Text-image Alignment for Diffusion-based Perception Supplementary Materials

A. Cross-attention analysis

Qualitative image-to-image variation analysis. We present a qualitative and quantitative analysis of the effect of off-target class names added to the prompt. In Fig. S1, we use the stable diffusion image to image (img2img) variation pipeline (with the original Stable Diffusion 1.5 weights) to qualitatively analyze the effects of prompts with off-target classes. The img2img variation pipeline encodes a real image into a latent representation, adds a user-specified amount of noise to the latent representation, and de-noises it (according to a user-specified prompt) to generate a variation on the original image. The amount of noise added is dictated by a strength ratio indicating how much variation should occur. A higher ratio results in more added noise and more denoising steps, allowing a relatively higher impact of the new text prompt on the image. We find that $\mathcal{C}_{\text{ClassNames}}$ (see caption for details) results in variations that incorporate the off-target classes. This effect is most clear looking across the panels left to right in which objects belonging to off-target classes (an airplane and a train) become more prominent. These qualitative results imply that this prompt modifies the latent representation to incorporate information about off-target classes, potentially making the downstream task more difficult. In contrast, using the BLIP prompt changes the image, but the semantics (position of objects, classes present) of the image variation are significantly closer to the original. These results suggest a mechanism for how off-target classes may impact our vision models. We quantitatively measure this effect using a fully trained Oracle model in the following section.

Copy-Paste Experiment. An interesting property in Fig. 4 is that the word bottle has strong cross-attention over the neck of the bird. We hypothesize that diffusion models seek to find the nearest match for each token since they are trained to generate images that correspond to the prompt. We test this hypothesis on a base image of a dog and a bird. We first visualize the cross-attention maps for a set of object labels. We find that the words bottle, cat, and horse have a strong cross-attention to the bird, dog, and dog, respectively. We paste a bottle, cat, and horse into the base image to see if the diffusion model will localize the "correct" objects if they are present. In Fig. S2, we show that the cross-attention maps prefer to localize the "correct" object, suggesting our hypothesis is correct.

Averaged EOS Tokens: Averaging vs. EOS? Averaged EOS Tokens create diffuse attention maps that empirically harm performance. What is the actual cause of the decrease in performance? Is it averaging, or is it the usage of many EOS tokens? We replace the averaged EOS tokens with single prompt EOS tokens and find that the attention maps are still diffuse. This indicates that the usage of EOS tokens is the primary cause of the diffuse attention maps and not the averaging.

Quantitative effect of $\mathcal{C}_{\text{ClassNames}}$ on Oracle model. To quantify the impact of the off-target classes on the downstream vision task, we measure the averaged pixel-wise scores (normalized via Softmax) per class when passing the $\mathcal{C}_{\text{ClassNames}}$ to the Oracle segmentation model for Pascal VOC 2012 (Fig. S4). We compare this to the original oracle prompt. We find that including the off-target prompts significantly increases the probability of a pixel being misclassified as one of the semantically nearby off-target classes. For example, if the original image contains a cow, including the words dog and sheep, it significantly raises the probability of misclassifying the pixels belonging to the cow as pixels belonging to a dog or a sheep. These results indicate that the task-specific head picks up the effect of off-target classes and is incorporated into the output.

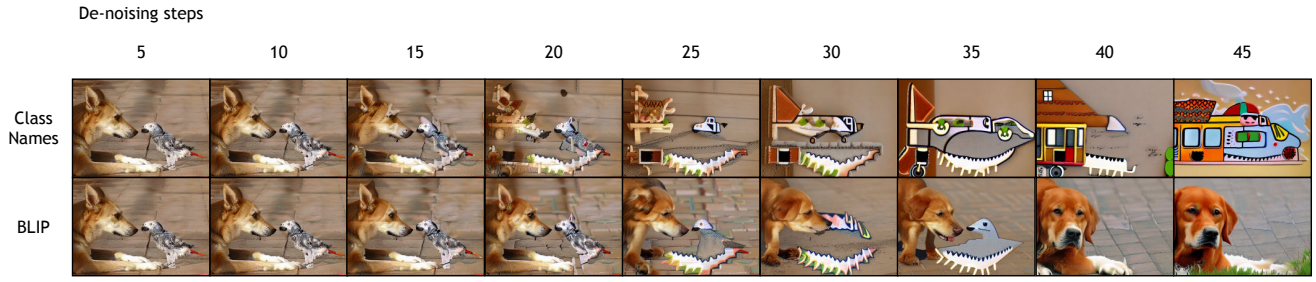

Figure S1. **Qualitative image-to-image variation.** An untrained stable diffusion model is passed an image to perform image-to-image variation. The number of denoising steps conducted increases from left to right (5 to 45 out of a total of 50). On the top row, we pass all the class names in Pascal VOC 2012: *"background airplane bicycle bird boat bottle bus car cat chair cow dining table dog horse motorcycle person potted plant sheep sofa train television"*. In the bottom row we pass the BLIP caption *"a bird and a dog"*.

