Stronger, Fewer, & Superior: Harnessing Vision Foundation Models for Domain Generalized Semantic Segmentation

Supplementary Material

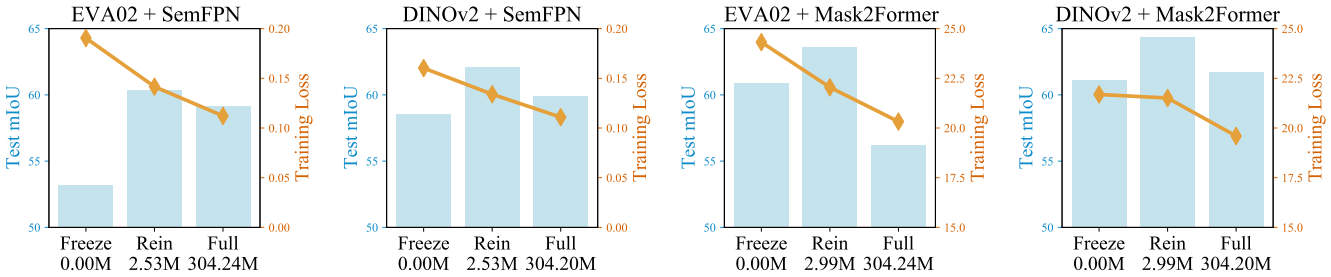

Figure 1. The curves of training loss and test metrics display consistent trends across different VFMs and decode heads: intuitively, as trainable parameters increase from $0.00M(Freeze) \rightarrow 2.53M(Rein) \rightarrow 304.24M(Full)$, the training loss monotonically decreases, indicating that a greater number of trainable parameters indeed better fit the training dataset. However, the test metrics on the target dataset initially rise and then fall, forming an inverted U-shape. This pattern suggests that the "Full" baseline overfits the training data, leading to diminished test performance. These findings are aligned with our motivation that **Leveraging Stronger pre-trained models and Fewer trainable parameters for Superior generalizability**. The blue bar charts in the figure represent the average mIoU tested on the Cityscapes, BDD100K, and Mapillary datasets, while the yellow line denotes the training loss during fine-tuning on GTAV dataset.

1. Fewer Trainable Parameters

Classical neural network theory [10, 12] points out that as model capacity increases, the empirical risk (or training risk) monotonically decreases, indicating an improved fit to training data. Conversely, the true risk (or test risk) typically exhibits a "U-shaped" curve, initially decreasing and then increasing, a phenomenon known as overfitting. From a modern viewpoint, the scaling law [15] suggests that on a smaller fixed dataset, performance stops to improve as model parameters increase, leading to overfitting.

In the majority of general tasks, the practice of early-stopping, based on evaluation data, can partly mitigate overfitting. However, in the field of domain generalization, the unknown test data distribution makes acquiring a valid evaluation dataset unavailable. Moreover, fine-tuning datasets are often smaller compared to ImageNet [7] or LVD-142M [24]. Hence, employing fewer trainable parameters emerges as a strategic approach to mitigate overfitting.

In our main paper, extensive experiments comprehensively demonstrate Rein's pivotal role in enhancing the generalization capabilities of VFMs. This enhancement may be attributed to two factors: 1) Rein's improved fitting capability for VFMs, ensuring better alignment with training data; 2) Rein's reduction of overfitting in VFMs during fine-tuning on smaller datasets, thus exhibiting enhanced generalization in testing. To delve into this, we analyze and compare the average training loss in the final 1000 iterations of the fine-tuning phase and their corresponding test metrics for various VFMs and decode heads.

Fig. 1 showcases a consistent trend across four different configurations. As trainable parameters increase from $0.00M(Freeze) \rightarrow 2.53M(Rein) \rightarrow 304.24M(Full)$, the training loss monotonically decreases. However, the test metrics on the target dataset peak with Rein, which employs 2.53 million parameters and incurs a sub-optimal training loss. In contrast, the "Full" baseline, despite recording the lowest training loss, only achieves sub-optimal test performance, a clear indicator of overfitting when compared to other setups. This observation aligns with the conclusions in [10, 15], supporting ours observation that **leveraging Stronger pre-trained models and Fewer trainable parameters can lead to Superior generalizability**.

