PASTA: Proportional Amplitude Spectrum Training Augmentation for Syn-to-Real Domain Generalization (Supplementary Material)

Prithvijit Chattopadhyay* Kartik Sarangmath* Vivek Vijaykumar Judy Hoffman
Georgia Institute of Technology
{prithvijit3, ksarangmath3, vivekvjk, judy}@gatech.edu

A. Overview

This supplementary is organized as follows. In Sec. A.1, we first expand on implementation and training details from the main paper. Then, in Sec. A.2, we provide per-class synthetic-to-real generalization results (see Sec. 5.1 of the main paper). Sec. A.3 goes through an empirical analysis of the amplitude spectra for synthetic and real images. Sec. A.4 discusses the computational complexity of PASTA. Next, Sec. A.5 contains more qualitative examples of PASTA augmentations and predictions for semantic segmentation. Finally, Sec. A.6 summarizes the licenses associated with different assets used in our experiments.

A.1. Implementation and Training Details

In this section, we outline our training and implementation details for each of the three tasks – Semantic Segmentation, Object Detection, and Object Recognition. We also summarize these details in Tables. 1a, 1b, and 1c.

Semantic Segmentation (see Table. 1a). For our primary semantic segmentation (SemSeg) experiments (in Tables 1, 4, and 6), we use the DeepLabv3+ [3] architecture with backbones – ResNet-50 (R-50) [9] and ResNet-101 (R-101) [9]. In Sec. A.2, we report additional results with DeepLabv3+ using the MobileNetv2 (Mn-v2) [22] backbone. We adopt the hyper-parameter (and distributed training) settings used by [5] for training. Similar to [5], we train ResNet-50, ResNet-101 and MobileNet-v2 models in a distributed manner across 4, 4 and 2 GPUs respectively. We use SGD (with momentum 0.9) as the optimizer with an initial learning rate of 10^{-2} and a polynomial learning rate schedule [14] with a power of 0.9. Our models are initialized with supervised ImageNet [13] pre-trained weights. We train all our models for 40k iterations with a batch size of 16 for GTAV. Our segmentation models are trained on the train split of GTAV and evaluated on the validation splits of the target datasets (Cityscapes, BDD100K and Mapillary). For segmentation, PASTA is applied with a base set of positional and photometric augmentations (PASTA first and then the base augmentations) – GaussianBlur, ColorJitter, RandomCrop, RandomHorizontalFlip and RandomScaling. For RandAugment [7], we only consider the vocabulary of photometric augmentations for segmentation & detection. We conduct ablations (within computational constraints) for the best performing RandAugment setting using R-50 for syn-to-real generalization and find that best performance is achieved when 8 (photometric) augmentations are sampled at the highest severity level (30) from the augmentation vocabulary for application. Whenever we train a prior generalization approach, say ISW [5] or IBN-Net [16], we follow the same set of hyper-parameter configurations as used in the respective papers. Table. 1a includes details for SegFormer and HRDA runs. All models except SegFormer and HRDA were trained across 3 random seeds.

Object Detection (see Table. 1b). For object detection (ObjDet), we use the Faster-RCNN [19] architecture with ResNet-50 and ResNet-101 backbones (see Tables 2, 5 in the main paper). Consistent with prior work [12], we train on the entirety of Sim10K [11] (source dataset) for 10k iterations and pick the last checkpoint for Cityscapes (target dataset) evaluation. We use SGD with momentum as our optimizer with an initial learning rate of 10^{-2} (adjusted according to a step learning rate schedule) and a batch size of 32. Our models are initialized with supervised ImageNet [13] pre-trained weights. All models are trained on 4 GPUs in a distributed manner. For detection, we also compare PASTA against RandAugment [7] and Photometric Distortion (PD). The sequence of operations in PD to augment input images are – randomized brightness, randomized contrast, RGB→HSV conversion, randomized saturation & hue changes, HSV→RGB conversion, randomized contrast, and randomized channel swap.

Object Recognition (see Table. 1c). For our primary object recognition (ObjRec) experiments (see Table 3 in main paper), we train classifiers with ResNet-101 [9] and ViT-B/16 [8] backbones. For ResNet-101, we start from super-

[*] Equal Contribution. Correspondence to prithviijt3@gatech.edu

(a) Semantic Segmentation Training

Config	**Value**	**Value** (SegFormer)	**Value** (HRDA)
Source Data	GTAV (Train Split)	GTAV (Entirety)	GTAV (Entirety)
Target Data	Cityscapes (Val Split) BDD100K (Val Split) Mapillary (Val Split)	Cityscapes (Val Split)	Cityscapes (Val Split)
Segmentation Architecture	DeepLabv3+ [3]	SegFormer [23]	HRDA [10]
Backbones	ResNet-50 (R-50) [9] ResNet-101 (R-101) [9] MobileNetv2 (Mn-v2) [22]	MiT-B5 [23]	MiT-B5 [23]
Training Resolution	Original GTAV resolution	Original GTAV resolution	Original GTAV resolution
Optimizer	SGD	AdamW	AdamW
Initial Learning Rate	10^{-2}	6×10^{-5}	6×10^{-5}
Learning Rate Schedule	Poly-LR	Poly-LR	Poly-LR
Initialization	Imagenet Pre-trained Weights [13]	Imagenet Pre-trained Weights [13]	Imagenet Pre-trained Weights [13]
Iterations	40k	160k	40k
Batch Size	16	4	2
Augmentations w/ PASTA	Gaussian Blur, Color Jitter, Random Crop Random Horizontal Flip, Random Scaling	Photometric Distortion Random Crop, Random Flip	Photometric Distortion Random Crop, Random Flip
Model Selection Criteria	Best in-domain validation performance	End of training	End of training
GPUs	4 (R-50/101) or 2 (Mn-v2)	4	1

(b) Object Detection Training

Config	**Value**
Source Data	Sim10K
Target Data	Cityscapes (Val Split)
Segmentation Architecture	Faster-RCNN [19]
CNN Backbones	ResNet-50 (R-50) [9] ResNet-101 (R-101) [9]
Training Resolution	Original Sim10K resolution for R-50, R-101
Optimizer	SGD (momentum = 0.9)
Initial Learning Rate	10^{-2}
Learning Rate Schedule	Step-LR, Warmup 500 iterations, Warmup Ratio 0.001 Steps 6k & 8k iterations
Initialization	Imagenet Pre-trained Weights [13]
Iterations	10k
Batch Size	32
Augmentations w/ PASTA	Resize, Random Flip
Model Selection Criteria	End of training
GPUs	4

(c) Object Recognition Training

Config	**Value**	**Value** (for Table. 3, following [4])
Source Data	VisDA-C Synthetic	VisDA-C Synthetic
Target Data	VisDA-C Real	VisDA-C Real
Backbone	ResNet-101 (R-101) [9] ViT-B/16 [8] (Sup & DINO [1])	ResNet-101 (R-101) [9]
Optimizer	SGD w/ momentum (0.9)	SGD w/ momentum (0.9)
Initial Learning Rate	2×10^{-4}	10^{-4}
Weight Decay	10^{-4}	5×10^{-4}
Initialization	Imagenet [13]	Imagenet [13]
Epochs	10	30
Batch Size	128	32
Augmentations w/ PASTA	RandomCrop, RandomHorizontalFlip	RandAugment [7]
Model Selection Criteria	Best in-domain val performance	Best in-domain val performance
GPUs	1 (CNN) or 4 (ViT)	1

Table 1: **Implementation & Optimization Details.** We summarize details surrounding dataset, training, optimization and model selection criteria for our semantic segmentation, object detection and object recognition experiments. More detailed configs in code.

Method	Real mIoU ↑				
	G→C	G→B	G→M	Avg	Δ
1 Baseline (B) [5]*	25.92	25.73	26.45	26.03	
2 B + PASTA	**39.75**	**37.54**	**43.28**	**40.19** ±0.45	+14.16
3 IBN-Net [16]*	30.14	27.66	27.07	28.29	
4 IBN-Net + PASTA	**37.57**	**36.97**	**40.91**	**38.48** ±0.75	+10.19
5 ISW [5]*	30.86	30.05	30.67	30.53	
6 ISW + PASTA	**37.99**	**37.49**	**42.44**	**39.31** ±1.26	+8.79

Table 2: **MobileNet-v2 [22] GTAV→Real (SemSeg) Generalization Results.** Semantic Segmentation DeepLabv3+ models trained on GTAV (G) and evaluated on {Cityscapes (C), BDD100K (B), Mapillary (M)}. * indicates numbers drawn from published manuscripts. **Bold** indicates best. Δ indicates (absolute) improvement. PASTA improves a vanilla baseline (rows 1, 2) and is complementary to existing methods (rows 3-6).

vised ImageNet [13] pre-trained weights. For ViT-B/16 we start from both supervised and self-supervised (DINO [1]) ImageNet pre-trained weights. We train these classifiers for 10 epochs with SGD (momentum 0.9, weight decay 10^{-4}) with an initial learning rate of 2×10^{-4} with cosine annealing – the newly added classifier and bottleneck layers [2] were trained with 10× more learning rate as the rest of the network. We train on 90% of the (synthetic) VisDA-C training split (and use the remaining 10% for model selection) with a batch size of 128 in a distributed manner across 4 GPUs. We use RandomCrop, RandomHorizontalFlip as additional augmentations with PASTA. In Sec. A.2, we provide additional results demonstrating how PASTA is complementary to CSG [4], a state-of-the-art generalization method on VisDA-C. For these experiments, to ensure fair comparisons, we train ResNet-101 based classifiers (with supervised ImageNet pre-trained weights) with same configurations as [4]. This includes the use of an SGD (with momentum 0.9) optimizer with a learning rate of 10^{-4}, weight decay of 5×10^{-4} and a batch size of 32. These models are trained for 30 epochs. CSG [4] also uses RandAugment [7] as an augmentation – we check the effectiveness of PASTA when applied with and without RandAugment during training.