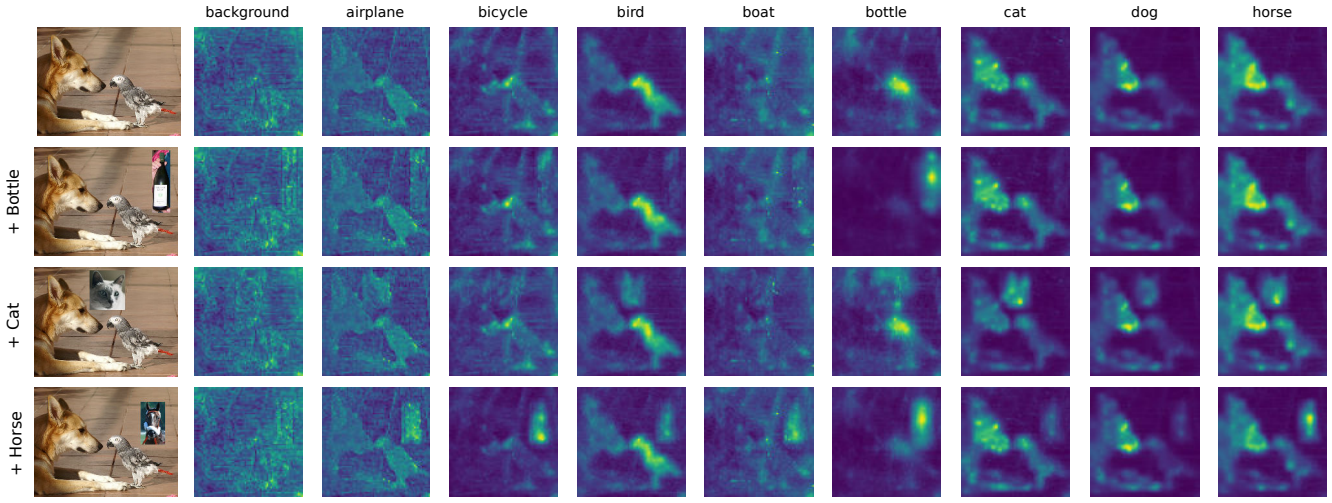

Figure S2. **Copy-Paste Experiment.** A bottle, a cat, and a horse from different images are copied and pasted into our base image to see how the cross-attention maps change. The label on the left describes the category of the item that has been pasted into the image. The labels above each map describe the cross-attention map corresponding to the token for that label.

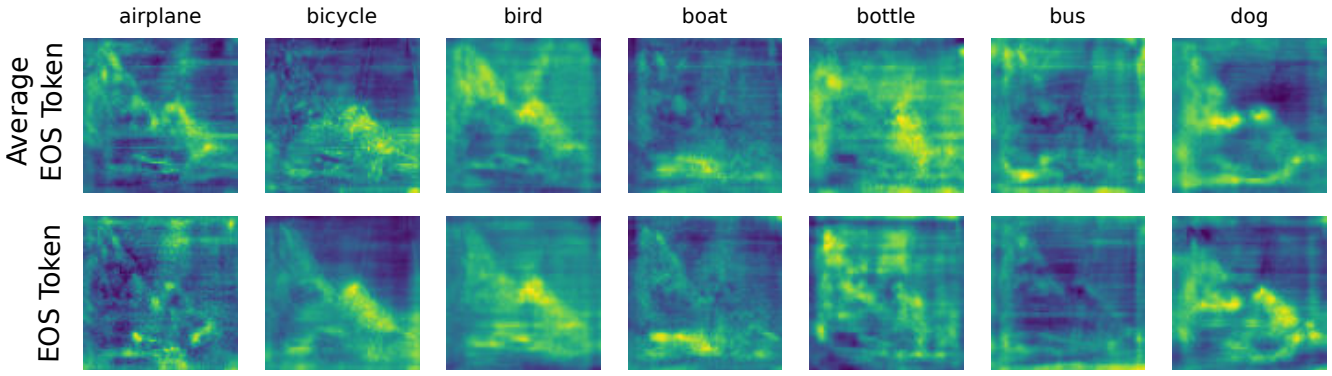

Figure S3. **Averaging vs. EOS.** In [53], for each class name, the EOS token from 80 prompts (containing the class name) was averaged together. The averaged EOS tokens for each class were concatenated together and passed to the diffusion model as text input. We explore if averaging drives the diffuse nature of the cross-attention maps. We replace the 80 prompt templates with a single prompt template: "a photo of a {class name}" and visualize the cross-attention maps. In the top row, we show the averaged template EOS tokens. In the bottom row, we show the single template EOS tokens.

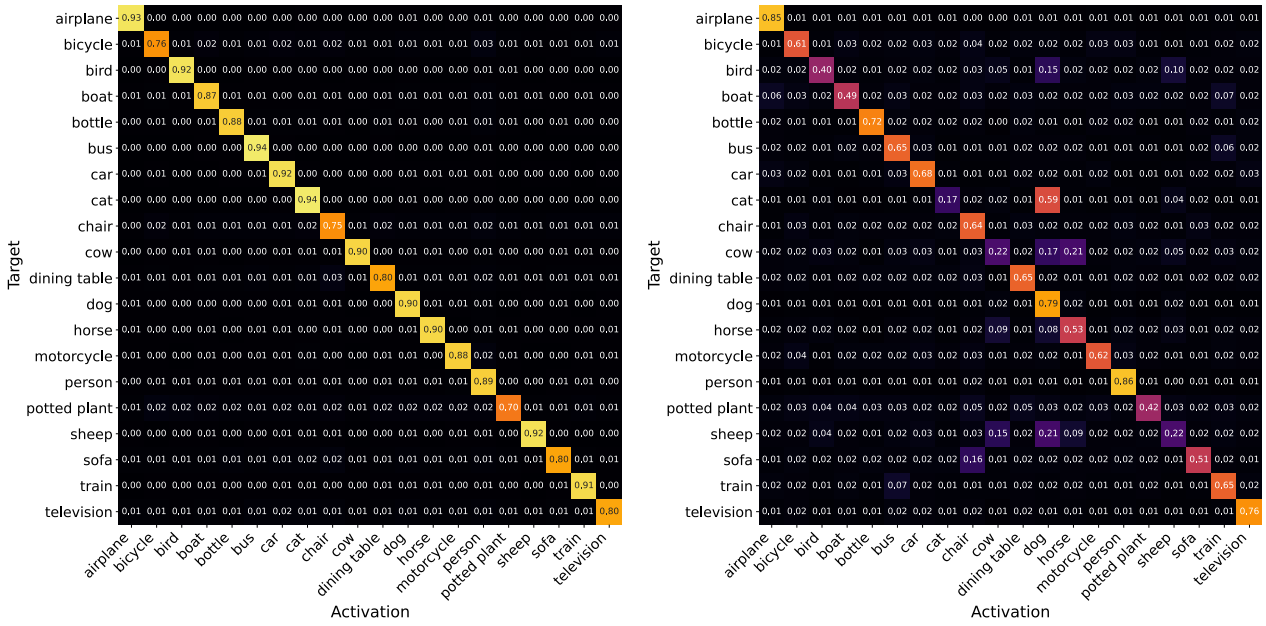

Figure S4. **Impact of off-target classes on semantic segmentation performance.** The matrices show normalized scores averaged over pixels on Pascal VOC 2012 for an oracle-trained model when receiving either present class names (left) or all class names (right).

