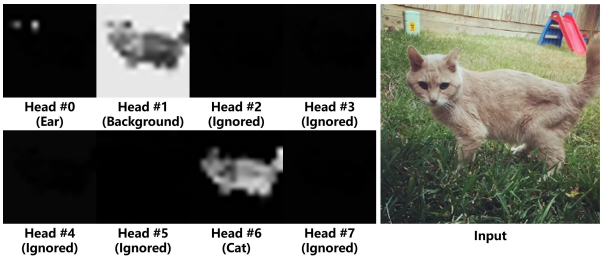

Figure 7. Head-wise auto-aggregation weights in a certain layer of SD v1.5.

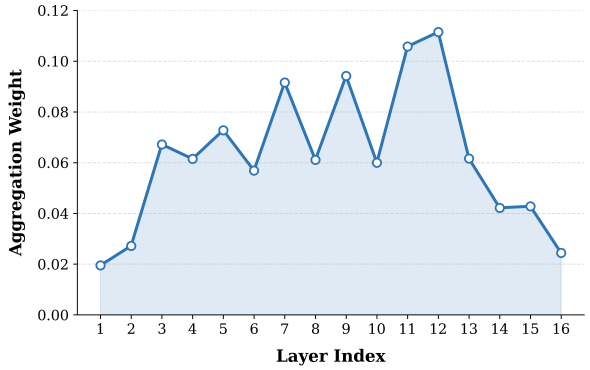

Figure 8. Layer-wise auto-aggregation weights of SD v1.5, using the cat input as shown in Figure 7.

A. Reproduction Details

In Table 1, we show the reproduced results of FTTM [39] instead of its originally reported results. This is because, in the original paper, a different setting is adopted. To be specific, Xiao *et al*. [39] used the predictions given by a training-free diffusion segmentor as image-segmentation pairs to train another supervised neural network to yield the final result. In our setting, however, the predictions given by training-free diffusion segmentors are directly adopted as the final result. As a consequence, we choose to reproduce the FTTM method in our setting.

B. Visualization of Auto Weights

We show example head-wise and layer-wise weights derived with auto-aggregation in Figure 7 and Figure 8. Both examples are based on the SD v1.5 model. The head-wise weights derived with auto-aggregation are per-pixel weights, meaning that each pixel has its own head weights, allowing fine-grained aggregation. The layer-wise weights, in contrast, are shared by all pixels.

C. Complexity Analysis

The computation overhead introduced by all components of GoCA is nearly negligible. Specifically, head-wise aggregation is linear to the number of heads and very minor, layer-wise aggregation is linear to the number of layers, and per-pixel re-scaling uses constant time.

To further strengthen our claim of negligible overhead, we also actually track the running time in Table 4. From the results, we can observe that the overhead introduced by GoCA is indeed small.

Table 4. Running time comparison on VOC 2012 dataset with the SD v1.5 model.

Method	Running Time (s)
SD v1.5 Vanilla	753.53
SD v1.5 Baseline	750.09
SD v1.5	767.73

D. Reliance on External Object Detectors

Due to the use of per-token re-normalization, training-free diffusion segmentors need to rely on external object detectors. To be specific, even when an object does not exist in the given image and has very low scores, its corresponding scores will still be normalized to $[0, 1]$ with per-token re-normalization. As a result, it is likely that some pixels will be assigned to this class nonetheless. Therefore, it is usually considered necessary to first filter non-existent objects from the prompts to obtain favorable segmentation results [17, 36, 39]. We follow this practice in this study, but see this as a drawback to overcome in future work. Moreover, since we are using a different external object detector from previous studies, we conduct an additional prompt alignment experiment in Section E.4 to show fairness.

E. Additional Ablation

E.1. Similarity Metrics

We use dot-product similarity for both head- and layer-wise aggregation. In this section, we ablate this choice by comparing with other similarity metrics, with results in Table 5. For head-wise aggregation, we have tried l_2-norm and cosine similarity beyond dot-product similarity. For layer-wise aggregation, we have tried the MSE loss and continuous IoU between self-attention maps beyond dot-product similarity. We can observe that dot-product is not always the best choice across different models. However, it is relatively robust, yielding satisfying results in different cases. Therefore, we choose dot-product similarity in the method design.