Source Domain	Cityscapes mIoU
GTAV	66.4
+Synthia	68.1
+UrbanSyn	**78.4**
+1/16 of Cityscapes Training set	**82.5**

Table 1. Results on Cityscapes validation set.

2. Value of synthetic data

As Tab. 1 illustrates, trained on synthetic *UrbanSyn [11]+GTA+Synthia* datasets, Rein achieved a **78.4% mIoU** on the Cityscapes validation set. Further improvement is possible with additional synthetic data and higher-quality images generated by diffusion models, like [1]. This result can also be a valuable pre-trained weight for data-efficient training, reaching an 82.5% mIoU with 1/16 of

Rank r		4	8	16	32	64
Params		2.67M	2.77M	2.99M	3.42M	4.28M
EVA02 (Large) [8]	Citys	62.6	63.5	65.3	63.8	63.4
	BDD	58.5	58.9	60.5	60.5	60.2
	Map	63.7	63.8	64.9	64.5	64.3
	Avg.	61.6	62.1	**63.6**	62.9	62.7

Table 2. Ablation study on lora dim r.

VFMs	Method	Training Time	GPU Memory	Storage
EVA02 (Large)	Full	11.8 h	15.9 GB	1.22 GB
	Rein	10.5 h	12.5 GB	1.23 GB

Table 3. Training Time, GPU Memory, and Storage.

Cityscapes training set. This is a significant performance for semi-supervised semantic segmentation.

3. Ablation on decode head

Our experiments on Rein employ the Mask2Former [5] decode head, which shares structures or core concepts with numerous methods in dense prediction tasks [2, 4, 21, 23, 30]. The universality of Mask2Former highlights the significance of our findings for a range of segmentation tasks, including instance and panoptic segmentation. Furthermore, to demonstrate Rein's effectiveness in enhancing backbone generalization and its robustness across various decode heads, we conduct supplementary experiments using the popular SemFPN decode head [17], in the *GTAV*→ *Cityscapes* + *BDD100K* + *Mapillary* setting.

As shown in Table 4, Rein surpasses the "Full" and "Freeze" baselines, employing 2.53 million trainable parameters within the backbone, while the SemFPN decode head comprises 1.63 million parameters. Owing to the absence of object queries in SemFPN, the "linking tokens to instance" mechanism, described in Sec.3.3, is not utilized, resulting in a reduction of Rein's trainable parameters from 2.99 million to 2.53 million. When compared to the complete Rein configuration using the Mask2Former, using SemFPN achieves sub-optimal performance, evident in the 64.3% mIoU reported in Table 2 and 62.1% mIoU in Table 9, both implemented with DINOv2-Large. As shown in Table 5, the Mask2Former brings the 11.7% mIoU for ResNet101.These findings guide our decision to focus on experiments involving Mask2Former in the main paper.

4. Ablation on EVA02

Study on rank r As shown in Table 2, with EVA02 as the backbone, the optimal results are observed at $r = 16$.

Speed, memory, and storage. As shown in Table 3, compared to "Full" baseline, proposed Rein improves training speed and reduces GPU memory usage.

Backbone	Fine-tune Method	Trainable Params*	mIoU			
			Citys	BDD	Map	Avg.
EVA02 [8, 9] (Large)	Full	304.24M	58.5	56.9	62.0	59.1
	Freeze	0.00M	54.1	51.2	54.3	53.2
	Rein	2.53M	**61.4**	**58.5**	**62.0**	**60.7**
DINOv2 [24] (Large)	Full	304.20M	61.2	55.9	62.5	59.9
	Freeze	0.00M	58.9	56.4	60.3	58.5
	Rein	2.53M	**63.6**	**59.0**	**63.7**	**62.1**

Table 4. Performance Comparison with the proposed **Rein with SemFPN [17]** as Backbones under the *GTAV* → *Cityscapes (Citys)* + *BDD100K (BDD)* + *Mapillary (Map)* generalization setting. Models are fine-tuned on GTAV and tested on Cityscapes, BDD100K and Mapillary. The best results are **highlighted**. $*$ denotes trainable parameters in backbones.

Backbone	Decoder	Tune	mIoU
ResNet101 [20]	DeeplabV3plus	Full	34.4
ResNet101	Mask2Former	Full	46.1
DINOv2	Mask2Former	Full	61.7
DINOv2	Mask2Former	Ours	**64.3**

Table 5. Results on *GTAV*→ *Citys*+*BDD*+*Map*. Metrics for first line are from Wildnet.