A.2. Synthetic-to-Real Generalization Results

MobileNet-v2 GTAV→Real Generalization Results. Our key generalization results for semantic segmentation (SemSeg) (in Tables. 1, 4 and 6) are with ResNet-50 and ResNet-101 backbones. In Table. 2, we also report results when PASTA is applied to DeepLabv3+ [3] models with MobileNetv2 – a lighter backbone tailored for resource constrained settings. We find that PASTA substantially improves a vanilla baseline (by 14+ absolute mIoU points; rows 1, 2) and is complementary to existing methods (rows 3-6).

Method	Accuracy	Δ
1 Oracle (IN-1k) [4]*	53.30	
2 Baseline (Syn. Training) [4]*	49.30	
3 CSG [4]*	64.05	
4 CSG (RandAug)	63.84 ±0.29	+14.54
5 CSG (PASTA)	64.29 ±0.56	+14.99
6 CSG (RandAug + PASTA)	**65.86** ±1.13	+16.56

Table 3: **PASTA is complementary to CSG [4].** We apply PASTA to CSG [4]. Since CSG inherently uses RandAug, we also report results with and without the use of RandAug when PASTA is applied. * indicates drawn directly from published manuscripts. We report class-balanced accuracy on the real (val split) target data of VisDA-C. Results reported across 3 runs. **Bold** indicates best. Δ indicates absolute improvement over baseline.

PASTA complementary to CSG [4]. To evaluate the efficacy of PASTA for object recognition (ObjRec), in Table. 3, we apply PASTA to CSG [4], a state-of-the-art generalization method on VisDA-C Syn→Real. Since CSG inherently uses RandAugment [7], we apply PASTA ($\alpha = 10, k = 1, \beta = 0.5$) both with (row 6) and without (row 5) RandAugment and find that applying PASTA improves over vanilla CSG (row 4) in both conditions.

Per-class GTAV→Real Generalization Results. Tables 4, 5 and 6 include per-class synthetic-to-real generalization results when a DeepLabv3+ (R-50 backbone) model trained on GTAV is evaluated on Cityscapes, BDD100K and Mapillary respectively. For GTAV→Cityscapes (see Table. 4), we find that Baseline + PASTA consistently improves over Baseline and RandAugment. For IBN-Net and ISW in this setting, we observe consistent improvements (except for the classes *terrain* and *fence*). For GTAV→BDD100K (see Table. 5), we find that for the Baseline, while PASTA outperforms RandAugment on the majority of classes, both are fairly competitive and outperform the vanilla Baseline approach. For IBN-Net and ISW, PASTA almost always outperforms the vanilla approaches (except for the class *wall*). For GTAV→Mapillary (see Table. 6), for Baseline, we find that PASTA outperforms the vanilla approach and RandAugment. For IBN-Net and ISW, PASTA outperforms the vanilla approaches with the exception of the classes *train* and *fence*.

PASTA helps SYNTHIA→Real Generalization. We conducted additional syn-to-real experiments using SYNTHIA [21] as the source domain and Cityscapes, BDD100K and Mapillary as the target domains. For a baseline DeepLabv3+ model (R-101), we find that PASTA – (1) provides strong improvements over the vanilla baseline (31.77% mIoU, +3.91% absolute improvement) and (2) is competitive with RandAugment (32.30% mIoU). More generally, we find that syn-to-real generalization performance is worse when SYNTHIA is used as the source domain as opposed to GTAV – for instance, ISW [5] achieves an average mIoU of 31.07% (SYNTHIA) as opposed to 35.58% (GTAV). SYN-

Method	road	building	vegetation	car	sidewalk	sky	pole	person	terrain	fence	wall	bicycle	sign	bus	truck	rider	light	train	motorcycle	mIoU
1 Baseline (B) [5]*	45.1	56.8	80.9	61.0	23.1	38.9	23.9	58.2	24.3	16.3	16.6	13.4	7.3	20.0	17.4	1.2	30.0	7.2	8.5	29.0
2 B + RandAug	58.5	56.3	77.3	83.7	30.3	45.1	27.3	57.8	20.6	20.9	11.4	16.9	9.7	20.3	15.0	2.4	28.1	14.0	10.3	31.9
3 B + PASTA	**84.1**	**80.5**	**85.8**	**85.9**	**40.1**	**81.8**	**31.9**	**66.0**	**31.4**	**28.1**	**29.0**	**21.8**	**28.5**	**24.5**	**28.7**	**7.0**	**32.9**	**23.4**	**27.2**	**44.1**
4 IBN-Net [16]*	51.3	59.7	85.0	76.7	24.1	67.8	23.0	60.6	**40.6**	**25.9**	14.1	15.7	10.1	23.7	16.3	0.8	30.9	4.9	11.9	33.9
5 IBN-Net + PASTA	**78.1**	**79.5**	**85.8**	**84.5**	**31.7**	**80.1**	**32.2**	**63.4**	38.8	21.7	**28.0**	**18.2**	**22.6**	**26.4**	**29.0**	**2.8**	**34.0**	**16.5**	**22.9**	**41.9**
6 ISW [5]*	60.5	65.4	85.4	82.7	25.5	70.3	25.8	61.9	38.5	**23.7**	21.6	15.5	12.2	25.4	21.1	0.0	33.3	9.3	16.8	36.6
7 ISW + PASTA	**76.6**	**78.4**	**85.6**	**83.7**	**32.5**	**83.1**	**33.1**	**63.4**	**40.4**	23.6	**27.3**	**17.4**	**22.3**	**25.7**	**30.1**	**3.3**	**35.9**	**18.2**	**19.9**	**42.1**

Table 4: **GTAV→Cityscapes per-class generalization results.** Per-class IoU comparisons for (SemSeg) syn-to-real generalization results when DeepLabv3+ (R-50 models trained on GTAV are evaluated on Cityscapes. Results are reported across 3 runs. * indicates drawn directly from published manuscripts. Class headers are in decreasing order of pixel frequency.

Method	road	sky	building	vegetation	car	sidewalk	fence	terrain	truck	pole	bus	wall	sign	person	light	bicycle	motorcycle	rider	train	mIoU
1 Baseline (B) [5]*	48.2	32.3	34.3	58.2	67.3	23.0	19.8	21.4	11.3	28.4	5.6	3.4	18.1	43.9	30.2	11.1	16.0	5.1	0.0	25.1
2 B + RandAug	75.4	82.8	67.7	**74.7**	74.1	**39.3**	**32.7**	**26.6**	22.7	**37.0**	**16.5**	5.0	23.9	51.7	35.8	12.0	29.3	**20.2**	0.0	38.3
3 B + PASTA	**86.0**	**86.6**	**74.8**	72.7	**82.8**	38.4	31.2	24.9	**23.7**	34.8	6.4	**11.2**	**26.2**	**55.1**	**37.0**	**13.3**	**38.3**	19.9	0.0	**40.2**
4 IBN-Net [16]*	68.9	66.9	56.7	66.6	70.3	28.8	21.4	22.1	12.8	31.9	7.2	6.0	21.7	50.2	35.0	18.1	23.2	5.8	0.0	32.3
5 IBN-Net + PASTA	**86.1**	**87.6**	**74.9**	**72.3**	**82.3**	**36.6**	**30.6**	**26.2**	**25.3**	**37.1**	**10.8**	**13.2**	**25.5**	**56.0**	**36.8**	**21.4**	**38.9**	**26.0**	0.0	**41.5**
6 ISW [5]*	74.9	77.4	65.2	69.0	72.4	30.4	22.6	26.2	16.2	34.9	6.1	**11.5**	22.2	50.3	36.9	11.4	31.3	10.0	0.0	35.2
7 ISW + PASTA	**86.5**	**87.9**	**74.0**	**73.0**	**83.2**	**37.7**	**28.6**	**28.1**	**23.4**	**37.2**	**7.8**	11.3	**25.0**	**55.1**	**37.8**	**23.6**	**35.5**	**22.4**	0.0	**41.0**

Table 5: **GTAV→BDD100K per-class generalization results.** Per-class IoU comparisons for (SemSeg) syn-to-real generalization results when DeepLabv3+ (R-50 models trained on GTAV are evaluated on BDD100K. Results are reported across 3 runs. * indicates drawn directly from published manuscripts. Class headers are in decreasing order of pixel frequency.

