Supplementary Material for "MatchED: Crisp Edge Detection Using End-to-End, Matching-based Supervision"

Contents

S1. Reasons for Edge Thickness

Both traditional and learning-based edge detection methods tend to produce thick raw edge predictions. This behavior arises from three primary and compounding factors: (i) annotation misalignment, (ii) merged supervision, and (iii) inherent image gradients.

Annotation Misalignment: Human annotators inevitably introduce small spatial inconsistencies when manually tracing edges in the scene. These pixel-level deviations prevent perfect alignment between RGB images and their ground-truths. As a result, during training, the supervision signal is spatially dispersed rather than concentrated on a single pixel. Consequently, models learn to activate not only the true boundary pixel but also its neighboring pixels, leading to thicker predictions.

Merged Supervision: Datasets such as BSDS [1] and MultiCue [6] provide multiple annotations per image. Annotators often disagree not only on precise boundary localization but also on the semantic validity of certain boundaries. When these differing annotations are aggregated into a single ground-truth map, the resulting supervision inherently broadens in spatial extent. Training with such thick supervision signals encourages models to produce similarly thick raw predictions.

Inherent Image Gradients: In natural RGB images, transitions between distinct regions rarely occur as ideal one-pixel step edges. Due to optical blur, illumination changes, sensor sampling, and other imaging factors, physical boundaries typically appear as gradual intensity transitions spanning multiple pixels. Therefore, even prior to annotation, the underlying image gradients already exhibit spatial thickness, further contributing to thick edge responses in learned models.

MATCHED explicitly addresses and mitigates all three factors, enabling the model to produce spatially precise and significantly thinner edge predictions.

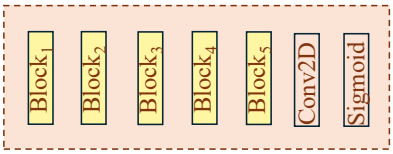
S2. Time Complexity of Our Matching Algorithm

In this section, we provide a detailed analysis of the time complexity involved in generating the proposed matching-based supervision. First, computing the cost matrix C in Eq. 3 of the main manuscript requires comparing all pixels in the ground-truth G and the predicted crisp edge map $\mathbf{E}^c$. Therefore, the time complexity of this step is $\mathcal{O}\big((W \cdot H)^2\big)$, where W and H denote the width and height of $\mathbf{E}^c$. Then, solving the bipartite matching problem in Eq. 4 of the main manuscript using the linear sum assignment algorithm has a cubic time complexity of $\mathcal{O}\big((N_G \cdot N_{\mathbf{E}^c})^3\big)$. Finally, recovering unmatched ground-truth edges in Eq. 6 of the main manuscript requires iterating over all pixels in the ground-truth $\mathbf{G}$, and it has a time complexity of $\mathcal{O}\big((W \cdot H)\big)$.

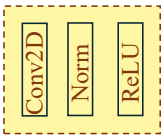
Consequently, the total time complexity of the proposed matching-based supervision algorithm is

$$\mathcal{O}\big((W \cdot H)^2 + (N_G \cdot N_{\mathbf{E}^c})^3 + (W \cdot H)\big). \tag{S1}$$

The dominant term is the cubic cost of the linear sum assignment. In practice, $N_{\mathbf{E}^c}$ can be large in the initial iterations due to noisy and thick edge predictions. However, as



(a) The architecture of lightweight CNN.



(b) The architecture of Block.

Figure S1. (a) A lightweight CNN consists of five blocks followed by a final Conv2D layer with a sigmoid activation. Therefore, it has only approximately 21K parameters. (b) Each block contains Conv2D, ReLU, and normalization layers, where the type of normalization (e.g., BatchNorm, LayerNorm) is chosen to match that used in the base edge detector.

the model learns a more accurate and crisp edge map, $N_{\mathbf{E}^c}$ decreases significantly, thereby reducing the computational cost in subsequent iterations.

S3. Details of Light-weight CNN

As shown in Figure S1(a), a light-weight CNN has five blocks and one Conv2D layer with sigmoid activation. Each block (Figure S1(b)) has one Conv2D layer with ReLU activation and one normalization layer. The normalization layer type matches that of the base edge detector. For example, PiDiNet [11] + MATCHED uses BatchNorm2D, whereas SAUGE [5] + MATCHED uses LayerNorm. In total, it introduces only ∼21k parameters.