Methods	Publication	mIoU			
		Citys	BDD	Map	Avg.
RobustNet [6]	CVPR 21	37.7	34.1	38.5	36.8
PintheMem [16]	CVPR 22	44.5	38.1	42.7	41.8
SAN-SAW [25]	CVPR 22	42.1	37.7	42.9	40.9
WildNet [20]	CVPR 22	43.7	39.9	43.3	42.3
DIGA [29]	CVPR 23	46.4	33.9	43.5	41.3
SPC [14]	CVPR 23	46.4	43.2	48.2	45.9
EVA02 - Frozen [8, 9]	arXiV 23	55.8	55.1	59.1	56.7
EVA02 + Rein	-	63.5	60.7	63.9	62.7
DINOv2 - Frozen [24]	arXiV 23	64.8	60.2	65.2	63.4
DINOv2 + Rein	-	**68.1**	**60.5**	**67.1**	**65.2**

Table 6. Performance Comparison of the proposed **Rein against other DGSS methods** under *GTAV* + *Synthia* → *Cityscapes (Citys)* + *BDD100K (BDD)* +*Mapillary (Map)* generalization.

5. Multi-source generalization.

In this part, we compare Rein against other DGSS methods under *GTAV* + *Synthia* → *Citys* + *BDD* + *Map* setting, in which networks are fine-tuned using both GTAV and Synthia datasets, and tested on Cityscapes, BDD100K, and Mapillary. As shown in Table 6, we report the performance of Rein employing two VFMs, EVA02 and DINOv2. Our results demonstrate that Rein significantly surpasses existing DGSS methods by a large margin in average mIoU (from 45.9% to 65.2%).

6. More details about VFMs

CLIP. In our study, we utilize the ViT-Large architecture, setting the patch size to 16×16. Each layer of this ar-

chitecture outputs features with a dimensionality of 1024, making use of the pre-trained weights from the foundational work [26]. Our model undergoes a pre-training phase through contrastive learning, employing publicly available image-caption data. This data is compiled through a blend of web crawling from select websites and integrating widely-used, existing image datasets. For the model's pre-trained weights, which have a patch size of 14×14 and an original pre-training image size of 224×224, we adopt bilinear interpolation to upscale the positional embeddings to a length of 1024. Moreover, trilinear interpolation is utilized to enlarge the kernel size of the patch embed layer to 16×16. Our model comprises 24 layers, and the features extracted from the 7th, 11th, 15th, and 23rd layers (counting from the zeroth layer) are subsequently channeled into the decoding head.

MAE. Employing the ViT-Large architecture, our model outputs features from each layer with a dimensionality of 1024, maintaining a patch size of 16×16. This model capitalizes on the pre-trained weights as delineated in the original work [13], and it undergoes self-supervised training using masked image modeling on ImageNet-1K. The architecture is composed of 24 layers, directing features from the 7th, 11th, 15th, and 23rd layers directly into the decoding head.

SAM. Aligning with the methodology described in the foundational paper [18], we employ the ViT-Huge architecture as our image encoder, making use of pre-trained weights that were trained on SA-1B [18] for a promptable segmentation task. The patch size of this model is set to 16×16, and each layer is designed to output features with a dimensionality of 1280, summing up to a total of 32 layers. The positional embeddings of the model are upscaled to a length of 1024 via bicubic interpolation. From this model, we extract features from the 7th, 15th, 23rd, and 31st layers and feed them into the decoder.

EVA02. In our approach, we adopt the largest scale configuration, EVA02-L, as our structural backbone, as suggested in the paper [8]. This particular model configuration determines its patch size as 16, with each layer producing feature maps of 1024 dimensions, across a total of 24 layers. EVA02 undergoes training through a combination of CLIP and Masked Image Modeling techniques on an aggregated dataset that includes IN-21K [7], CC12M [3], CC3M [28], COCO [22], ADE20K [31], Object365 [27], and OpenImages [19]. Mirroring the approach used in previous models, we upscale the positional embeddings to 1024 through bilinear interpolation, and the patch embed layer's convolutional kernel size is augmented to 16×16 via bicubic interpolation. Features from the 7th, 11th, 15th, and 23rd layers are then processed through the decode head.