B. Additional ADE20K Results

Method	4K Iters		8K Iters	
	mIoU^{ss}	mIoU^{ms}	mIoU^{ss}	mIoU^{ms}
VPD (null text)	41.5	-	46.9	-
VPD_{A32} [53]	43.1	44.2	48.7	49.5
VPD(R)	42.6	43.6	49.2	50.4
VPD(LS)	45.0	45.8	50.5	51.1
TADP-20 (Ours)	**50.2**	**50.9**	**52.8**	**54.1**
TADP(TA)-20 (Ours)	49.9	50.7	52.7	53.4

Table S1. **Semantic segmentation fast schedule on ADE20K.** Our method has a large advantage over prior work on the fast schedule with significantly better performance in both the single-scale and multi-scale evaluations for 4k and 8k iterations.

		Recall		
		0.50	**0.75**	**1.00**
Precision	**0.50**	49.53	52.00	55.22
	0.75	49.17	51.46	58.62
	1.00	50.20	54.82	63.29

Table S2. **ADE20K - Oracle Precision-Recall Ablations** We modify the oracle captions by randomly adding or removing classes such that the precision and recall are 0.50, 0.75, or 1.00. We train models on ADE20K on a fast schedule (4K) using these captions. The 4k iteration oracle equivalent is highlighted in blue.

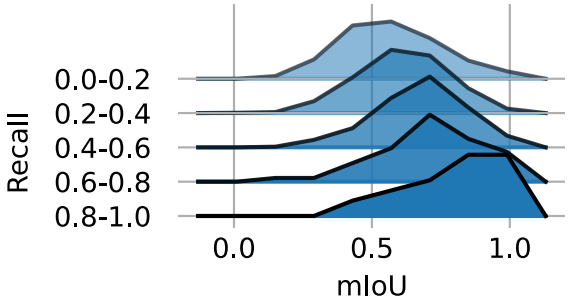

Figure S5. **Recall analysis.** ADE20k mIOU per image with respect to the recall of classes present in the caption. We embedded each word in our caption with CLIP's text encoder. We considered a cosine similarity of ≥ 0.9 with the embedded class name as a match. Linear regression analysis shows positive correlations between recall and mIoU ($r = 0.28$).

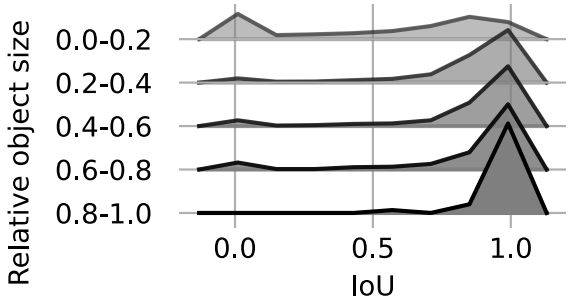

Figure S6. **Object size analysis.** ADE20k IOU per object image with respect to the relative object size (pixels divided by total pixels). Linear regression analysis shows positive correlations between relative object size and the IoU-score of a class ($r = 0.40$).

C. Qualitative Examples

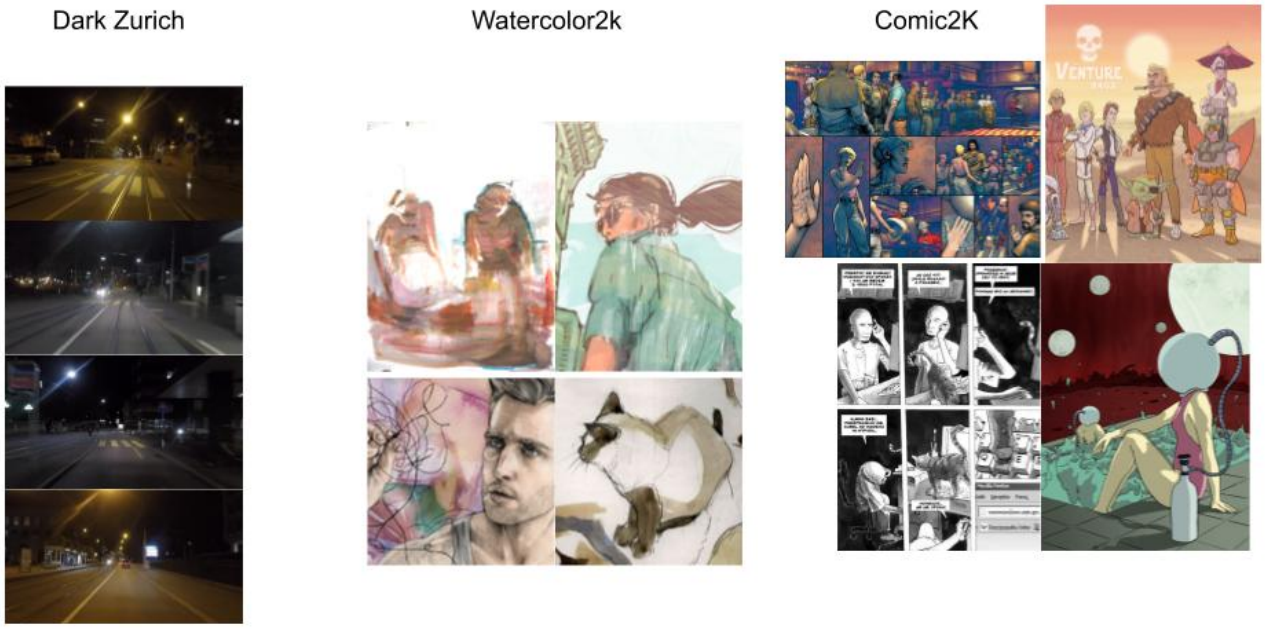

Figure S7. **Ground truth examples of the tokenized datasets.**

Figure S8. **Textual inversion and Dreambooth tokens of Cityscapes to Dark Zurich.**