Method	sky	road	vegetation	building	sidewalk	car	fence	pole	terrain	wall	sign	truck	person	bus	light	bicycle	rider	motorcycle	train	mIoU
1 Baseline (B) [5]*	42.2	46.8	64.9	33.5	24.9	72.3	14.4	27.7	23.8	6.7	8.5	23.7	53.7	7.0	35.8	18.4	4.9	15.5	10.8	28.2
2 B + RandAug [7]	51.7	59.6	75.5	39.5	33.9	81.3	22.6	37.2	24.6	4.2	32.4	31.2	56.5	13.4	36.2	18.0	11.9	21.4	5.0	34.5
3 B + PASTA	**93.3**	**83.0**	**76.3**	**76.9**	**40.2**	**83.3**	**27.1**	**40.9**	**37.1**	**19.2**	**50.4**	**35.2**	**63.3**	**19.5**	**41.4**	**29.4**	**25.2**	**38.1**	**15.2**	**47.1**
4 IBN-Net [16]*	82.0	66.4	73.5	57.1	32.9	73.1	24.9	31.5	28.4	10.5	38.9	30.7	56.4	16.0	38.0	18.6	9.1	16.6	**12.6**	37.7
5 IBN-Net + PASTA	**94.4**	**81.7**	**76.1**	**76.9**	**40.4**	**80.8**	**27.1**	**40.3**	**38.7**	**19.0**	**43.2**	**38.0**	**62.0**	**20.5**	**39.3**	**25.7**	**23.6**	**31.6**	12.4	**45.9**
6 ISW [5]*	88.2	74.8	74.3	66.1	36.2	78.7	**26.0**	35.4	30.2	15.2	36.6	33.3	58.6	14.4	37.9	17.8	11.1	20.4	11.0	40.3
7 ISW + PASTA	**94.6**	**82.9**	**76.7**	**76.0**	**41.9**	**81.8**	25.8	**40.4**	**40.9**	**18.8**	**43.1**	**34.1**	**61.6**	**19.9**	**40.2**	**24.5**	**22.3**	**30.5**	**11.6**	**45.7**

Table 6: **GTAV→Mapillary per-class generalization results.** Per-class IoU comparisons for (SemSeg) syn-to-real generalization results when DeepLabv3+ (R-50 models trained on GTAV are evaluated on Mapillary. Results are reported across 3 runs. * indicates drawn directly from published manuscripts. Class headers are in decreasing order of pixel frequency.

Method	Base Augmentations		Real mIoU	Δ
	Positional	Photometric		
1 Baseline (B)	✓	✓	26.99	
2 B + PASTA	✗	✗	40.25	+13.26
3 B + PASTA	✓	✗	40.37	+13.38
4 B + PASTA	✓	✓	**41.90**	+14.91

Table 7: **PASTA vs Base Augmentations.** Semantic Segmentation DeepLabv3+ (R-50) models trained on GTAV (at an input resolution of 1024 × 560 due to compute constraints) and evaluated on {Cityscapes, BDD100K, Mapillary}. **Bold** indicates best. Δ indicates (absolute) improvement over Baseline.

THIA has significantly fewer images compared to GTAV (9.4k vs 25k), which likely contributes to relatively worse generalization performance.

PASTA and Base Augmentations. PASTA is applied with some consistent color and positional augmentations (see Section. A.1). To understand if PASTA alone leads to any improvements, in Table. 7, we conduct a controlled experiments where we train a baseline DeepLabv3+ model (R-50) on GTAV (by downsampling input images to a resolution of 1024 × 560 due to compute constraints) with different augmentations and evaluate on real data (Cityscapes, BDD100K and Mapillary). We find that applying PASTA alone leads to significant improvements (13+ absolute mIoU points; row 2) and including the positional (row 3) and photometric (row 4) augmentations leads to further improvements.

A.3. Amplitude Analysis

PASTA relies on the empirical observation that synthetic images have less variance in their high frequency components compared to real images. In this section, we first show how this observation is widespread across a set of syn-to-real shifts over fine-grained frequency band discretizations and then demonstrate how PASTA helps counter this discrepancy.

Fine-grained Band Discretization. For Fig. 2 [Right] in the main paper, the low, mid and high frequency bands are chosen such that the first (lowest) band is $1/3$ the height of the image (includes all spatial frequencies until $1/3$rd of the image height), second band is up to $2/3$ the height of the image excluding band 1 frequencies, and the third band considers all the remaining frequencies. To investigate similar trends across fine-grained frequency band discretizations, we split the amplitude spectrum into 3, 5, 7, and 9 frequency bands in the manner described above, and analyze the diversity of these frequency bands across multiple datasets. Across 7 domain shifts (see Fig. 1 and 2) – {GTAV, SYNTHIA} → {Cityscapes, BDD100K, Mapillary}, and VisDA-C Syn→Real, we find that (1) for every dataset (whether synthetic or real), diversity decreases as we head towards higher frequency bands and (2) synthetic images exhibit less diversity in high-frequency bands at all considered levels of granularity.

Method	Time (s)	
	Fwd-Bwd Pass	Data Transf.
1 Base	0.838	0.069
2 RandAug	1.318	0.534
3 PASTA (CPU)	5.028	4.256
4 PASTA (GPU)	**1.176**	**0.048**

Table 8: **PASTA runtime on CPU and GPU.** Time (in seconds) taken by RandAugment and PASTA for 1914 × 1052 sized images on an A40 GPU. **Bold** indicates fastest.

Increase in amplitude variations post-PASTA. Next, we observe how PASTA effects the diversity of the amplitude spectrums on GTAV and VisDA-C. Similar to above, we split the amplitude spectrum into 3, 5, 7, and 9 frequency bands, and we analyze the diversity of these frequency bands before and after applying PASTA to images (see Fig. 3 and 4). For synthetic images from GTAV, when PASTA is applied, we observe that the standard deviation of amplitude spectrums increases from 0.4 to 0.497, 0.33 to 0.51 and 0.3 to 0.52 for the low, mid and high frequency bands respectively. As expected, we observe maximum increase for the high-frequency bands.

A.4. Computational complexity of PASTA

PASTA is a fast data augmentation step as it can be run entirely on GPUs using FFT from `torch.fft`. Overall, this makes our implementation faster than prior augmentation like RandAug which operates on CPU. Table 8 compares time taken for a GTAV batch (2 images) for the most expensive SemSeg setting (SegFormer-B5) and shows that PASTA (GPU) is faster than RandAug implementations. PASTA runtime depends on FFT, and therefore follows the scaling behavior of FFT for larger inputs.

A.5. Qualitative Examples

PASTA Augmentation Samples. Fig. 5 includes more examples of images from synthetic datasets (from GTAV and VisDA-C), when RandAugment and PASTA are applied.

Semantic Segmentation Predictions. We include qualitative examples of semantic segmentation predictions on Cityscapes made by Baseline, IBN-Net and ISW (DeepLabv3+, ResNet-50) trained on GTAV (corresponding to Tables 1, 4 and 6 in the main paper) Fig. 6, 8 and 10 respectively when different augmentations are applied (RandAugment and PASTA). The Cityscapes images we show predictions on were selected randomly. We include RandAugment predictions only for the Baseline. To get a better sense of the kind of mistakes made by different approaches, we also include the difference between the predictions and