E.2. Timesteps

In Table 6, we compare the performance on different timesteps. We can observe that GoCA is not very sensi-

Table 5. Ablation results to compare different similarity metrics for head- and layer-wise aggregation.

Model	Head Metric	Layer Metric	mIoU (Context)
SD v1.5	dot-product	dot-product	40.4
	l2-norm	dot-product	**42.0**
	cosine	dot-product	**42.0**
	dot-product	mse	33.9
	dot-product	iou-like	39.2
SD XL	dot-product	dot-product	**42.3**
	l2-norm	dot-product	40.3
	cosine	dot-product	40.3
	dot-product	mse	42.1
	dot-product	iou-like	41.9
PixArt-Sigma	dot-product	dot-product	43.2
	l2-norm	dot-product	43.2
	cosine	dot-product	41.2
	dot-product	mse	**44.1**
	dot-product	iou-like	43.4

tive to timesteps, and it is relatively easy to find the optimal timestep using grid search.

E.3. Dense Feature

In layer-wise auto aggregation, we propose to use a dense feature to calculate the pseudo self-attention. In Table 7, we compare the performance with different layers used to extract the dense features. In the table, we use the layer notation and reported feature quality as in [23]. We can observe that GoCA is not very sensitive to dense feature selection, and it is a generic principle to find a dense feature with relatively high quality.

E.4. Prompt Alignment

In our experiment, we use GPT-4o as the external object detector. This is because we find GoCA especially effective at segmenting background objects, but usually, background objects are not listed as the classes of common datasets, making it hard to harness this improvement. Additionally, our per-pixel rescaling step requires the comparison between multiple objects, but there are cases where only one class is annotated for an image, making GoCA not applicable. Hence, we utilize the open-vocabulary capability of an LLM to detect both in-vocabulary and out-of-vocabulary objects in the images. Afterward, instead of the vanilla background threshold approach usually adopted by previous methods, we combine it with the segmentation results of these background objects. To be specific, we set the background score of a pixel as the maximum value of the threshold and all background object scores at this pixel. This makes GoCA applicable to single-object samples and gives GoCA results a tighter contour.

Note that GPT-4o does not yield prompts so perfect that the improvement of GoCA totally comes from GPT-4o. For example, GPT-4o does not always follow the format instruction, potentially introducing errors into the prompt parsing process. Additionally, GPT-4o and the annotation standard of a specific dataset may name the same object differently or perceive the same scene differently, leading to degradation. In fact, we find that, though GPT-4o prompts can lead to improvement with GoCA, SOTA methods tend to degenerate with them. This is demonstrated by the results in Table 8 (excluding the final row), where we change the prompts of DiffSegmenter [36] to what we use for GoCA. In this table, *GPT-4o w/o background* indicates a setting where we discard the background objects detected by GPT-4o and only use the in-vocabulary ones. *GPT-4o w/ background* is where we adopt a similar strategy as with GoCA to set the background scores dynamically. For both settings, there is significant degradation of segmentation capability. This demonstrates that although GoCA indeed relies on GPT-4o to gain some improvement and become more widely applicable, the enhancement does not totally come from GPT-4o.

E.5. More Powerful Model with SOTA

In the main experiment, we do not apply *Baseline* to stronger diffusion models, but only SD v1.5. This is because manually tuning the layer-wise weights is not practical for more complex models. Nevertheless, experimenting with more powerful models under stronger settings may still be desirable. Hence, we propose a workaround: we switch the model in DiffSegmenter to SD XL in the bottom row of Table 8. We use a weight of 0.7 for level 0 attention maps and a weight of 0.3 for level 1 attention maps (level 1 has

Table 6. Ablation results to compare different timesteps for feature extraction.