S4. Hyperparameter Configurations

Hyperparameter settings for each base model and dataset are presented in Table S1. We tuned the β coefficient, the loss weight of MATCHED in Eq. 8 of the main manuscript, only for RankED [2], due to its grid-like artifacts [13] and down-sampled predictions. To mitigate the performance degradation caused by these two factors, we performed a dedicated search for β only on RankED.

For tuning the parameters α (the weight of the confidence score) and τ_c (the confidence threshold used to select edge pixels) in Eq. 3 of the main manuscript, our heuristic is that less noisy and crisper raw edge predictions benefit from a larger τ_c (e.g., 0.1) and a smaller τ_c (e.g., 5).

S5. Results on Multi-cue

Since Multi-Cue dataset lacks an official split, results are reported as the mean and variance over three random splits, following [2, 11, 16], which results in longer training times compared to other datasets. Due to resource constraints, MATCHED is integrated into only two models, (i) PiDiNet [11] and (ii) RankED [2], and their performance is compared with SOTA in Table S2. Additionally, due to the lack of official models for Multi-Cue, SOTA methods are reported with post-processing, whereas our method is evaluated without post-processing.

When integrated into PiDiNet [11], our method yields improvements of +4.8 and +5.1 in ODS and OIS, respectively, for the edge part, and scores of +0.7 and -0.1 for the boundary part. As discussed in Section 4.3 of the main manuscript, the down-scaled feature map size and interpolation artifacts cause the performance of MATCHED to drop to that of RankED [2] when compared to the post-processed results.

Table S1. Hyperparameter configurations used for integrating MATCHED into each base model across NYUD-v2 [9], BSDS [1], Multi-Cue [6], and BIPED [10] datasets. τ_c is the confidence threshold for selecting edge pixels in Eq. 4 of the main manuscript, α is the weight of the confidence score in Eq. 3 of the main manuscript, and β is the loss coefficient of MATCHED.

Model	Dataset	$\tau_c(\times 10^{-2})$	α	β
RankED [2]	NYUD	1	25	5
	BSDS	1	25	5
	Multi-Cue	1	25	5
	BIPED	–	–	–
PiDiNet [11]	NYUD	1	20	1
	BSDS	1	25	1
	Multi-Cue	1	55	1
	BIPED	1	55	1
DiffusionEdge [14]	NYUD	10	5	1
	BSDS	10	5	1
	Multi-Cue	–	–	–
	BIPED	10	5	–
SAUGE [5]	NYUD	5	20	1
	BSDS	5	20	1
	Multi-Cue	–	–	–
	BIPED	–	–	–

S6. Effects of hyperparameter

Effect of confidence threshold τ_c. We investigate the effects of different confidence thresholds τ_c in Eq. 3 of the main manuscript on the performance of MATCHED, using PiDiNet [11] on BSDS dataset, as shown in Table S3(a). According to the results, the best performance is achieved when nearly all predictions (i.e., pixels with confidence ¿ 0.01) are included in the matching process. Threshold values up to 0.1 yield similar performance, except for the AP metric. Excluding low-confidence pixels results in a lower AP score. Nevertheless, our method still outperforms standard post-processing for threshold values up to 0.2. During these experiments, the value of the confidence score α and distance threshold τ_d are used as 25 and 4, respectively.

Table S2. Comparison of SOTA results on Multi-Cue dataset. While MATCHED results are reported under CEval, other results are reported under SEval. ODS: Optimal Dataset Score, OIS: Optimal Image Score, AP: Average Precision, AC: Average Crispness. Best and second-best results are shown in bold and underlined, respectively.