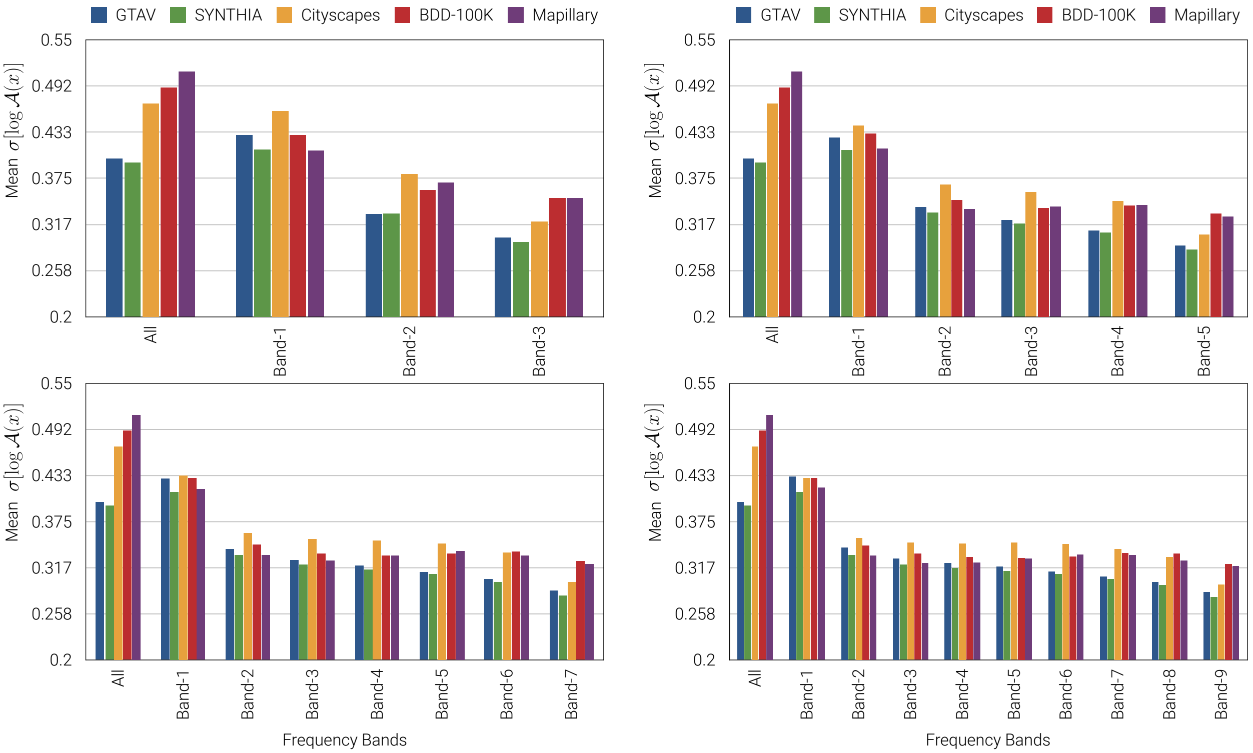

Figure 1: **Variations in amplitude values across fine-grained frequency bands (GTAV→Real and SYNTHIA→Real).** Across domain shifts GTAV→Real and SYNTHIA→Real, and four settings corresponding to fine-grained frequency bands (3, 5, 7 and 9 bands; increasing in frequency from Band-1 to Band-n), we find that synthetic images have less variance in high-frequency components of the amplitude spectrum compared to real images.

ground truth segmentation masks in Fig. 7, 9 and 11 (ordered accordingly for easy reference). The difference images show the predicted classes only for pixels where the prediction differs from the ground truth.

A.6. Assets Licenses

The assets used in this work can be grouped into three categories – Datasets, Code Repositories and Dependencies. We include the licenses of each of these assets below.

Datasets. We used the following publicly available datasets in this work – GTAV [20], Cityscapes [6], BDD100K [24], Mapillary [15], Sim10K [11], and VisDA-C [18]. For GTAV, the codebase used to extract data from the original GTAV game is distributed under the MIT license.[1] The license agreement for the Cityscapes dataset dictates that the dataset is made freely available to academic and non-academic entities for non-commercial purposes such as academic research, teaching, scientific publications, or personal experimentation and that permission to use the data is granted under certain conditions.[2] BDD100K is distributed under the BSD-3-Clause license.[3] Mapillary images are shared under a CC-BY-SA license, which in short means that anyone can look at and distribute the images, and even modify them a bit, as long as they give attribution.[4] Densely annotated images for Sim10k are available freely[5] and can only be used for non-commercial applications. The VisDA-C development kit on github does not have a license associated with it, but it does include a Terms of Use, which primarily states that the dataset must be used for non-commercial and educational purposes only.[6]

Code Repositories. For our experiments, apart from code that we wrote ourselves, we build on top of three existing public repositories – RobustNet[7], MMDetection[8] and CSG[9]. RobustNet is distributed under the BSD-3-Clause license. MMDetection is distributed under Apache License 2.0[10].

[1] https://bitbucket.org/visinf/projects-2016-playing-for-data/src/master/

[2] https://www.cityscapes-dataset.com/license/

[3] https://github.com/bdd100k/bdd100k/blob/master/LICENSE

[4] https://help.mapillary.com/hc/en-us/articles/115001770409-Licenses

[5] https://fcav.engin.umich.edu/projects/driving-in-the-matrix

[6] https://github.com/VisionLearningGroup/taskcv-2017-public/tree/master/classification

[7] https://github.com/shachoi/RobustNet

[8] https://github.com/open-mmlab/mmdetection

[9] https://github.com/NVlabs/CSG

[10] https://github.com/open-mmlab/mmdetection/blob/master/LICENSE

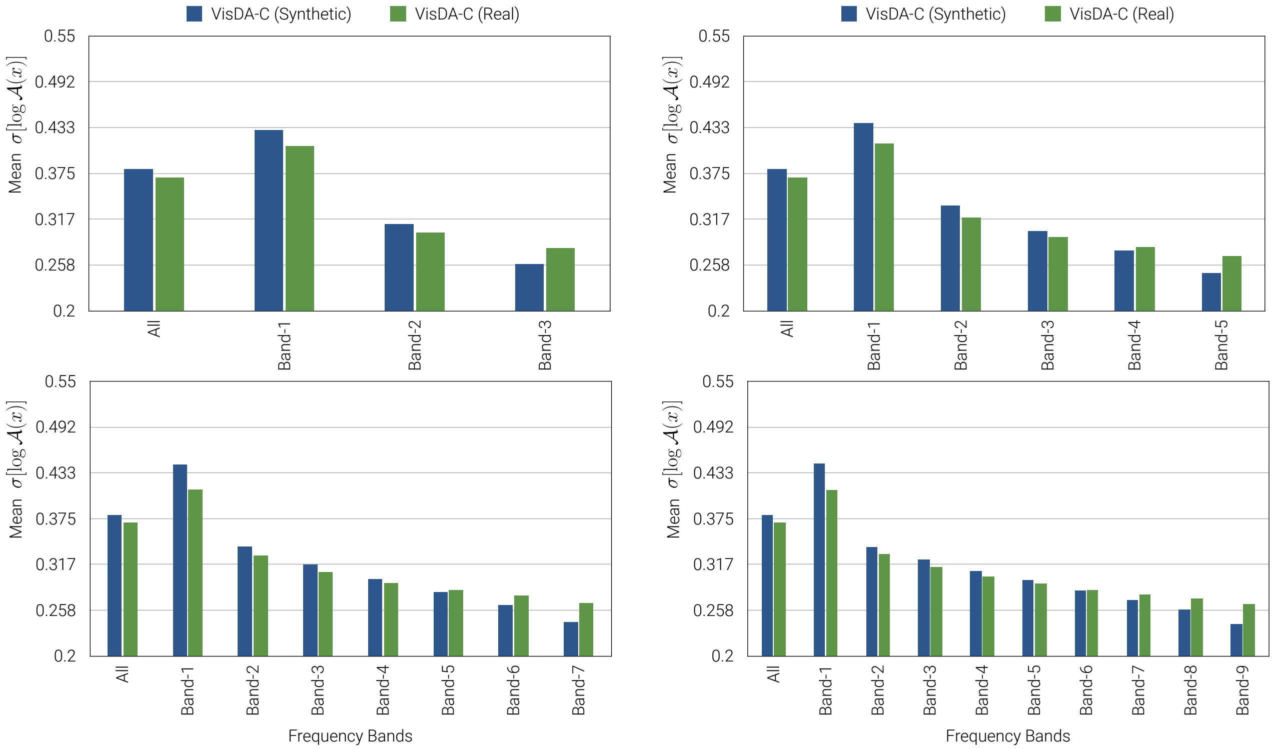

Figure 2: **Variations in amplitude values across fine-grained frequency bands (VisDA-C Synthetic→Real).** For the VisDA-C Synthetic→Real domain shift, and four settings corresponding to fine-grained frequency bands (3, 5, 7 and 9 bands; increasing in frequency from Band-1 to Band-n), we find that synthetic images have less variance in high-frequency components of the amplitude spectrum compared to real images.

CSG, released by NVIDIA, is released under a NVIDIA-specific license.[11]

Dependencies. We use Pytorch [17] as the deep-learning framework for all our experiments. Pytorch, released by Facebook, is distributed under a Facebook-specific license.[12]

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 2, 3

[2] Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 295–305, 2022. 3

[3] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 1, 2, 3

[4] Wuyang Chen, Zhiding Yu, Shalini De Mello, Sifei Liu, Jose M. Alvarez, Zhangyang Wang, and Anima Anandkumar. Contrastive syn-to-real generalization. In *International Conference on Learning Representations*, 2021. 2, 3

[5] Sungha Choi, Sanghun Jung, Huiwon Yun, Joanne T Kim, Seungryong Kim, and Jaegul Choo. Robustnet: Improving domain generalization in urban-scene segmentation via instance selective whitening. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11580–11590, 2021. 1, 3, 4, 15, 16