Model	Timestep	IoU (Context)
SD v1.5	50	40.1
	100	**40.4**
	150	40.3
	200	39.7

Table 7. Ablation results to compare different dense feature layers for layer-wise auto aggregation.

Model	Dense Layer	Dense Feature Quality (mIoU)	Resolution	IoU (Context)
SD v1.5	up-level3-repeat1-vit-block0-self-q	31.64	64	39.4
	up-level2-repeat1-vit-block0-cross-q	53.58	32	**40.4**
	up-level2-repeat2-vit-block0-cross-q	46.24	32	**40.4**
	up-level1-repeat1-vit-block0-cross-q	49.04	16	40.3

Table 8. Ablation results where we modify the implementation of DiffSegmenter [36], a SOTA method, and experiment on Context dataset.

Method	Context
Original	**60.4**
GPT-4o w/o background	47.1
GPT-4o w/ background	41.3
SD XL	43.2

a higher resolution). We can see that there is a significant performance drop, showing that SOTA methods can not effectively work with more powerful diffusion models.

F. Significance Experiment for Generation

In Table 3, the gain of FID scores is consistent but not very large. To better support the claim that GoCA is beneficial for S-CFG as its integrated component, we select one setting ($CFG = 5.0$) to conduct the same experiment under two additional random seeds. With three sets of results in total, we report the mean and standard deviation values in Table 9. As observed, the gain of *S-CFG* over *CFG* and *Ours+S-CFG* over *S-CFG* is consistent across all seeds, showing the benefit of GoCA as an integration in other techniques where training-free diffusion segmentors are needed.

G. Exact Feature Extraction Specification

Stable Diffusion v1.5: Extracting features at $t = 100$. Using cross-attention query activation from 2nd ViT, 3rd upsampling resolution to calculate the pseudo global attention. Treating "<sos>" (start-of-sentence) as semantic special tokens.

Stable Diffusion v1.5 (*Baseline*): For this setting, we extract cross-attention maps from layer 5 to 12, which are then averaged uniformly.

Stable Diffusion XL: Extracting features at $t = 100$. Using cross-attention query activation from 1st basic block, 1st ViT, 2nd upsampling resolution to calculate the pseudo global attention. This model has two text encoders, but their outputs are concatenated along the channels dimension, so we can treat them as a single text encoder when extracting attention maps. This model is a bit special about semantic special tokens: it has a dedicated pooled embedding to serve as the global embedding, and all "<sos>" tokens have exactly zero scores. However, in this model, stop word tokens have much higher scores, somehow serving the role of semantic special tokens. Hence, our decision to exclude both semantic special tokens and stop word tokens can work with Stable Diffusion XL.

PixArt-Sigma: Extracting features at $t = 100$. Using cross-attention query activation from 14th basic block to calculate the pseudo global attention. Treating "<eos>" (end-of-sentence) as semantic special tokens.

Flux: Extracting features at $t = 150$. Using attention output activation from 28th basic block to calculate the pseudo global attention. Specially, in the MMDiT [18] architecture, there are both image-query-text-key/value attention sections and text-query-image-key/value attention sections. We only extract attention maps from where the image serves as the query and text serves as the key/value, to be aligned with U-Net and DiT settings. Additionally, this model has two text encoders, but only one of them produces the per-token embedding, so we can treat this model as having only one text encoder when extracting attention maps. Treating "<eos>" (end-of-sentence) as semantic special tokens.

Besides the model-specific implementation, we simply treat all tokens that are neither semantic special tokens nor

Table 9. Experimental results with three different seeds under $CFG = 5.0$ setting for FID and CLIP score evaluation.

Method	FID (1)	FID (2)	FID (3)	FID (AVG)	CLIP (1)	CLIP (2)	CLIP (3)	CLIP (AVG)
CFG	19.27	19.97	21.09	20.11±0.92	31.34	31.12	30.95	31.13±0.20
S-CFG	19.15	18.52	20.41	19.36±0.96	31.35	31.26	31.16	31.25±0.10
Ours+S-CFG	**18.82**	**18.01**	**20.32**	**19.05±1.17**	**31.42**	**31.28**	**31.16**	**31.28±0.13**

objects of interest (content word tokens) as stop word tokens. In this way, we have a generic way to define the token categories.