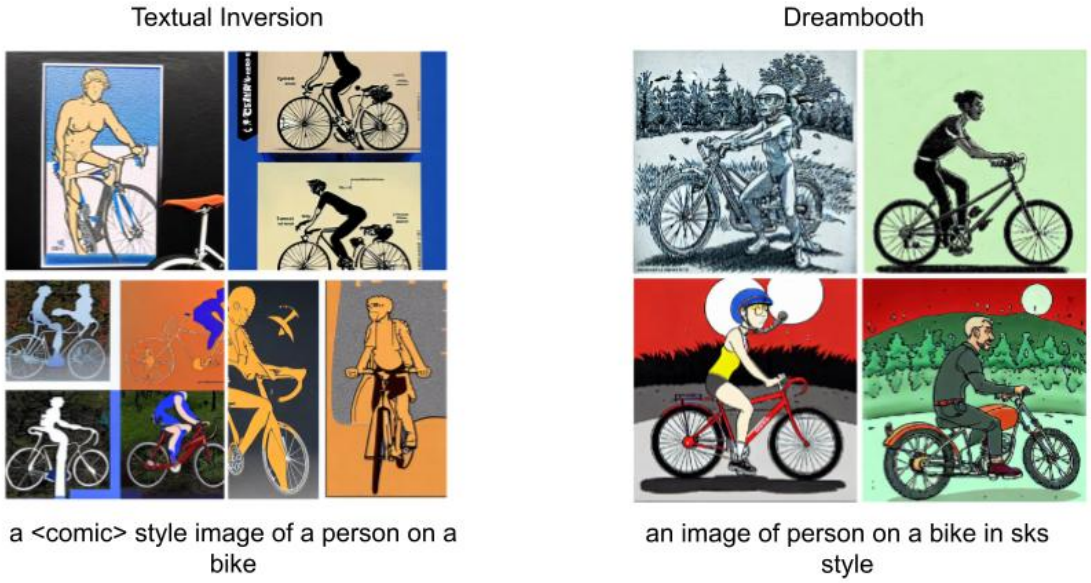

Figure S9. **Textual inversion and Dreambooth tokens of VOC to Comic.**

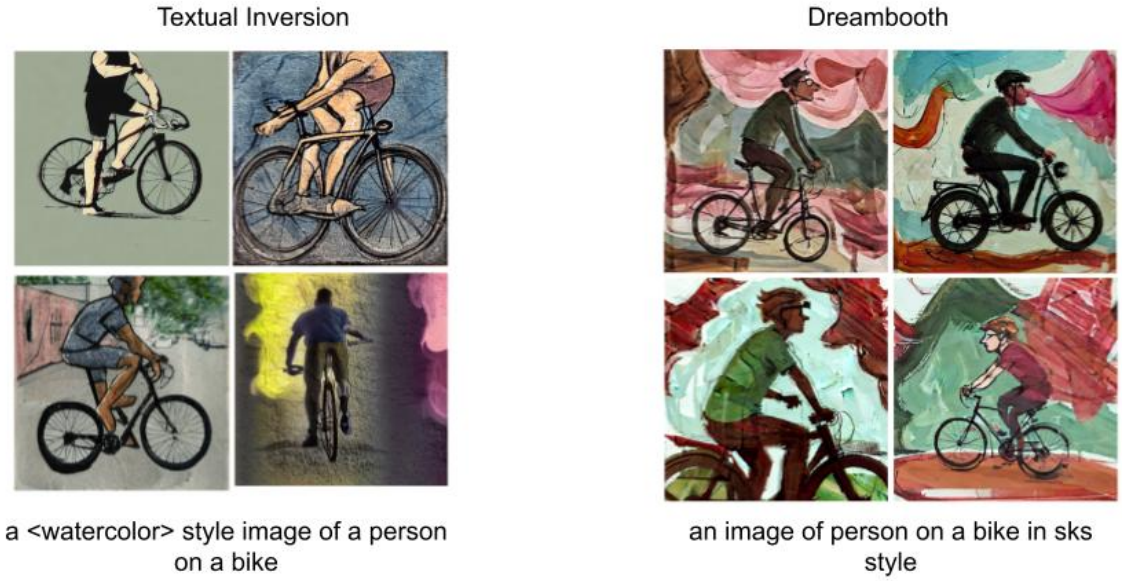

Figure S10. **Textual inversion and Dreambooth tokens of VOC to Watercolor.**

Figure S11. **Predictions (top) and Ground Truth (bottom) visualizations for Pascal VOC2012.**

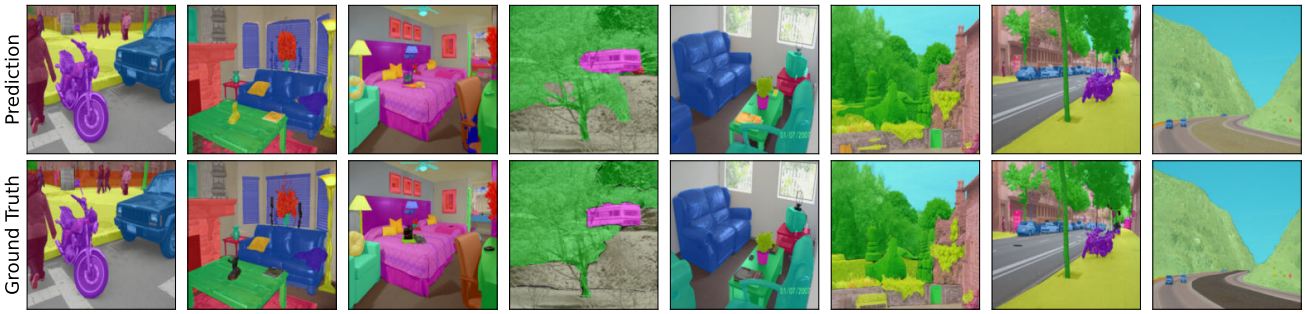

Figure S12. **Predictions (top) and Ground Truth (bottom) visualizations for ADE20K.**

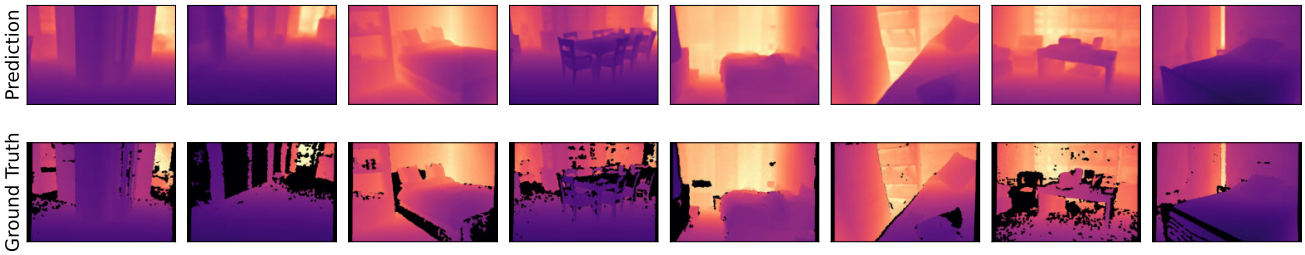

Figure S13. **Predictions (top) and Ground Truth (bottom) visualizations for NYUv2 Depth.**