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 6

[7] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020. 1, 2, 3, 4, 10

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is

[11] https://github.com/NVlabs/CSG/blob/main/LICENSE.md

[12] https://github.com/pytorch/pytorch/blob/master/LICENSE

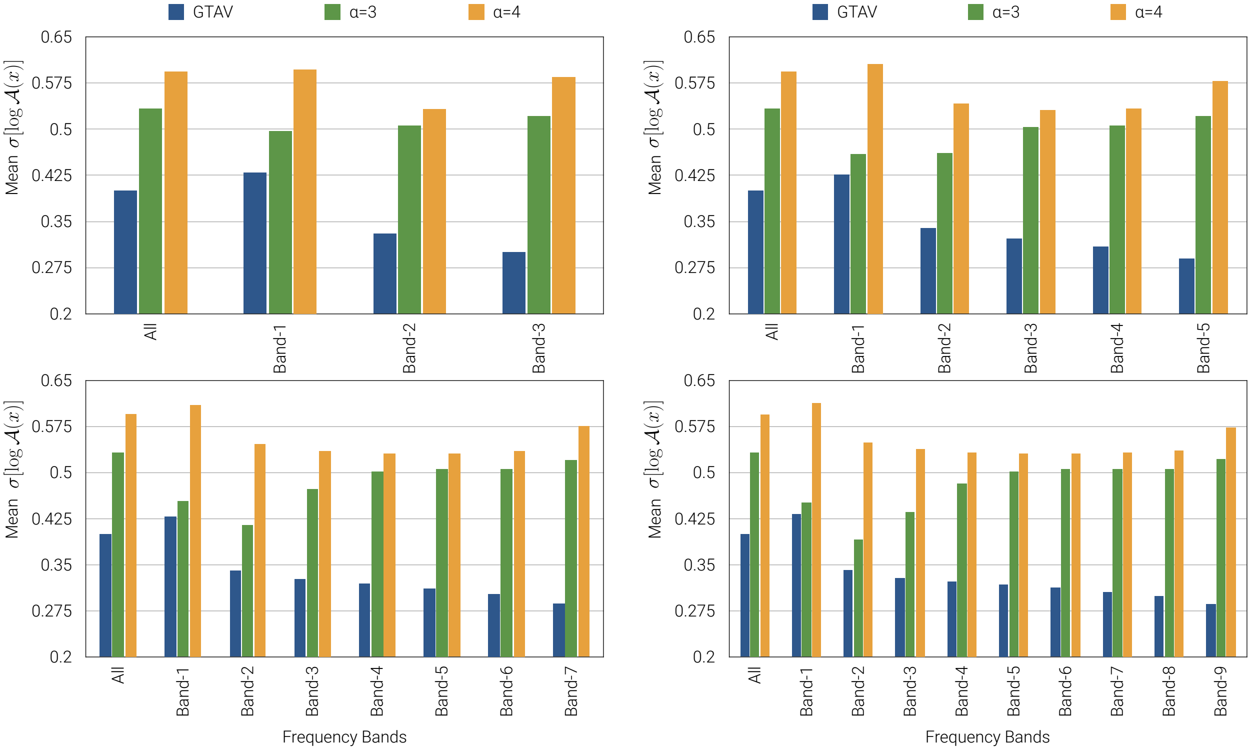

Figure 3: **Variations in amplitude values across fine-grained frequency bands for synthetic images post-PASTA (GTAV).** For GTAV, we find that applying PASTA increases variations in amplitude values across different frequency bands. Four plots correspond to fine-grained frequency bands (3, 5, 7 and 9 bands; increasing in frequency from Band-1 to Band-n). We find the maximum amount of increase for the highest frequency bands across different granularity levels.

worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 1, 2

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 2

[10] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Hrda: Context-aware high-resolution domain-adaptive semantic segmentation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXX*, pages 372–391. Springer, 2022. 2

[11] Matthew Johnson-Roberson, Charles Barto, Rounak Mehta, Sharath Nittur Sridhar, Karl Rosaen, and Ram Vasudevan. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, page 746–753. IEEE Press, 2017. 1, 6

[12] Vaishnavi Khindkar, Chetan Arora, Vineeth N Balasubramanian, Anbumani Subramanian, Rohit Saluja, and CV Jawahar. To miss-attend is to misalign! residual self-attentive feature alignment for adapting object detectors. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3632–3642, 2022. 1

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1, 2, 3

[14] Wei Liu, Andrew Rabinovich, and Alexander C Berg. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, 2015. 1

[15] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE international conference on computer vision*, pages 4990–4999, 2017. 6

[16] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 464–479, 2018. 1, 3, 4, 13, 14

[17] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information*

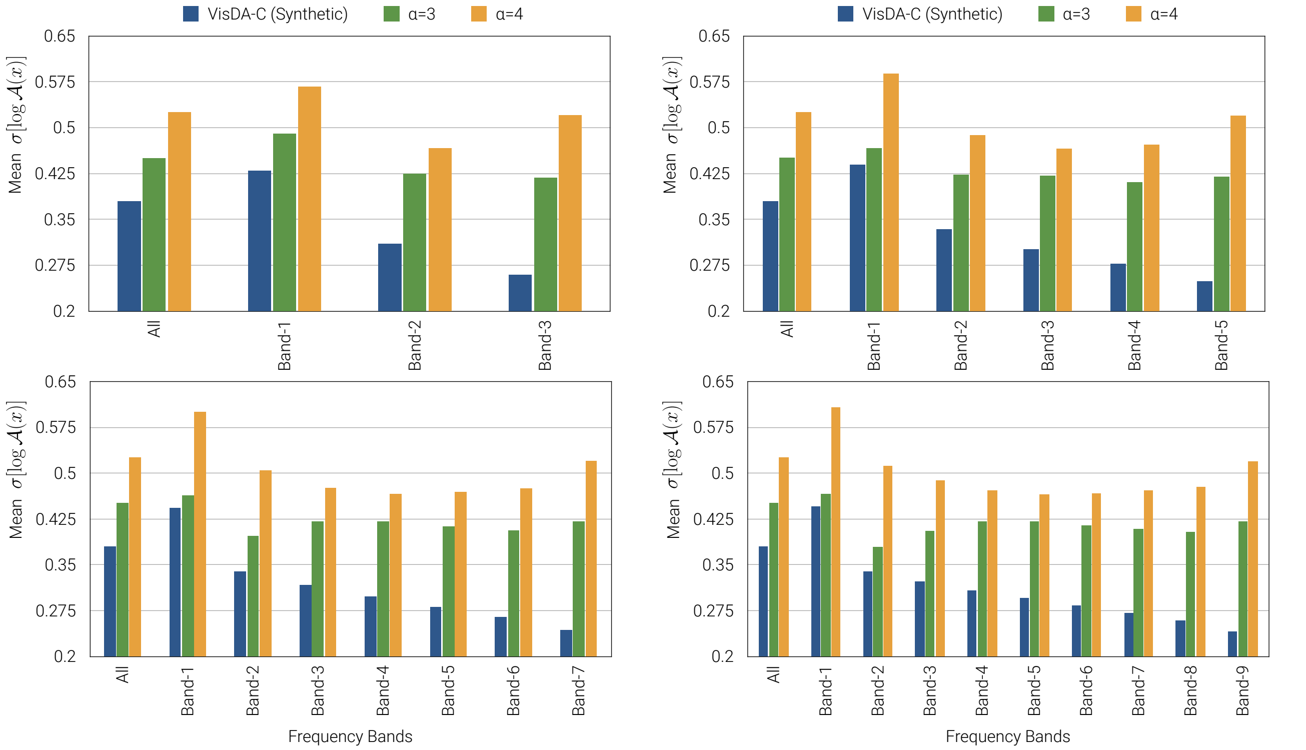

Figure 4: **Variations in amplitude values across fine-grained frequency bands for synthetic images post-PASTA (VisDA-C).** For VisDA-C (Synthetic), we find that applying PASTA increases variations in amplitude values across different frequency bands. Four plots correspond to fine-grained frequency bands (3, 5, 7 and 9 bands; increasing in frequency from Band-1 to Band-n). We find the maximum amount of increase for the highest frequency bands across different granularity levels.

Processing Systems 32, pages 8024–8035. Curran Associates, Inc., 2019. 7

[18] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017. 6

[19] 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*, 28, 2015. 1, 2

[20] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European conference on computer vision*, pages 102–118. Springer, 2016. 6