There are also other diffusion models where the outputs of multiple text encoders are concatenated along the sequence length dimension. We can extract corresponding tokens from all text encoders. For example, if "cat" is encoded by both text encoders, there will be two tokens for this word, and we simply average their scores. This is the same as how we deal with regular multi-word objects, such as "potted plants".

H. Additional Generation Results

We show additional SD v1.5 generation results in Figure 9. *Ours* can enhance the generation quality more than *Vanilla*, showing the effectiveness of GoCA.

I. Additional Segmentation Visualization

We provide three groups of segmentation visualization in three increasingly more complex scenes, in Figure 10, Figure 11, and Figure 12. Notably, we can observe a failure pattern from Figure 12: when an object is very small with respect to how many pixels it takes, the segmentation for this object tends to fail. This might be because the VAE compresses the original image, or because small objects are long-tail knowledge during training [19, 37, 38, 44].

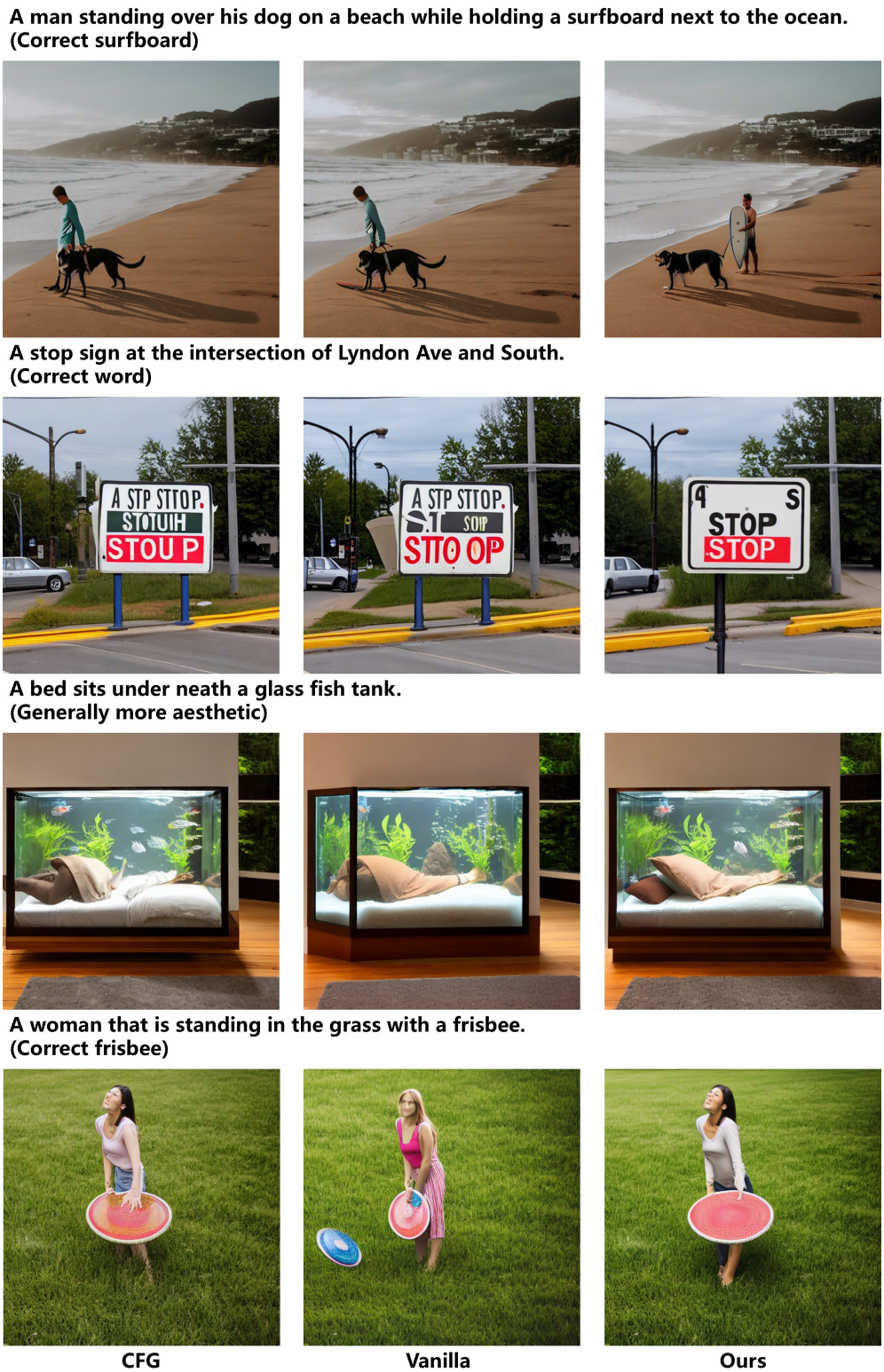

Figure 9. Generated images with SD v1.5 using $CFG = 7.5$ setting. Corresponding prompts and why *Ours* is considered better are annotated above each row of images.