Edge					
Method		**ODS**	**OIS**	**AP**	
Human			.750±.024	–	–
Multicue [6]	(VR'16)	.830±.002	–	–	
HED [12]	(ICCV'15)	.851±.014	.864±.011	–	
RCF [4]	(CVPR'17)	.857±.004	.862±.004	–	
BDCN [3]	(CVPR'19)	.891±.001	.898±.002	.935±.002	
PiDiNet [11]	(ICCV'21)	.855±.007	.860±.005	–	
EDTER [7]	(CVPR'22)	.894±.005	.900±.003	.944±.002	
UAED [15]	(CVPR'23)	.895±.002	.902±.001	.949±.002	
MuGE [16]	(CVPR'24)	.898±.004	.900±.004	.950±.004	
Diff.Edge [14]	(AAAI'24)	.904	.909	–	
RankED [2]	(CVPR'24)	**.962±.003**	**.965±.003**	**.973±.006**	
PiDiNet + MATCHED		.903±.002	.911±.004	.973±.007	
RankED + MATCHED		.924±.010	.929±.010	.969±.007	
Boundary					
Method		**ODS**	**OIS**	**AP**	
Human			.760±.017	–	–
Multicue [6]	(VR'16)	.720±.014	–	–	
HED [12]	(ICCV'15)	.814±.011	.822±.008	.869±.015	
RCF [4]	(CVPR'17)	.817±.004	.825±.005	–	
BDCN [3]	(CVPR'19)	.836±.001	.846±.003	.893±.001	
PiDiNet [11]	(ICCV'21)	.818±.003	.830±.005	–	
EDTER [7]	(CVPR'22)	.861±.003	.870±.004	.919±.003	
UAED [15]	(CVPR'23)	.864±.004	.872±.006	.927±.006	
MuGE [16]	(CVPR'24)	.875±.006	.879±.006	.932±.006	
RankED [2]	(CVPR'24)	**.963±.002**	**.967±.002**	**.995±.001**	
PiDiNet + MATCHED		.825±.003	.829±.002	.889±.003	
RankED + MATCHED		.941±.009	.946±.008	**.995±.006**	

Effect of distance threshold τ_d. We analyze how varying the distance threshold τ_d in Eq. 3 of the main manuscript impacts the performance of MATCHED on BSDS dataset using PiDiNet [11], as presented in Table S3(b). According to the results, all tested threshold values yield similar performance, except for $\tau_d = 2$ in the AP metric, which shows a 1.0-point lower score. Notably, the best performance is achieved with $\tau_d = 4$, which aligns with the distance threshold used during evaluation. During these experiments, the weight of the confidence score α and the confidence threshold τ_c are used as 25 and 0.01, respectively.

Effect of confidence score's weight α. We examine the impact of varying the confidence score weight α in Eq. 3 of the main manuscript on the performance of MATCHED on BSDS dataset, when integrated into PiDiNet [11], as shown in Table S3(c). The results show that $\alpha = 5$ leads to a significant performance drop across all metrics. The best performance is achieved with $\alpha = 25$, while $\alpha = 30$ yields nearly the same results. Moreover, using $\alpha = 15$ or 20 still outperforms standard post-processing methods. During these experiments, the distance threshold τ_d and the confidence threshold τ_c are used as 2 and 0.01, respectively.

Heuristic for parameter choice. Our experiments indicate that tuning the distance threshold τ_d is unnecessary, as it has minimal impact on performance and works well when aligned with the evaluation threshold.

For the τ_c parameter, there is a trade-off between training time and performance: increasing τ_c reduces training time because fewer pixels are included in the matching process (see Section S2), but slightly lowers performance. For lower computational cost, τ_c can be set in the range 0.1–0.2; for optimal performance, all pixels can be used, i.e., $\tau_c = 0.01$. For crisp and accurate raw edge predictions, such as [14], setting $\tau_c = 0.1$ yields good performance.

The α parameter should be small for crisp raw results (e.g., $\alpha = 5$) and larger for noisy or thick raw results (e.g., $\alpha = 20$).

S7. GPU memory overhead

Table S4 presents the GPU memory overhead introduced by MATCHED under varying image resolutions and confidence thresholds τ_c. As expected, memory consumption increases substantially with larger input sizes, rising from around 1.7 GB for 80×80 inputs to over 28 GB for 320×320 inputs. Moreover, increasing the confidence threshold τ_c leads to decreasing GPU memory usage, as a higher τ_c reduces the number of edge pixels involved in the matching process. Note that the memory consumption is sensitive to the number of edge pixels in the ground-truth as well as the number of pixels exceeding the confidence threshold τ_c. As these numbers increase, the required GPU memory grows accordingly.