DINOv2. Our choice of backbone for this study is DINOv2-L, which has been distilled from DINOv2-g. As noted in the original documentation, DINOv2-L occasionally surpasses the performance of DINOv2-g [24]. Sharing the same patch size, dimensionality, and layer count as EVA02-L, we apply equivalent processing to both the positional embeddings and patch embed layer of DINOv2-L. The features extracted from the 7th, 11th, 15th, and 23rd layers are subsequently fed into the decode head. DINOv2 is originally pretrained in a self-supervised fashion on the LVD-142M [24] dataset, following the procedures outlined in its respective paper.

VPT, LoRA, and AdaptFormer. Based on extensive experimentation, we have optimized the implementation of PEFT methods for DINOv2, utilizing configurations that enhance performance. These methods include: 1) VPT: It is deep and has 150 tokens. 2) LoRA: Applied to the query and value MLP components, LoRA is configured with a rank of 8. Additionally, it incorporates a minimal dropout rate of 0.1%. 3) AdaptFormer: This method employs a bottleneck design with a width of 64, initialized using LoRA. Notably, it omits layer normalization.

7. Algorithm of Proposed Rein

Algorithm 1 outlines the training procedure for Rein, wherein the weights conform to the constraints specified in Eq. (11). In this context, the variable c represents the number of channels in the feature maps of model $\mathcal{M}$, N denotes the total number of layers within $\mathcal{M}$, T indicates the overall number of training iterations, and r is defined as a hyperparameter that is considerably smaller than c.

8. Qualitative Results and Future works

In this section, we showcase our prediction results across various datasets, including Cityscapes, BDD100K, and Mapillary, as depicted in Fig.2, Fig.4, and Fig.3. All models are trained on the GTAV dataset without any fine-tuning on real-world urban-scene datasets. Our method outshines other approaches in accuracy, especially in categories like traffic signs, bicycles, traffic lights, sidewalks, roads, and trucks, demonstrating high precision for both large objects and smaller targets. Notably, despite not specifically optimizing for night-time segmentation, Rein's performance during night conditions is surprisingly high, almost akin to daytime performance, as illustrated in Fig.2.

With the rapid development of generative models research, we anticipate that our work could leverage high-quality generated samples to approach the performance of models trained with supervision on real datasets. Furthermore, we are prepared to investigate how VFMs can enhance the performance of semantic segmentation models trained on real datasets under various adverse weather conditions or on special road types. Finally, further exploration is necessary to investigate how Rein can be extended to

Algorithm 1: Training process of Rein.

Input: A sequence of input data and corresponding labels $\{(x_i, y_i) \mid t \in \mathbb{N}, 1 \leq i \leq N_d\}$; Pre-trained Vision Foundation Model $\mathcal{M}$, consisting of a patch embed layer L_{emb}, and layers $L_1, L_2, \ldots, L_N$; a decode head $\mathcal{H}$; and a proposed module Rein $\mathcal{R}$. The module Rein comprises the following matrices and vectors, initialized as specified:

$$
\begin{array}{ll}
A_i \in \mathbb{R}^{m \times r}, & \text{uniformly initialized,} \\
B_i \in \mathbb{R}^{r \times c}, & \text{uniformly initialized,} \\
W_{T_i} \in \mathbb{R}^{c \times c}, & \text{uniformly initialized,} \\
W_{f_i} \in \mathbb{R}^{c \times c}, & \text{initialized to zero,} \\
W_{Q_i} \in \mathbb{R}^{c \times c'}, & \text{uniformly initialized,} \\
b_{T_i} \in \mathbb{R}^{c}, & \text{initialized to zero,} \\
b_{f_i} \in \mathbb{R}^{c}, & \text{initialized to zero,} \\
b_{Q_i} \in \mathbb{R}^{c'}, & \text{initialized to zero,}
\end{array}
$$

for each $i \in \mathbb{N}, 1 \leq i \leq N$. Additionally, $W_Q \in \mathbb{R}^{3c' \times c'}$ is uniformly initialized, and $b_Q \in \mathbb{R}^{c'}$ is initialized to zero.