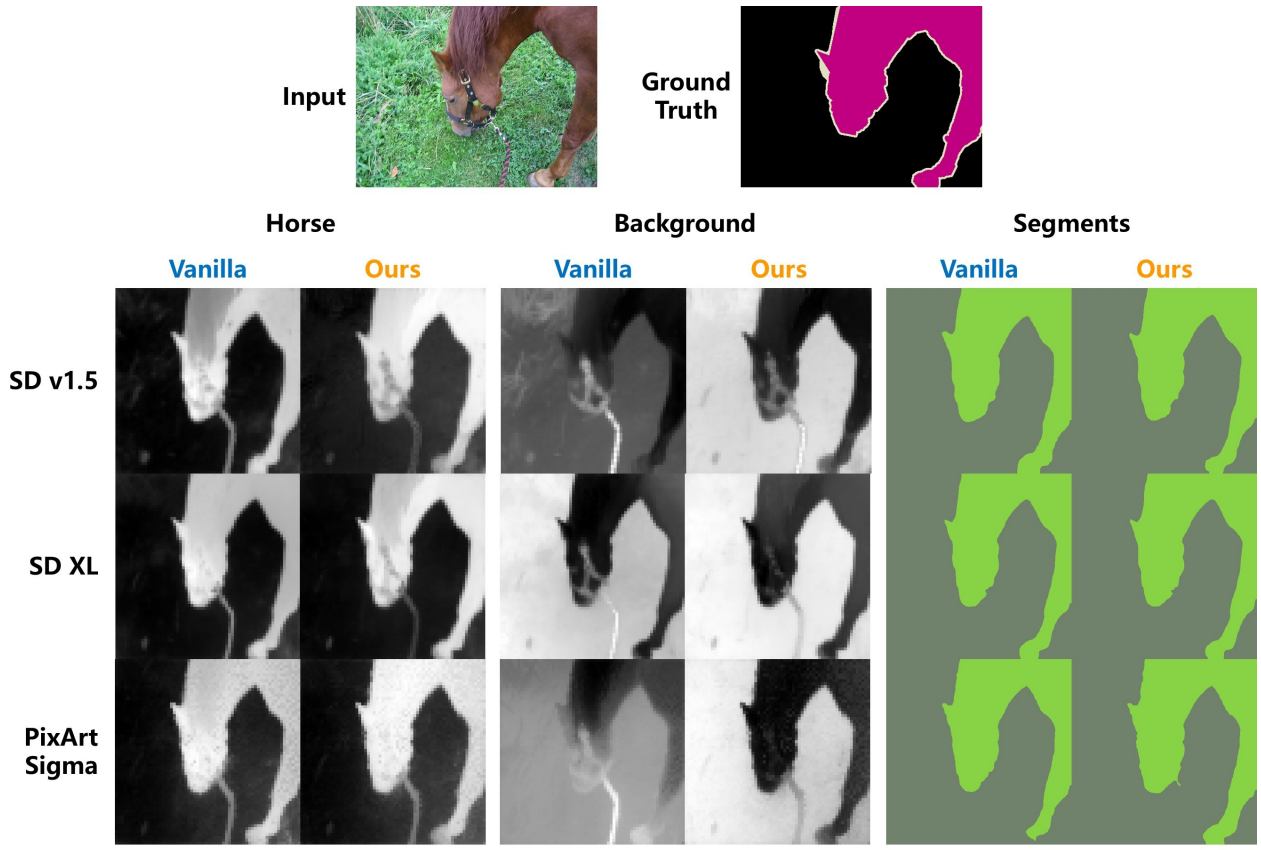

Figure 10. In a simple scene, though there are quality gaps in per-class maps, the final segmentation maps are almost identical.