[21] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016. 3

[22] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 1, 2, 3

[23] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021. 2

[24] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020. 6

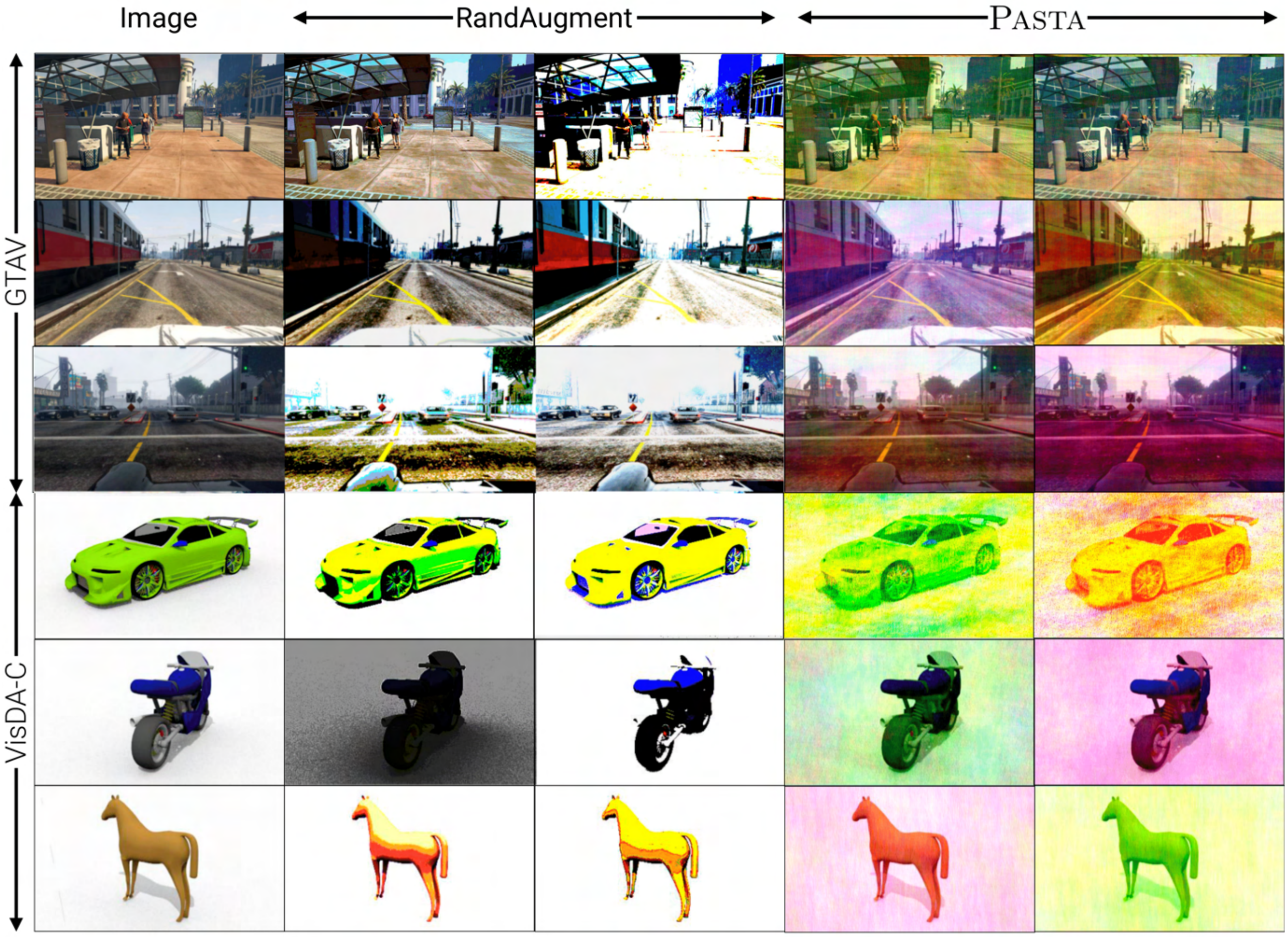

Figure 5: **PASTA augmentation samples.** Examples of images from different synthetic datasets when augmented using PASTA and RandAugment [7]. Rows 1-3 include examples from GTAV and rows 4-6 from VisDA-C.

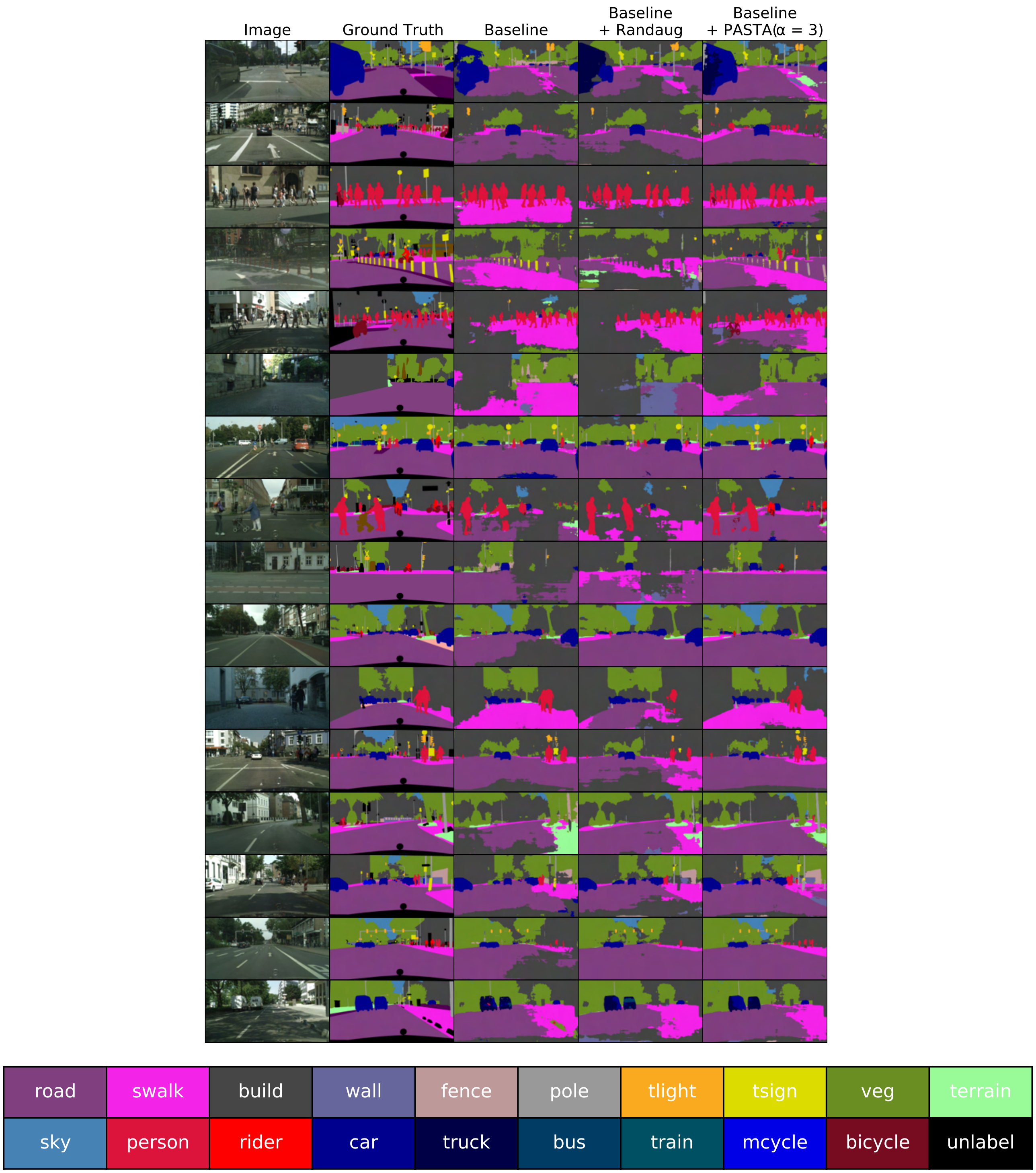

Figure 6: **GTAV→Cityscapes Baseline SemSeg Predictions.** Qualitative predictions made on randomly selected Cityscapes validation images by a Baseline DeepLabv3+ model (R-50 backbone) trained on GTAV synthetic images. The first two columns indicate the original image and the associated ground truth and rest indicate the listed methods.