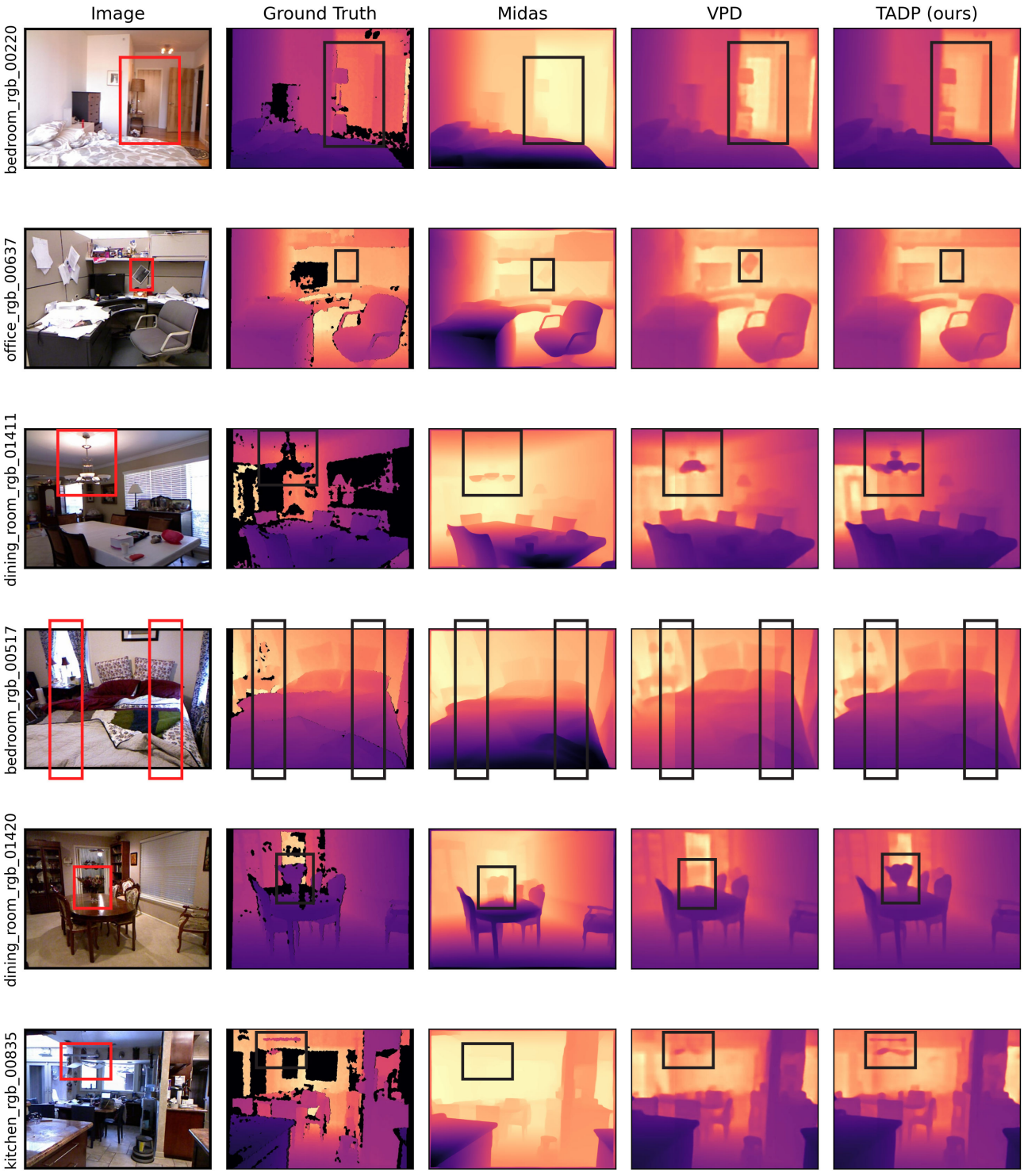

Figure S14. **Depth Estimation Comparison: Image, Ground Truth, and Prediction visualizations for Midas, VPD, and TADP (ours) in NYUv2 Depth.** Black boxes (red on original image) show where TADP is better than Midas and/or VPD.