S8. More Visual Results

This section presents visual results on the NYUD, BSDS, BIPED, and Multi-Cue datasets. To enable a fair comparison for each base edge detector with an officially released checkpoint (on NYUD and on BSDS), we report two types of raw visual results:

- **Official Results**: Edge maps produced directly from the publicly released model weights. These serve as the canonical reference for each base model.
- **MATCHED's Pipeline**: Raw edge maps produced by the same base model when used within the MATCHED's training pipeline. These results serve as the proper baseline for visually assessing how MATCHED refines the raw predictions.

Minor visual and numerical discrepancies between these two outputs are expected. Such differences arise from (i) non-determinism in modern training pipelines (e.g., random initialization and data loading order), and (ii) the influence

Table S3. Effect of (a) confidence threshold τ_c, (b) distance threshold τ_d, and (c) confidence score weight α on BSDS dataset using PiDiNet. The baseline (PiDiNet) performance is ODS = .789 and OIS = .803.

(a) Confidence threshold τ_c				**(b) Distance threshold τ_d**				**(c) Confidence score weight α**			
Param.	**ODS**	**OIS**	**AP**	**Param.**	**ODS**	**OIS**	**AP**	**Param.**	**ODS**	**OIS**	**AP**
0.01	**.800**	**.811**	**.866**	2	.797	.807	.856	5	.630	.639	.653
0.03	.799	.811	.858	4	**.800**	**.811**	**.866**	10	.783	.790	.827
0.05	.799	.811	.845	6	.797	.809	**.866**	15	.794	.802	.845
0.10	.799	.808	.836	8	.797	.808	**.866**	20	.798	**.811**	.857
0.20	.788	.803	.820	—	—	—	—	25	**.800**	**.811**	.866
0.30	.761	.771	.803	—	—	—	—	30	.798	**.811**	**.867**

Table S4. GPU memory overhead introduced by MATCHED at different image sizes and confidence thresholds τ_d.

Image Size	τ_d	**GPU Memory (GB)**
80×80	0.01	1.72
	0.05	1.65
	0.10	1.57
320×320	0.01	28.43
	0.05	27.30
	0.10	25.86

of MATCHED's loss gradients on the base model during an end-to-end optimization.

Across all datasets and base models, the visual results demonstrate that MATCHED consistently generates superior crisp edge maps. Notably, MATCHED successfully achieves thinning without altering the thickness of the raw input edge map.

S8.1. More Visual Results on NYUD

In this section, we present visual results of MATCHED on NYUD-v2 [8] dataset using the four base models: (i) DiffusionEdge [14] (Figure S2), (ii) PiDiNet [11] (Figure S3), (iii) RankED [2] (Figure S4), and SAUGE [5] (Figure S5). Each figure compares the raw outputs from official model checkpoints, the outputs from the MATCHED pipeline before and after NMS, and the outputs of MATCHED integrated versions.

S8.2. More Visual Results on BSDS

In this section, we present visual results of MATCHED on BSDS [1] dataset using the four base models: (i) DiffusionEdge [14] (Figure S6), (ii) PiDiNet [11] (Figure S7), (iii) RankED [2] (Figure S8), and SAUGE [5] (Figure S9). Each figure compares the raw outputs from official model checkpoints, the outputs from the MATCHED pipeline before and after NMS, and the outputs of MATCHED integrated versions.

S8.3. More Visual Results on BIPED

In this section, we present visual results of MATCHED on BIPED-v2 [10] dataset using the two base models: (i) DiffusionEdge [14] (Figure S10) and (ii) PiDiNet [11] (Figure S11). Each figure compares the raw outputs from the MATCHED pipeline before and after NMS, and the outputs of MATCHED integrated versions.

S8.4. More Visual Results on Multi-Cue

In this section, we present visual results of MATCHED on Multi-Cue [6] dataset using the two base models: (i) PiDiNet [11] (Figure S12) and (ii) RankED [2] S13).