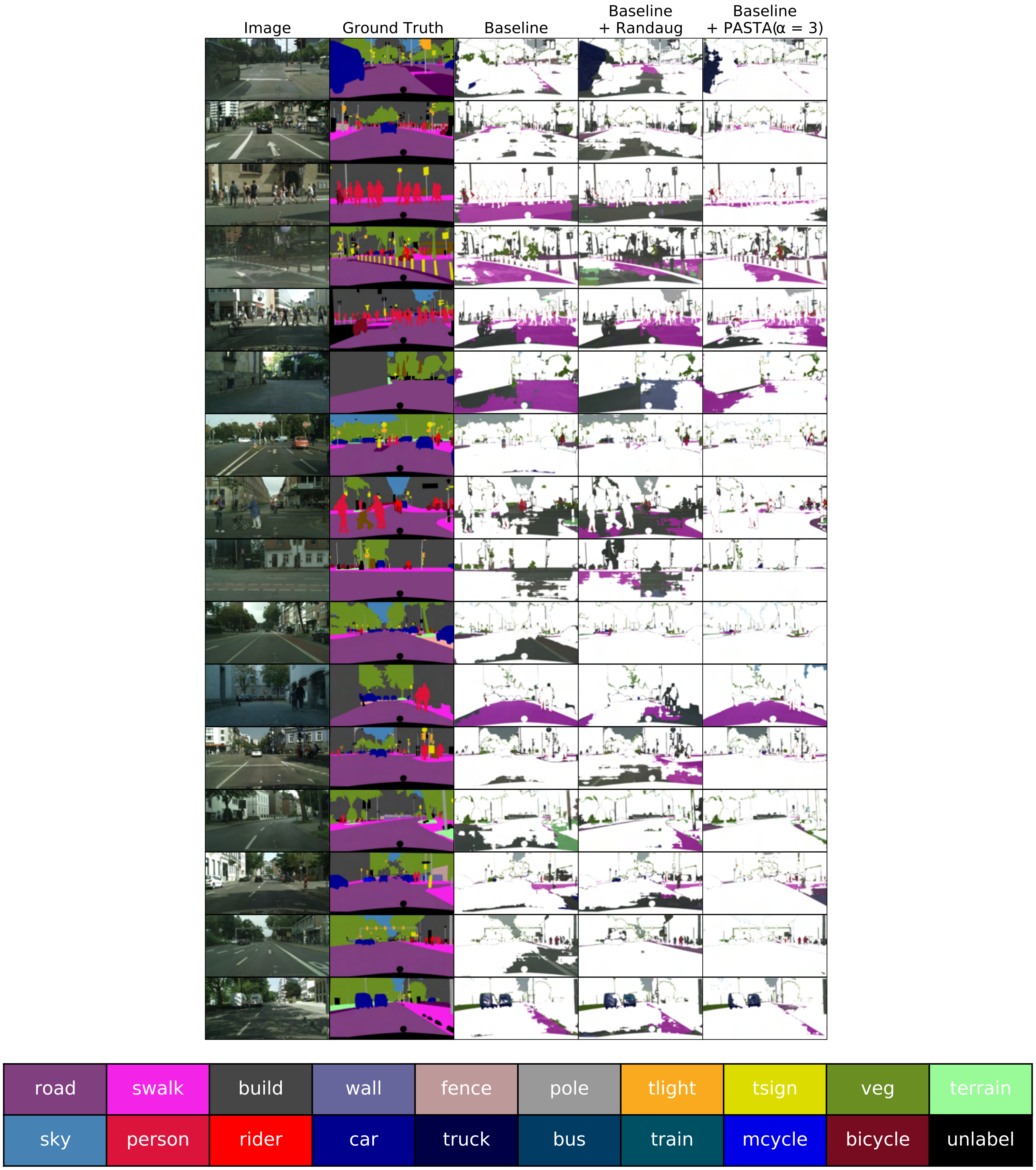

Figure 7: **GTAV→Cityscapes Baseline SemSeg Prediction Diffs.** Differences between prediction and ground truth for predictions made on randomly selected Cityscapes validation images by a Baseline DeepLabv3+ model (R-50 backbone) trained on GTAV synthetic images. The first two columns indicate the original image and the associated ground truth and rest indicate the listed methods.

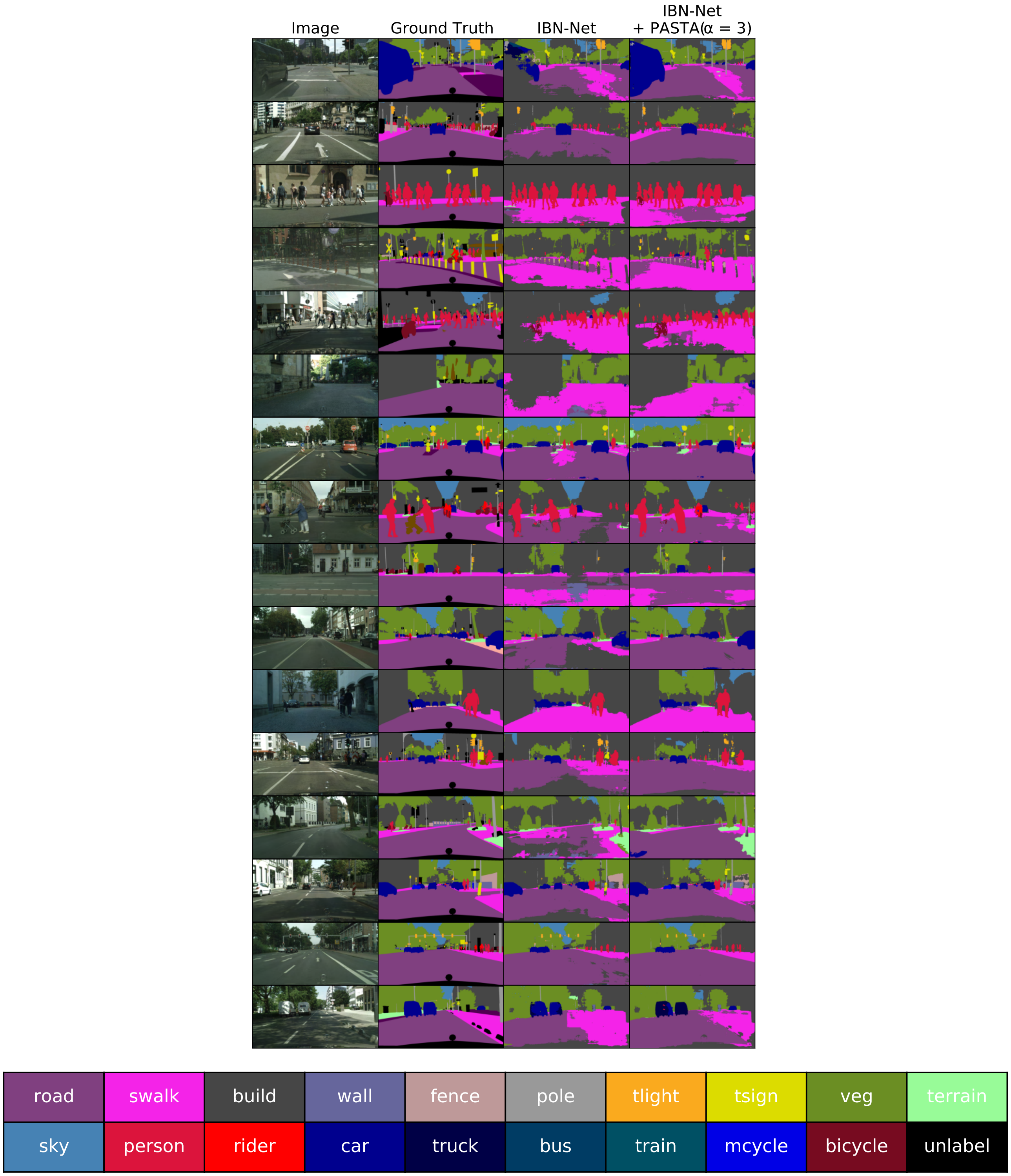

Figure 8: **GTAV→Cityscapes IBN-Net [16] SemSeg Predictions.** Qualitative predictions made on randomly selected Cityscapes validation images by IBN-Net (DeepLabv3+ model with R-50 backbone) trained on GTAV synthetic images. The first two columns indicate the original image and the associated ground truth and rest indicate the listed methods.

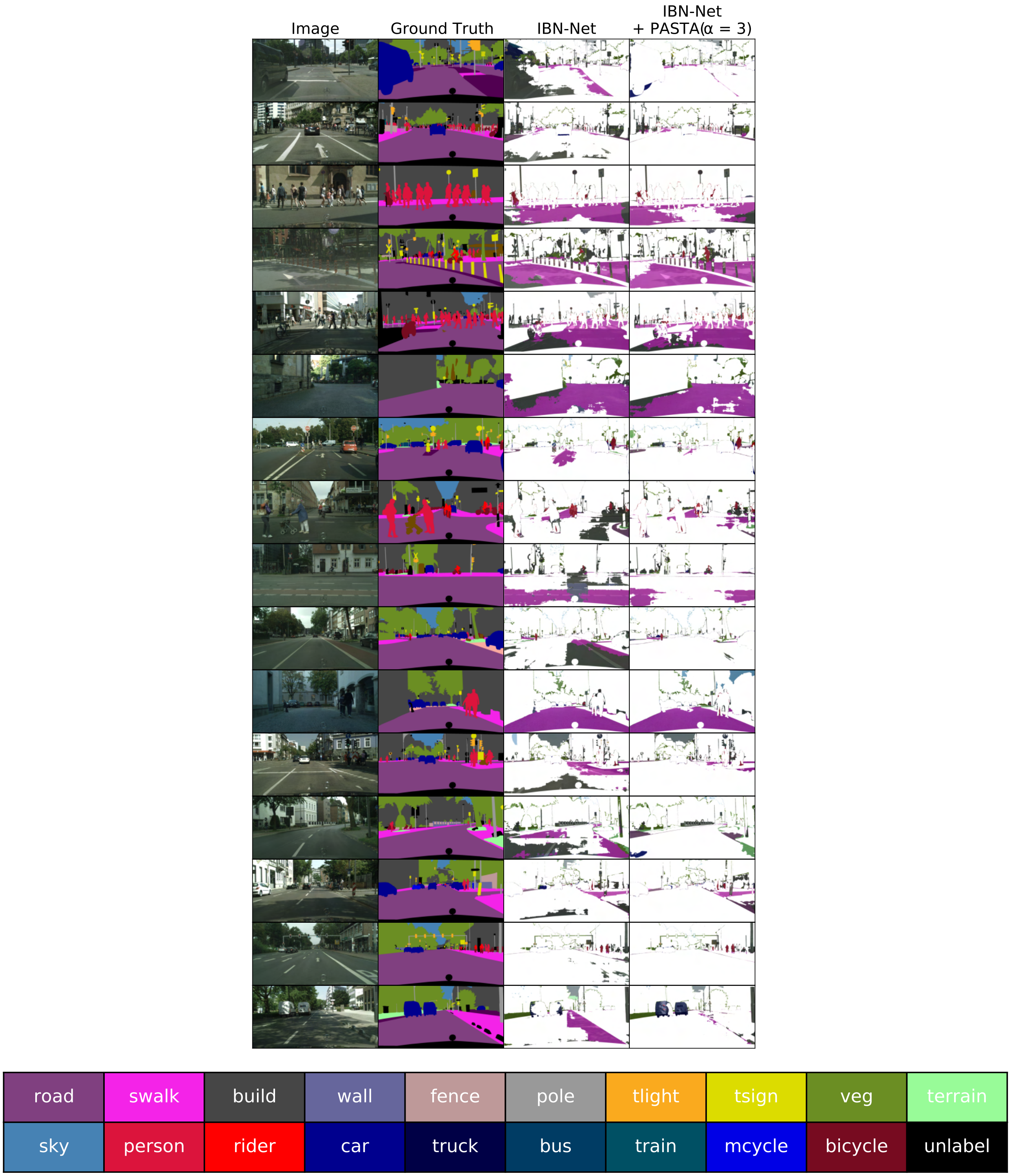

Figure 9: **GTAV→Cityscapes IBN-Net [16] SemSeg Prediction Diffs.** Differences between prediction and ground truth for predictions made on randomly selected Cityscapes validation images by IBN-Net (DeepLabv3+ model with R-50 backbone) trained on GTAV synthetic images. The first two columns indicate the original image and the associated ground truth and rest indicate the listed methods.