Output: The optimized $\mathcal{H}$ and $\mathcal{R}$.

for $t \leftarrow 1$ **to** T **do**

- Get batch data:(x, y)
- $f_0 = L_{\text{emb}}(x)$
- **for** $i \leftarrow 1$ **to** N **do**
  - $f_i = L_i(f_{i-1})$
  - $T_i = A_i \times B_i$
  - $S_i = Softmax(\frac{f_i \times T_i^{\mathrm{T}}}{\sqrt{c}})$
  - $\Delta \bar{f}_i = S_i(:, 2:m) \times [T_i(2:m) \times W_{T_i} + b_{T_i}]$
  - $\Delta f_i = (\Delta \bar{f}_i + f_i) \times W_{f_i} + b_{f_i}$
  - $Q_i = T_i \times W_{Q_i} + b_{Q_i}$
  - $f_i = f_i + \Delta f_i$
- $\mathbb{F}_t \subseteq \{f_0, f_1, \ldots, f_N\}$
- Calculate Q_{max} and Q_{avg} by Eq. (9)
- $Q = Concat([Q_{max}, Q_{avg}, Q_N]) \times W_Q + b_Q$
- $\bar{y_t} = \mathcal{H}(\mathbb{F}_t, Q)$
- Optimize $\mathcal{H}$ and $\mathcal{R}$ by $\mathcal{L}oss(\bar{y}, y)$

tasks such as instance segmentation, panoptic segmentation, open-vocabulary segmentation, and even object detection.

Figure 2. Prediction results of DINOv2+Rein on the BDD100K validation set. The model is fine-tuned exclusively on the GTAV dataset, without access to any real-world urban-scene datasets.