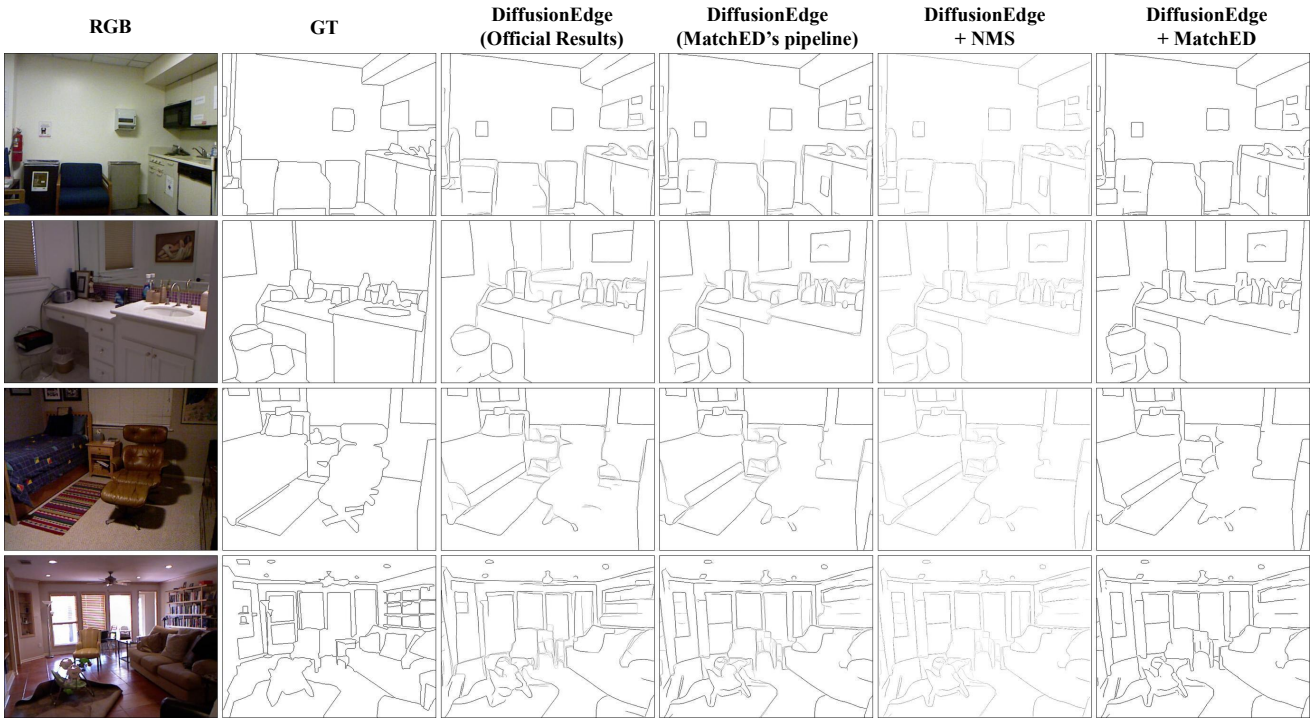


Figure S2. Qualitative comparisons on NYUD dataset using DiffusionEdge [14]. We show results from DiffusionEdge [14] (official checkpoint), raw outputs from our MATCHED pipeline, raw outputs after applying NMS, and their corresponding MATCHED integrated results, respectively. Best viewed zoomed-in.