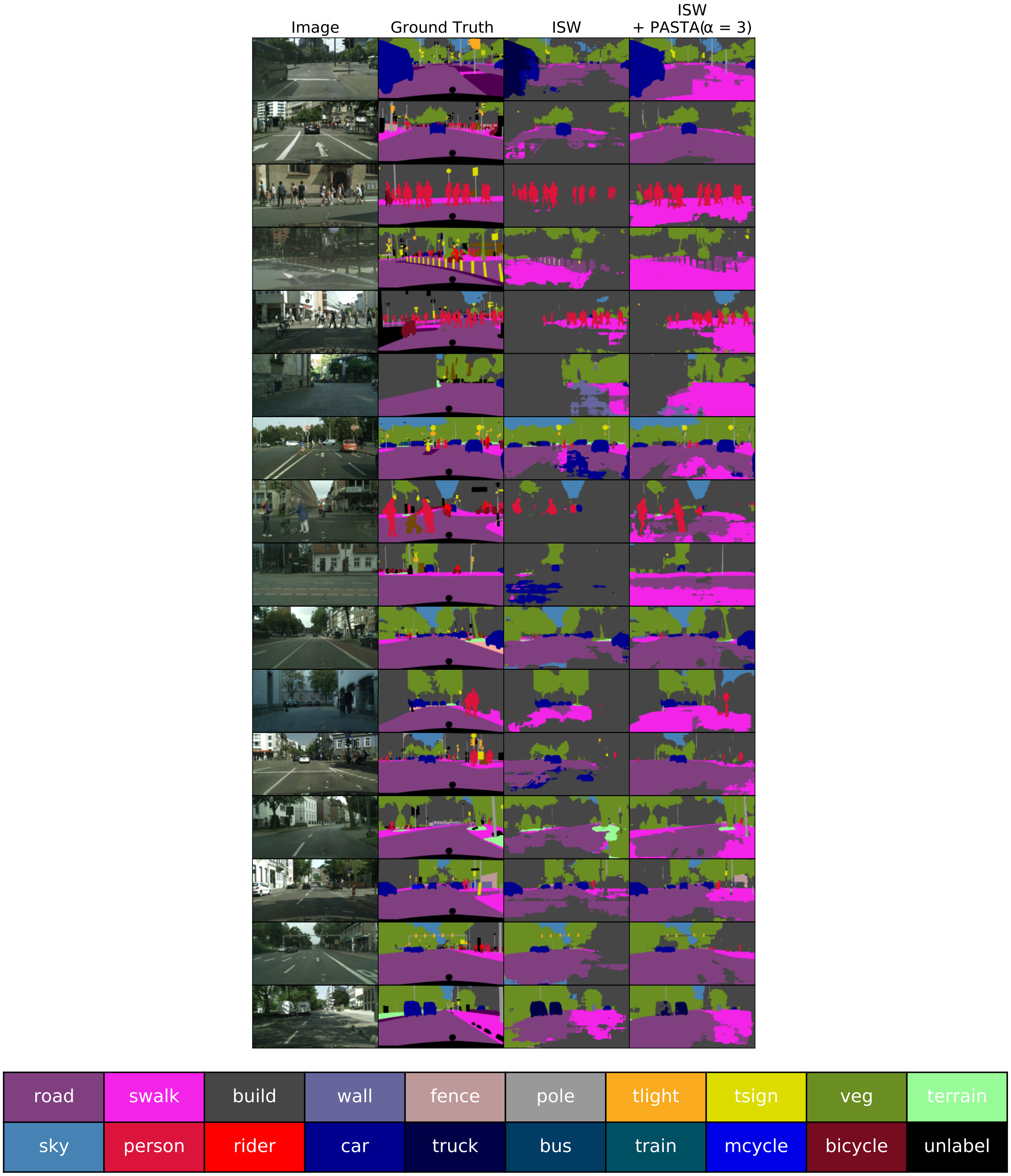

Figure 10: **GTAV→Cityscapes ISW [5] SemSeg Predictions.** Qualitative predictions made on randomly selected Cityscapes validation images by ISW (DeepLabv3+ model with R-50 backbone) trained on GTAV synthetic images. The first two columns indicate the original image and the associated ground truth and rest indicate the listed methods.

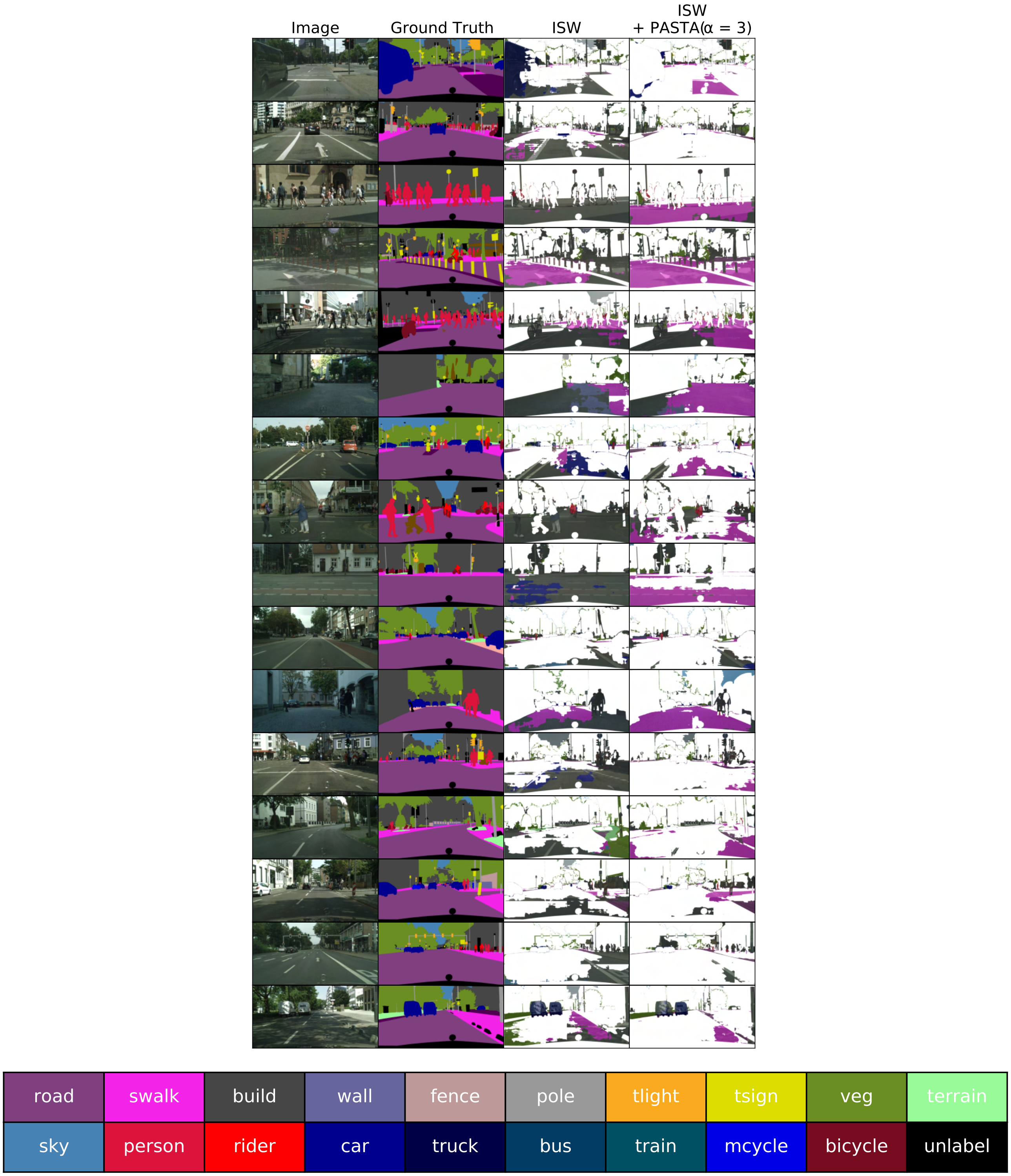

Figure 11: **GTAV→Cityscapes ISW [5] SemSeg Prediction Diffs.** Differences between prediction and ground truth for predictions made on randomly selected Cityscapes validation images by ISW (DeepLabv3+ model with R-50 backbone) trained on GTAV synthetic images. The first two columns indicate the original image and the associated ground truth and rest indicate the listed methods.