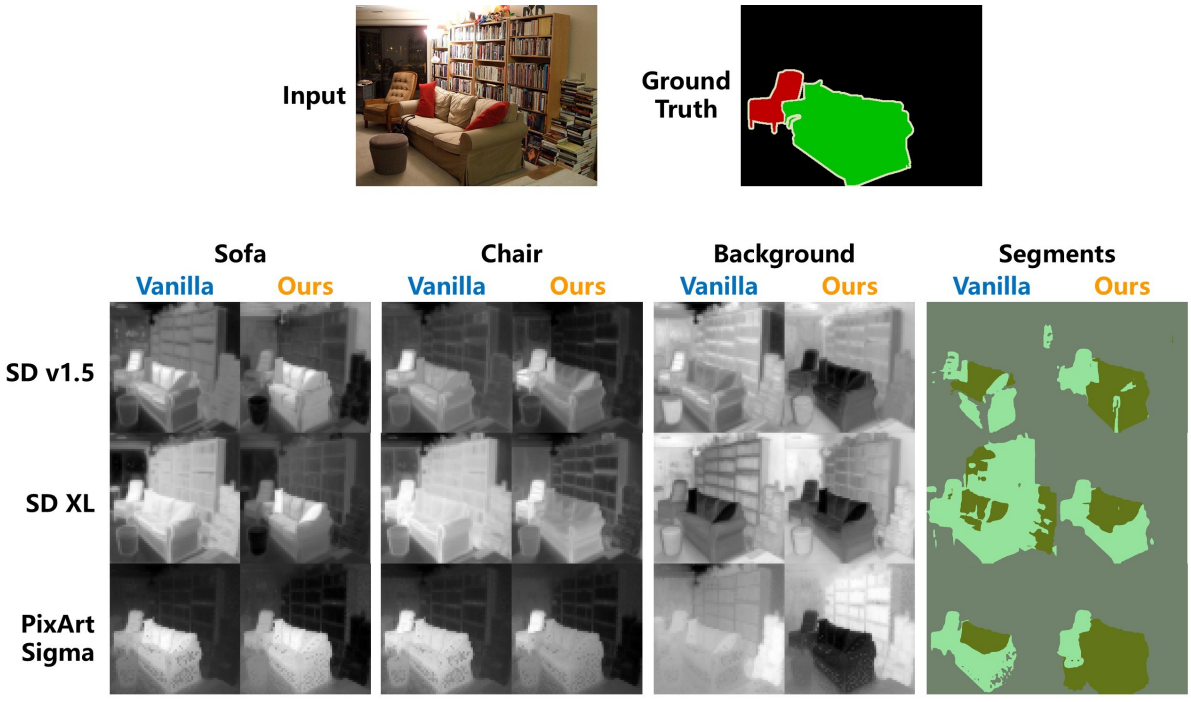

Figure 11. In a moderately complex scene, *Ours* gains significant quality improvement, especially in the background regions.