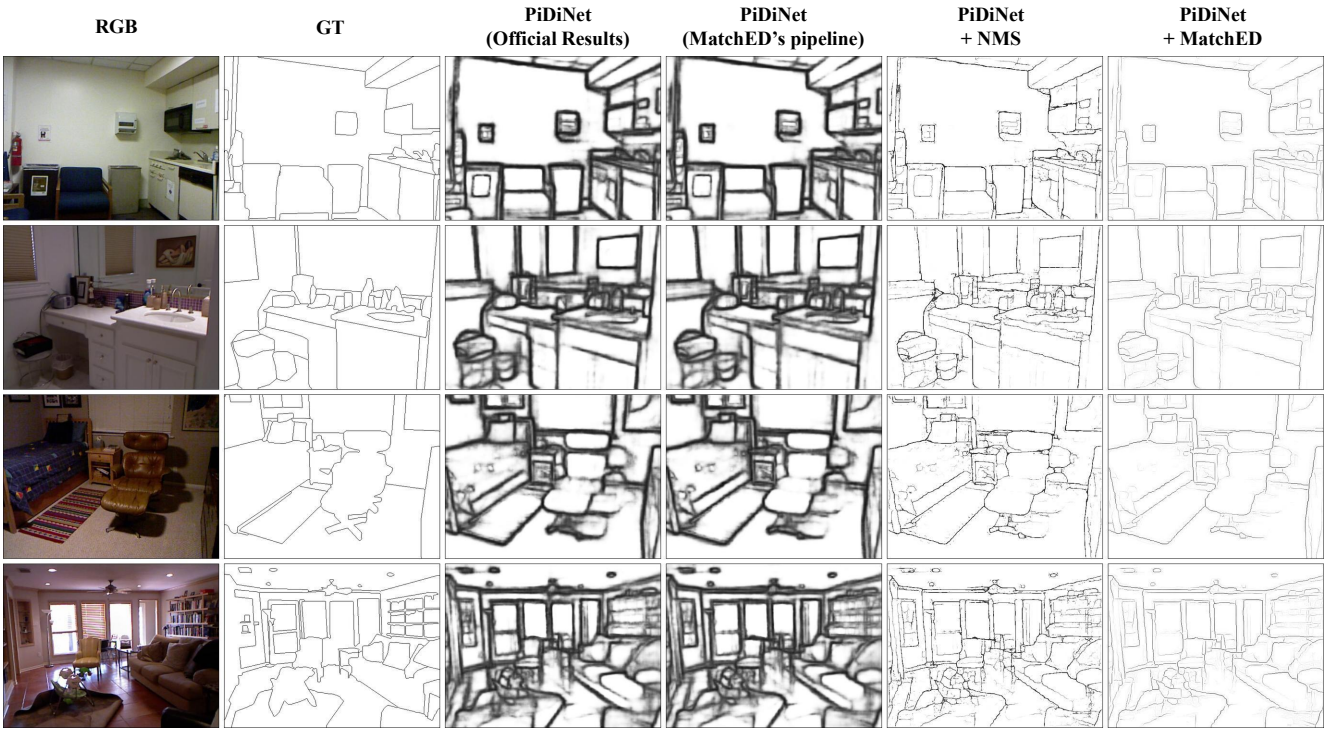


Figure S3. Qualitative comparisons on NYUD dataset using PiDiNet [11]. We show results from PiDiNet [11] (official checkpoint), raw outputs from our MATCHED pipeline, raw outputs after applying NMS, and their corresponding MATCHED integrated results, respectively. Best viewed zoomed-in.