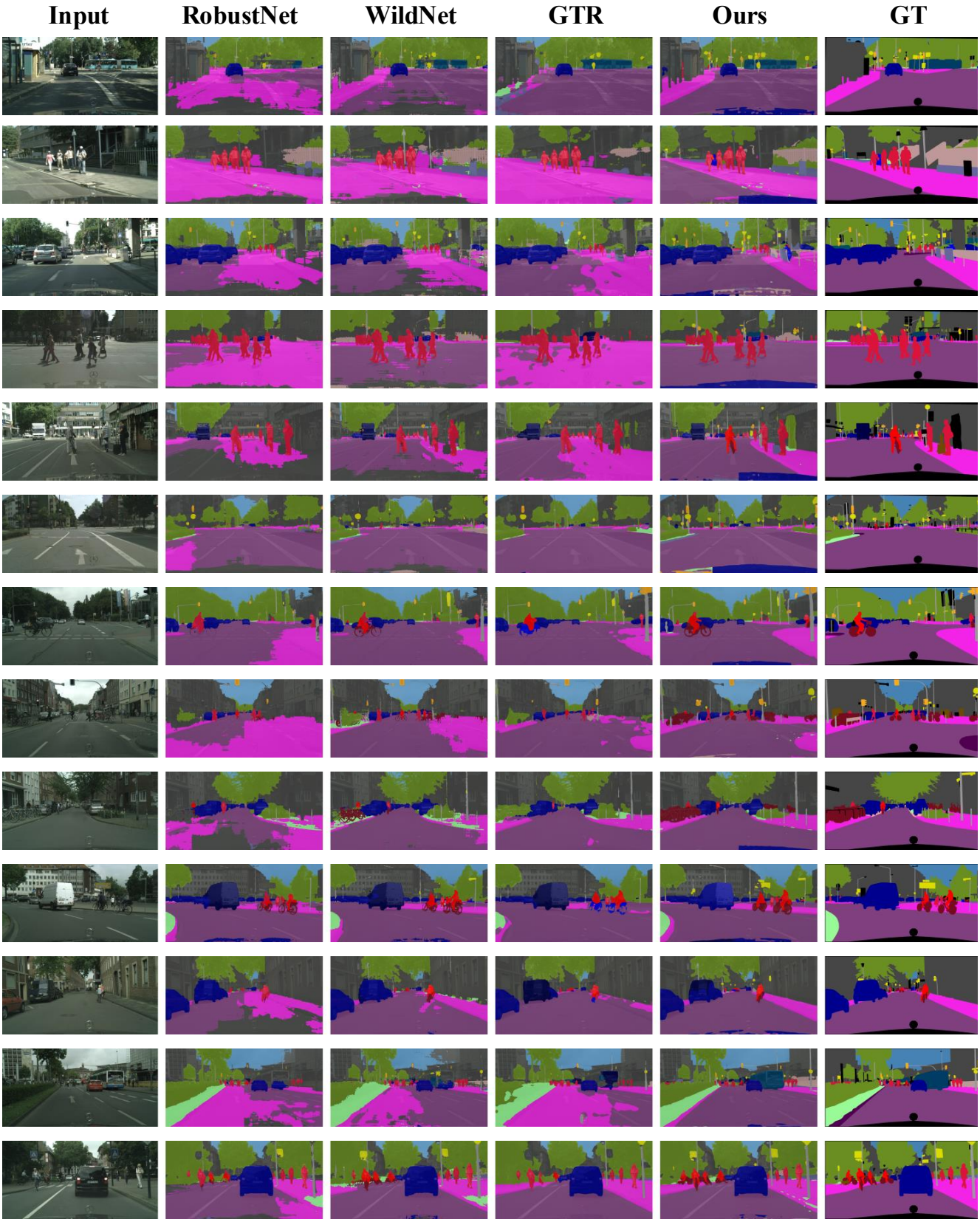

Figure 3. Prediction results of DINOv2+Rein on the Cityscapes validation set. The model is fine-tuned exclusively on the GTAV dataset, without access to any real-world urban-scene datasets.