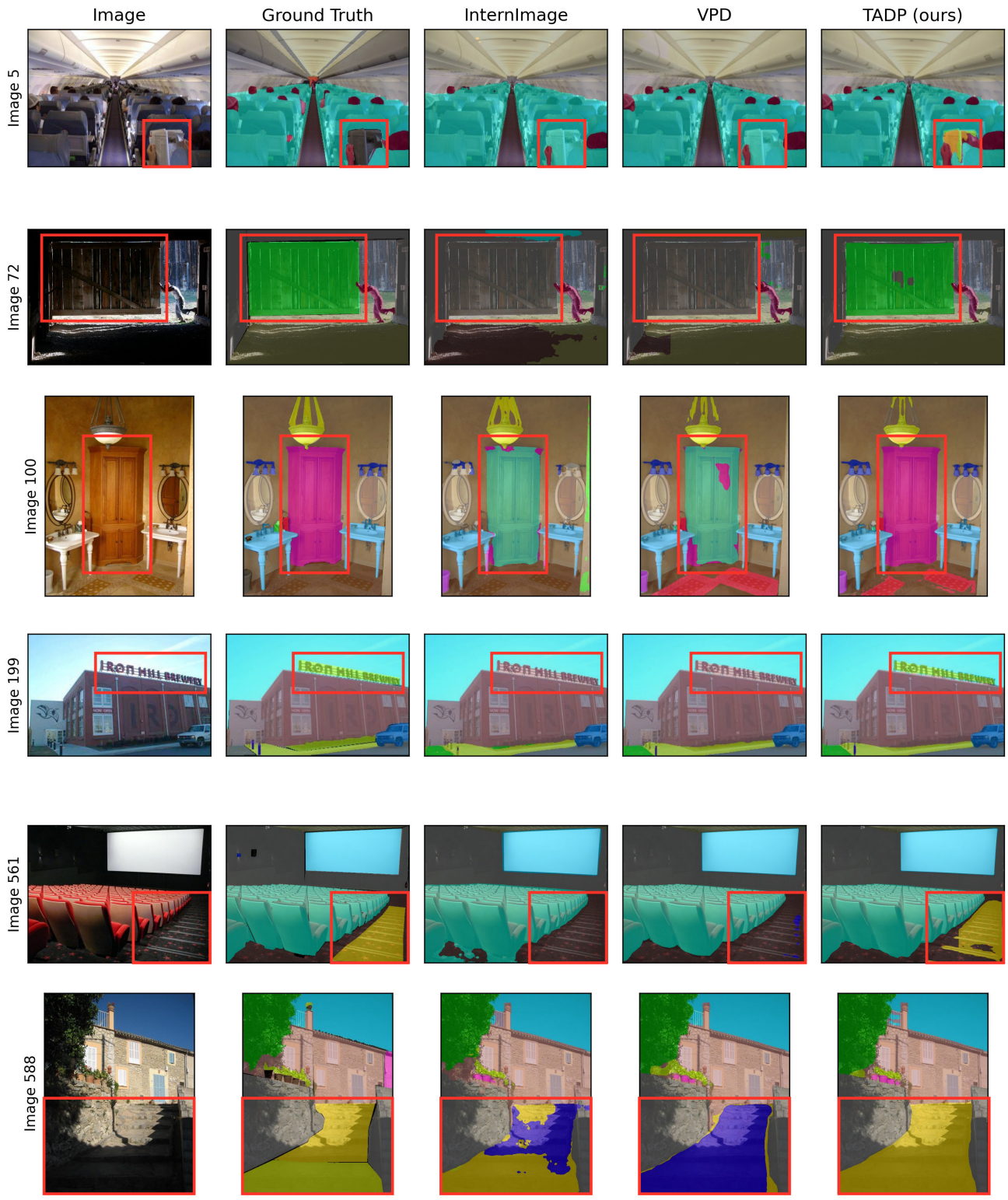

Figure S15. **Image Segmentation Comparison: Image, Ground Truth, and Prediction visualizations for InternImage, VPD, and TADP (ours) in ADE20K.** Red boxes show where TADP is better than InternImage and/or VPD.

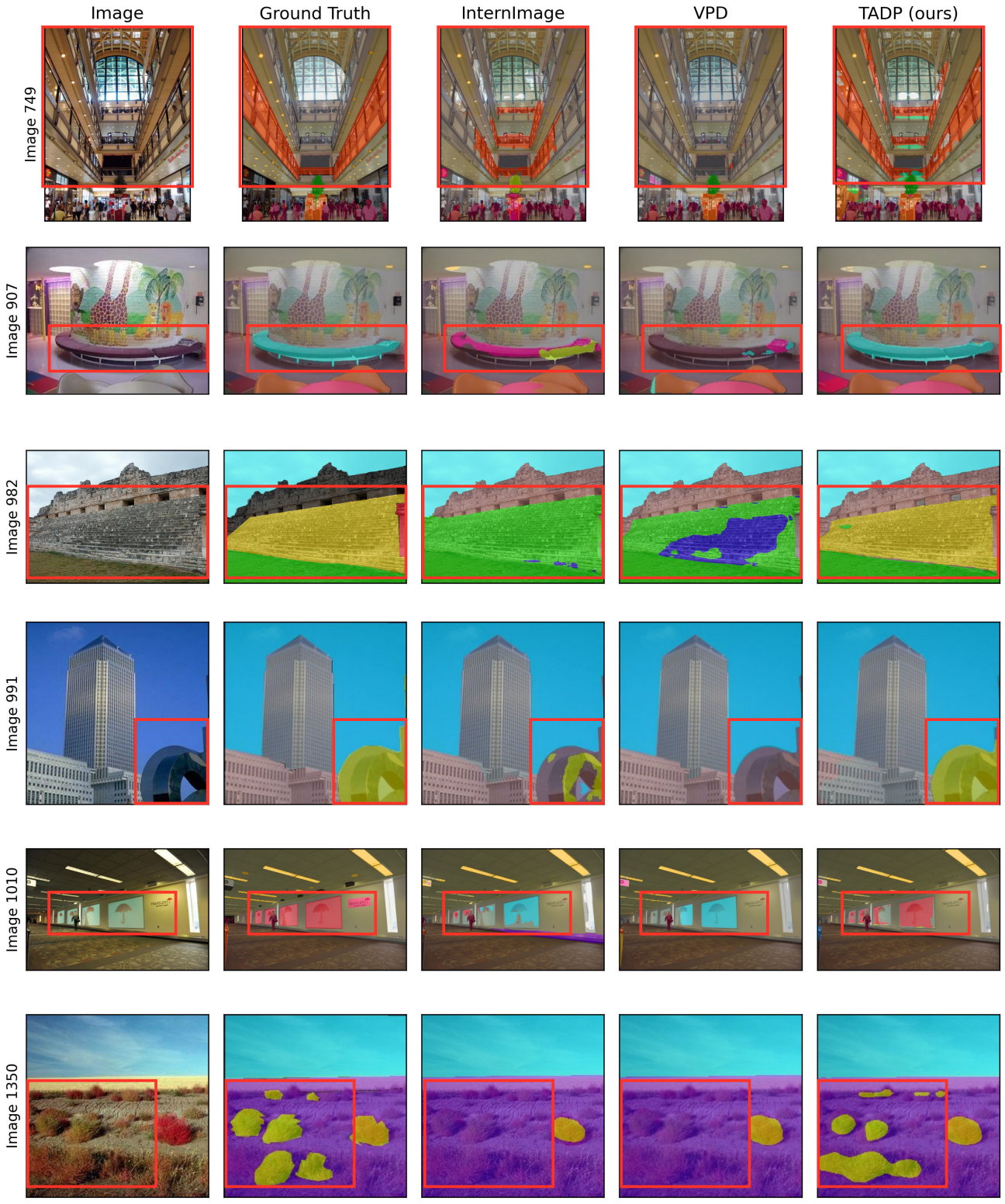

Figure S16. **Image Segmentation Comparison: Image, Ground Truth, and Prediction visualizations for InternImage, VPD, and TADP (ours) in ADE20K.** Red boxes show where TADP is better than InternImage and/or VPD.