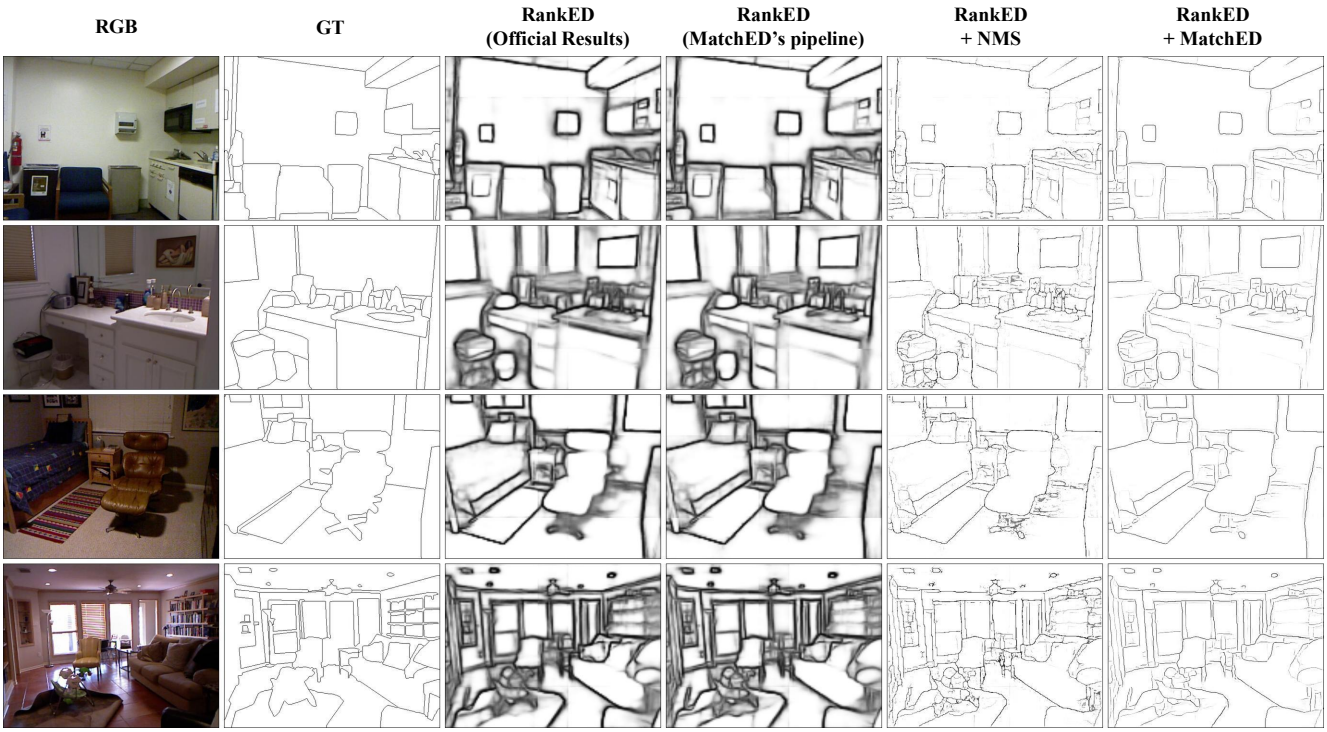


Figure S4. Qualitative comparisons on NYUD dataset using RankED [2]. We show results from RankED [2] (official checkpoint), raw outputs from our MATCHED pipeline, raw outputs after applying NMS, and their corresponding MATCHED integrated results, respectively. Best viewed zoomed-in.

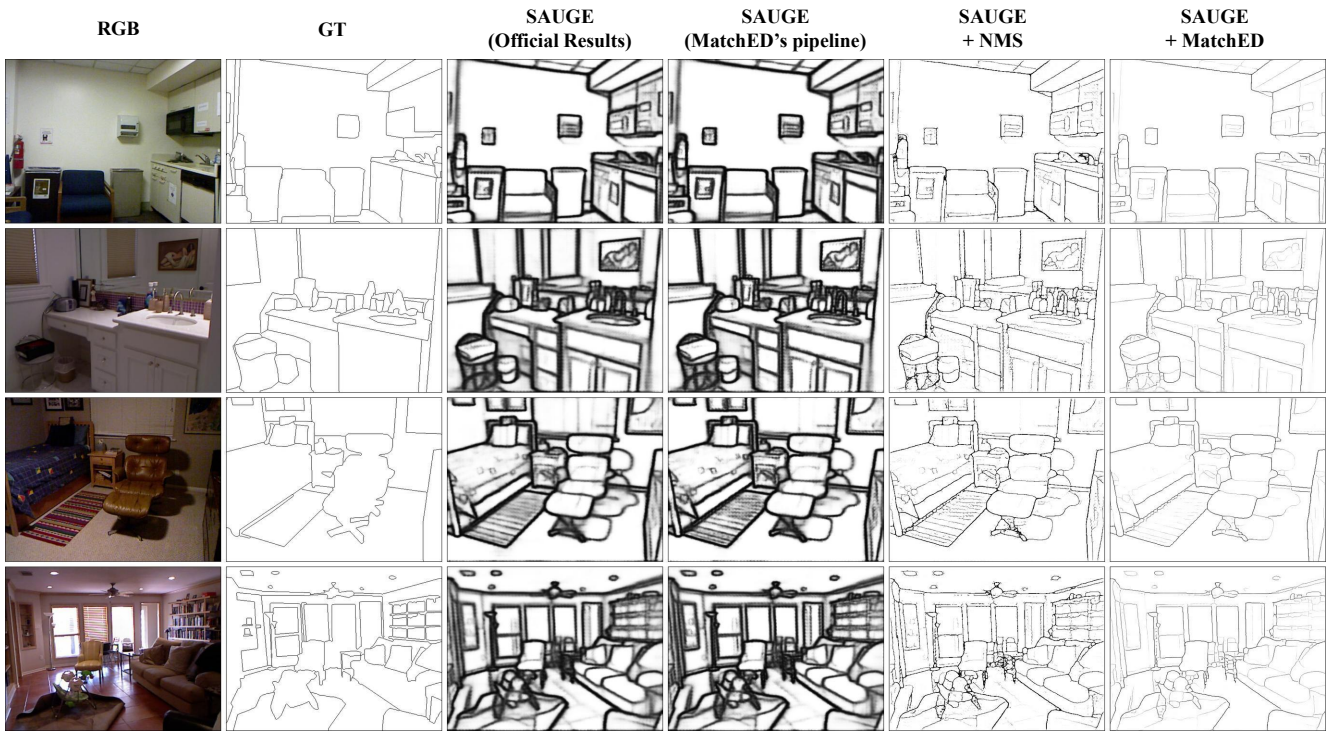


Figure S5. Qualitative comparisons on NYUD dataset using SAUGE [5]. We show results from SAUGE [5] (official checkpoint), raw outputs from our MATCHED pipeline, raw outputs after applying NMS, and their corresponding MATCHED integrated results, respectively. Best viewed zoomed-in.

