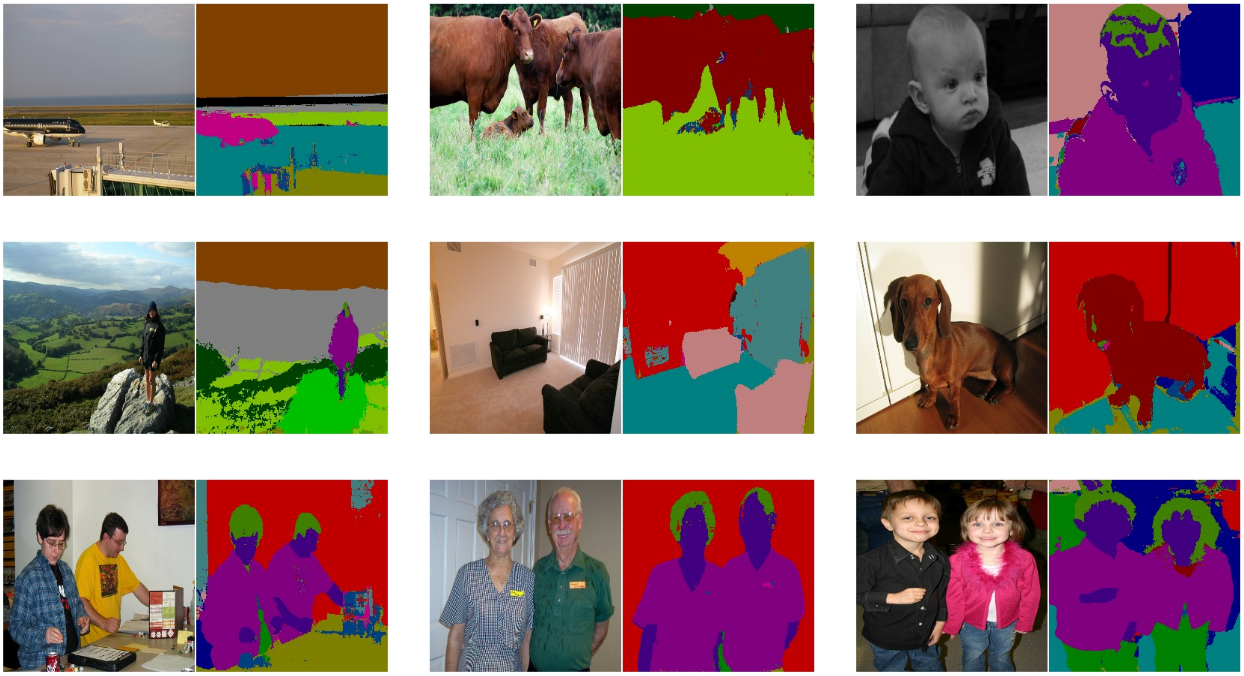


Figure 30. **Segmentation mask result from STEGO on Pascal VOC dataset**. Cluster number k is 21. Images are resized squarely for better visualization. The color map is shared among the overall dataset. Best viewed in color.