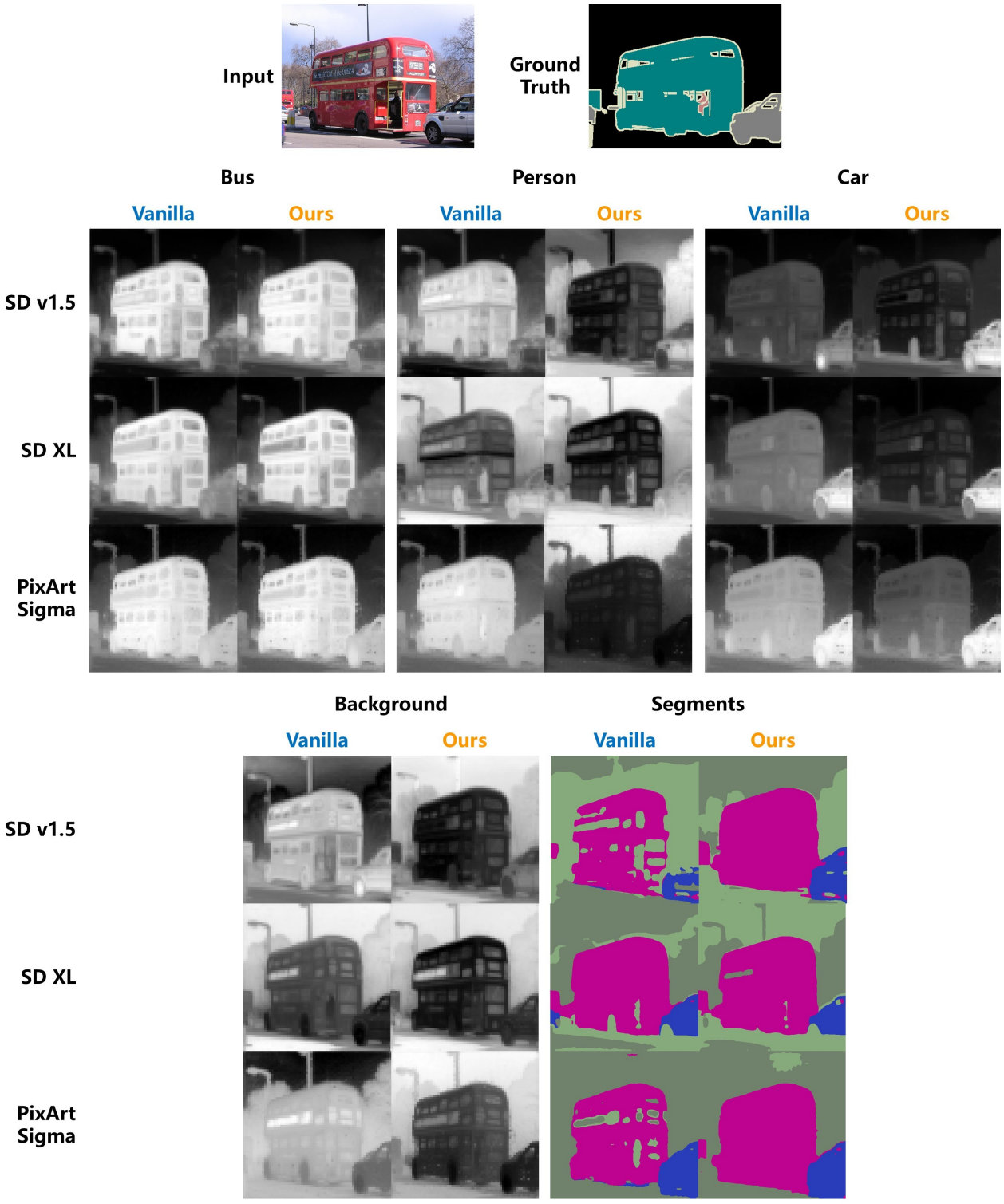

Figure 12. All models and both methods fail to capture the small person object in the complex scene, as latent diffusion models compress the original image with VAE. Nevertheless, we can still observe obvious quality enhancement in the segmentation of background regions, covering the sky, trees, and the road.