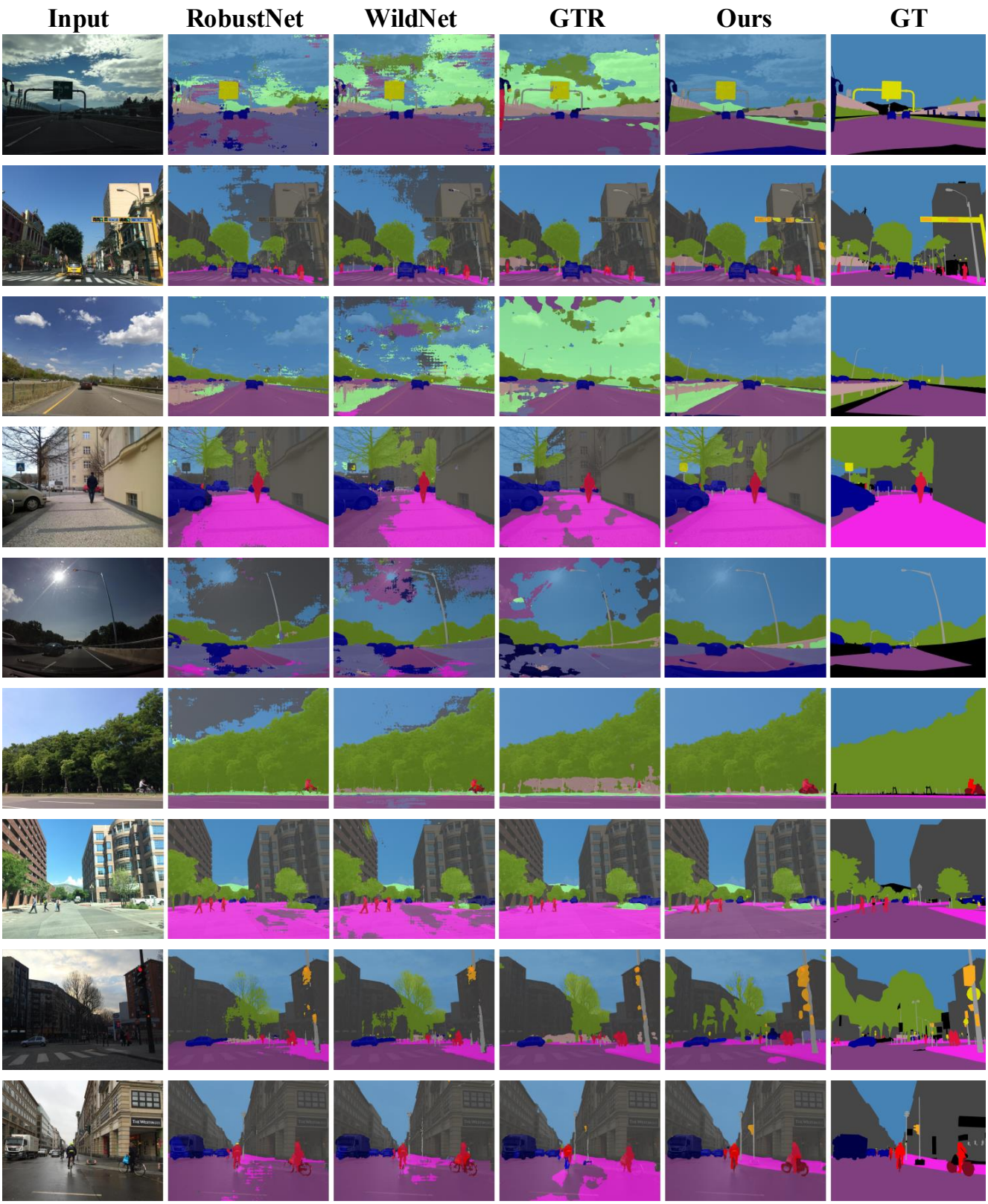

Figure 4. Prediction results of DINOv2+Rein on the Mapillary validation set. The model is fine-tuned exclusively on the GTAV dataset, without access to any real-world urban-scene datasets.

References

[1] Yasser Benigmim, Subhankar Roy, Slim Essid, Vicky Kalogeiton, and Stéphane Lathuilière. Collaborating foundation models for domain generalized semantic segmentation, 2023. 1

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 2

[3] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3558–3568, 2021. 3

[4] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, 34:17864–17875, 2021. 2

[5] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022. 2

[6] 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. 2

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1, 3

[8] Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva-02: A visual representation for neon genesis. *arXiv preprint arXiv:2303.11331*, 2023. 2, 3

[9] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19358–19369, 2023. 2

[10] Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58, 1992. 1

[11] Jose L. Gómez, Manuel Silva, Antonio Seoane, Agnès Borrás, Mario Noriega, Germán Ros, Jose A. Iglesias-Guitian, and Antonio M. López. All for one, and one for all: Urbansyn dataset, the third musketeer of synthetic driving scenes, 2023. 1

[12] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009. 1

[13] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 3

[14] Wei Huang, Chang Chen, Yong Li, Jiacheng Li, Cheng Li, Fenglong Song, Youliang Yan, and Zhiwei Xiong. Style projected clustering for domain generalized semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3061–3071, 2023. 2

[15] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 1

[16] Jin Kim, Jiyoung Lee, Jungin Park, Dongbo Min, and Kwanghoon Sohn. Pin the memory: Learning to generalize semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4350–4360, 2022. 2

[17] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6399–6408, 2019. 2

[18] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, 2023. 3

[19] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, 128(7):1956–1981, 2020. 3

[20] Suhyeon Lee, Hongje Seong, Seongwon Lee, and Euntai Kim. Wildnet: Learning domain generalized semantic segmentation from the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9936–9946, 2022. 2

[21] Feng Li, Hao Zhang, Huaizhe Xu, Shilong Liu, Lei Zhang, Lionel M. Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3041–3050, 2023. 2

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 3

[23] Rodrigo Marcuzzi, Lucas Nunes, Louis Wiesmann, Jens Behley, and Cyrill Stachniss. Mask-based panoptic lidar segmentation for autonomous driving. *IEEE Robotics and Automation Letters*, 8(2):1141–1148, 2023. 2

[24] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez,

Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 1, 2, 3

[25] Duo Peng, Yinjie Lei, Munawar Hayat, Yulan Guo, and Wen Li. Semantic-aware domain generalized segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2594–2605, 2022. 2

[26] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3

[27] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019. 3

[28] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018. 3

[29] Wei Wang, Zhun Zhong, Weijie Wang, Xi Chen, Charles Ling, Boyu Wang, and Nicu Sebe. Dynamically instance-guided adaptation: A backward-free approach for test-time domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24090–24099, 2023. 2

[30] Bowen Zhang, Zhi Tian, Quan Tang, Xiangxiang Chu, Xiaolin Wei, Chunhua Shen, et al. Segvit: Semantic segmentation with plain vision transformers. *Advances in Neural Information Processing Systems*, 35:4971–4982, 2022. 2

[31] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127:302–321, 2019. 3