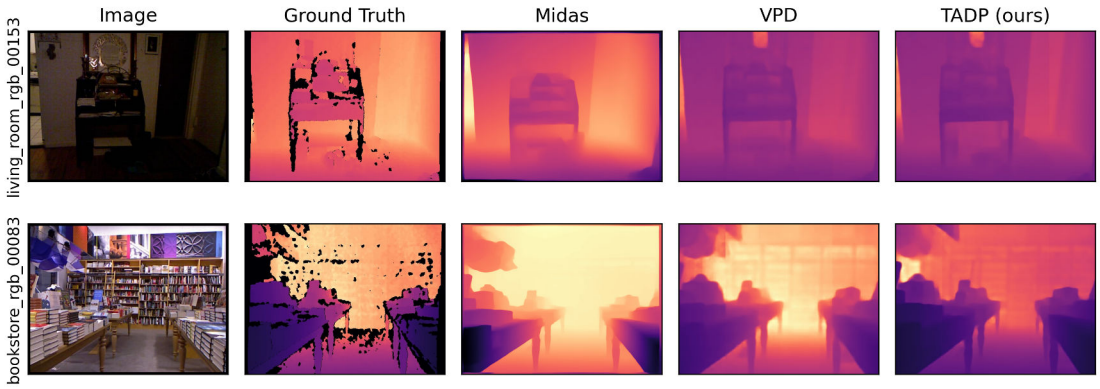

Figure S17. **Depth Estimation Comparison: Image, Ground Truth, and Prediction visualizations for Midas, VPD, and TADP (ours) in NYUv2 Depth.** TADP is worse than Midas and/or VPD in these images in terms of the general scale

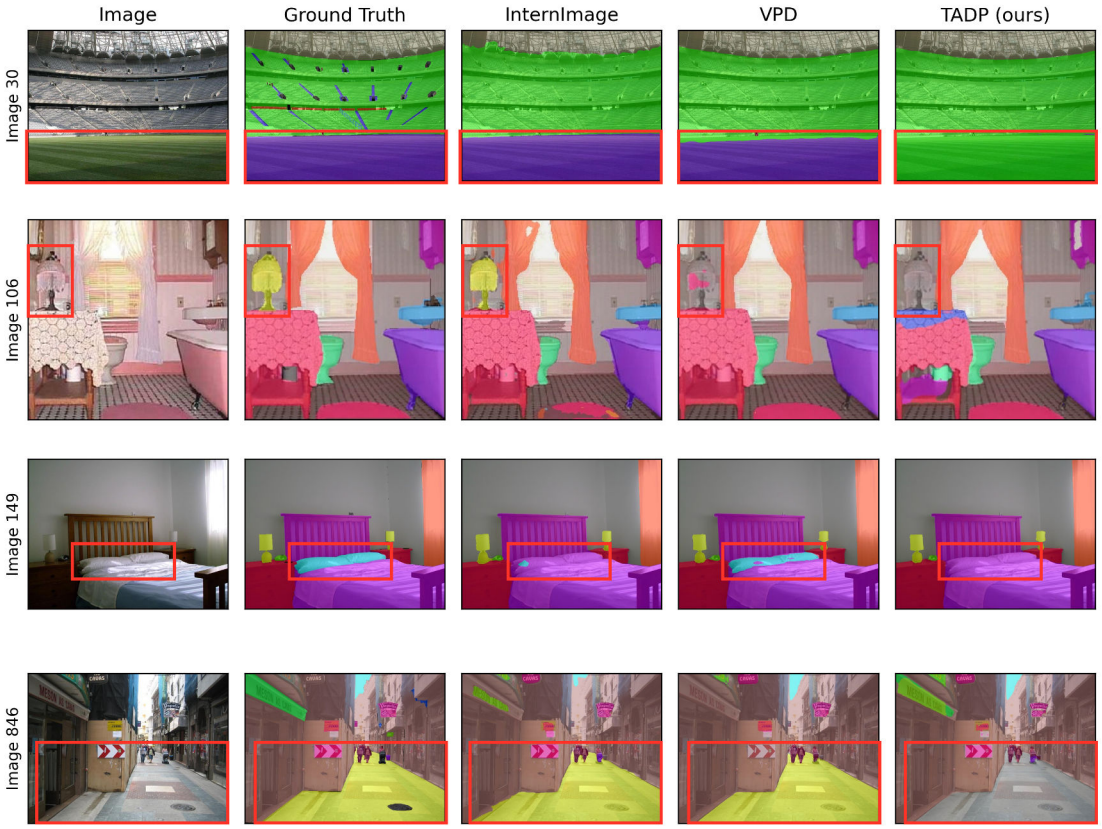

Figure S18. **Image Segmentation Comparison: Image, Ground Truth, and Prediction visualizations for InternImage, VPD, and TADP (ours) in ADE20K.** Red boxes show where TADP is worse than InternImage and/or VPD.

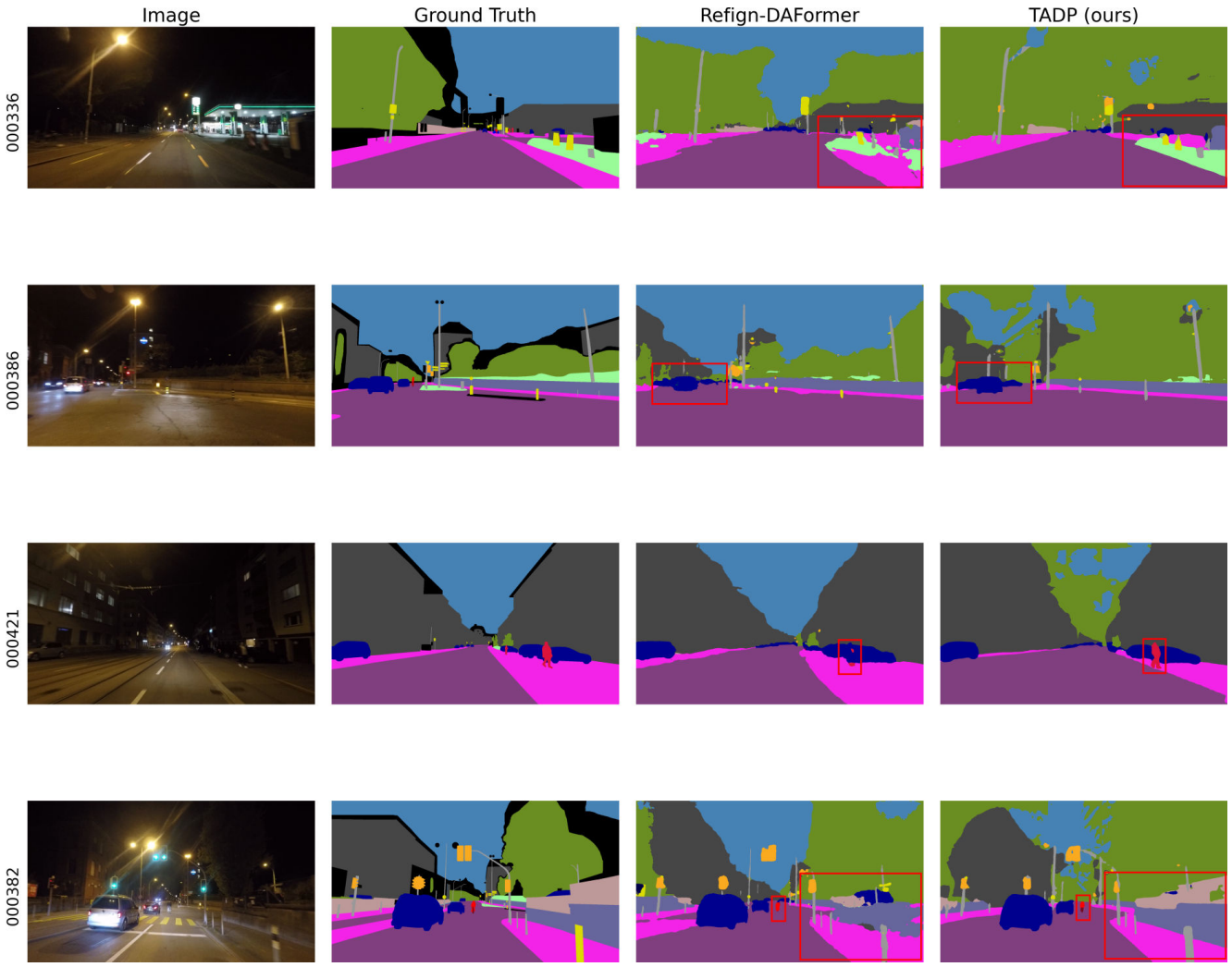

Figure S19. **Cross-domain Image Segmentation Comparison: Image, Ground Truth, and Prediction visualizations for Refign-DAFormer, and TADP (ours) for Cityscapes to Dark Zurich Val.** Red boxes show where TADP is better than Refign-DAFormer.

Figure S20. **Cross-domain Object Detection Comparison: Image, Ground Truth, and Prediction visualizations for DASS, and TADP (ours) for Pascal VOC to Watercolor2k.** Red boxes show the detections of each model. Notice that TADP not only beats DASS mostly, but also finds more objects than the ones annotated in the ground truth.

D. Implementation Details

To isolate the effects of our text-image alignment method, we ensure our model setup precisely follows prior work. Following VPD [53], we jointly train the task-specific head and the diffusion backbone. The learning rate of the backbone is set to 1/10 the learning rate of the head to preserve the benefits of pre-training better. We describe the different tasks by describing H and $\mathcal{L}_H$. We use an FPN [24] head with a cross-entropy loss for segmentation. We use the same convolutional head used in VPD for monocular depth estimation with a Scale-Invariant loss [12]. For object detection, we use a Faster-RCNN head with the standard Faster-RCNN loss [34][1]. Further details of the training setup can be found in Tab. S3 and Tab. S4. In our single-domain tables, we include our reproduction of VPD, denoted with a (R). We compute our relative gains with our reproduced numbers, with the same seed for all experiments.

Hyperparameter	**Value**
Learning Rate	0.00008
Batch Size	2
Optimizer	AdamW
Weight Decay	0.005
Warmup Iters	1500
Warmup Ratio	$1e-6$
U-Net Learning Rate Scale	0.01
Training Steps	80000

(a) ADE20k - full schedule

Hyperparameter	**Value**
Learning Rate	0.00016
Batch Size	2
Optimizer	AdamW
Weight Decay	0.005
Warmup Iters	150
Warmup Ratio	$1e-6$
Unet Learning Rate Scale	0.01
Training Steps	8000

(b) ADE20k - fast schedule 8k

Hyperparameter	**Value**
Learning Rate	0.00016
Batch Size	2
Optimizer	AdamW
Weight Decay	0.005
Warmup Iters	75
Warmup Ratio	$1e-6$
Unet Learning Rate Scale	0.01
Training Steps	4000

(c) ADE20k - fast schedule 4k

Hyperparameter	**Value**
Learning Rate	$5e-4$
Batch Size	3
Optimizer	AdamW
Weight Decay	0.1
Layer Decay	0.9
Epochs	25
Drop Path Rate	0.9

(d) NYUv2

Hyperparameter	**Value**
Learning Rate	$5e-4$
Batch Size	3
Optimizer	AdamW
Weight Decay	0.1
Layer Decay	0.9
Epochs	1
Drop Path Rate	0.9

(e) NYUv2 - fast schedule

Hyperparameter	**Value**
Learning Rate	0.00001
Batch Size	2
Gradient Accumulation	4
Epochs	15
Optimizer	AdamW
Weight Decay	0.01

(f) Pascal VOC

Table S3. **Single-Domain Hyperparameters.**

[1] Object detection was not explored in VPD.

Hyperparameter	Value
Learning Rate	0.00008
Batch Size	2
Optimizer	AdamW
Weight Decay	0.005
Warmup Iters	1500
Warmup Ratio	$1e-6$
Unet Learning Rate Scale	0.01
Training Steps	40000

(a) Cityscapes → Dark Zurich & NightTime Driving

Hyperparameter	Value
Learning Rate	0.00001
Batch Size	2
Epochs	100
Optimizer	AdamW
Weight Decay	0.01
Learning Rate Schedule	Lambda

(b) Pascal VOC → Watercolor & Comic

Hyperparameter	Value
Prior Preservation Cls Images	200
Learning Rate	$5e-6$
Training Steps	1000

(c) Dreambooth Hyperparameters

Hyperparameter	Value
Steps	3000
Learning Rate	$5.0e-04$
Batch Size	1
Gradient Accumulation	4

(d) Textual Inversion Hyperparameters

Table S4. **Cross-Domain Hyperparameters.**

D.1. Model personalization

For textual inversion, we use 500 images from DZ-train and five images for W2K and C2K and train all tokens for 1000 steps. We use a constant learning rate scheduler with a learning rate of $5e-4$ and no warmup. For Dreambooth, we use the same images as in textual inversion but train the model for 500 steps (DZ) steps or 1000 steps (W2K and C2K). We use a learning rate of $2e-6$ with a constant learning rate scheduler and no warmup. We use no prior preservation loss.