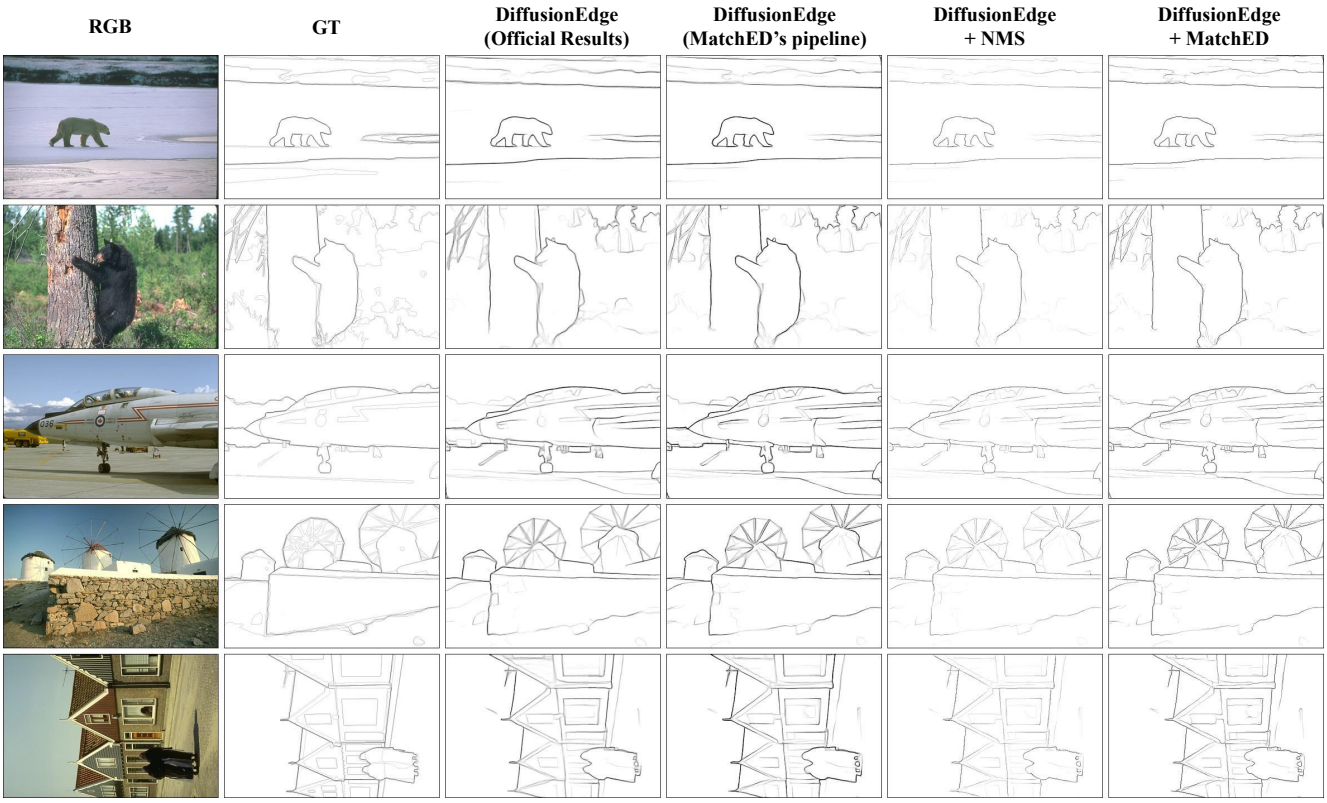


Figure S6. Qualitative comparisons on BSDS dataset using DiffusionEdge [14]. We show results from DiffusionEdge [14] (official checkpoint), raw outputs from our MATCHED pipeline, raw outputs after applying NMS, and their corresponding MATCHED integrated results, respectively. Best viewed zoomed-in.

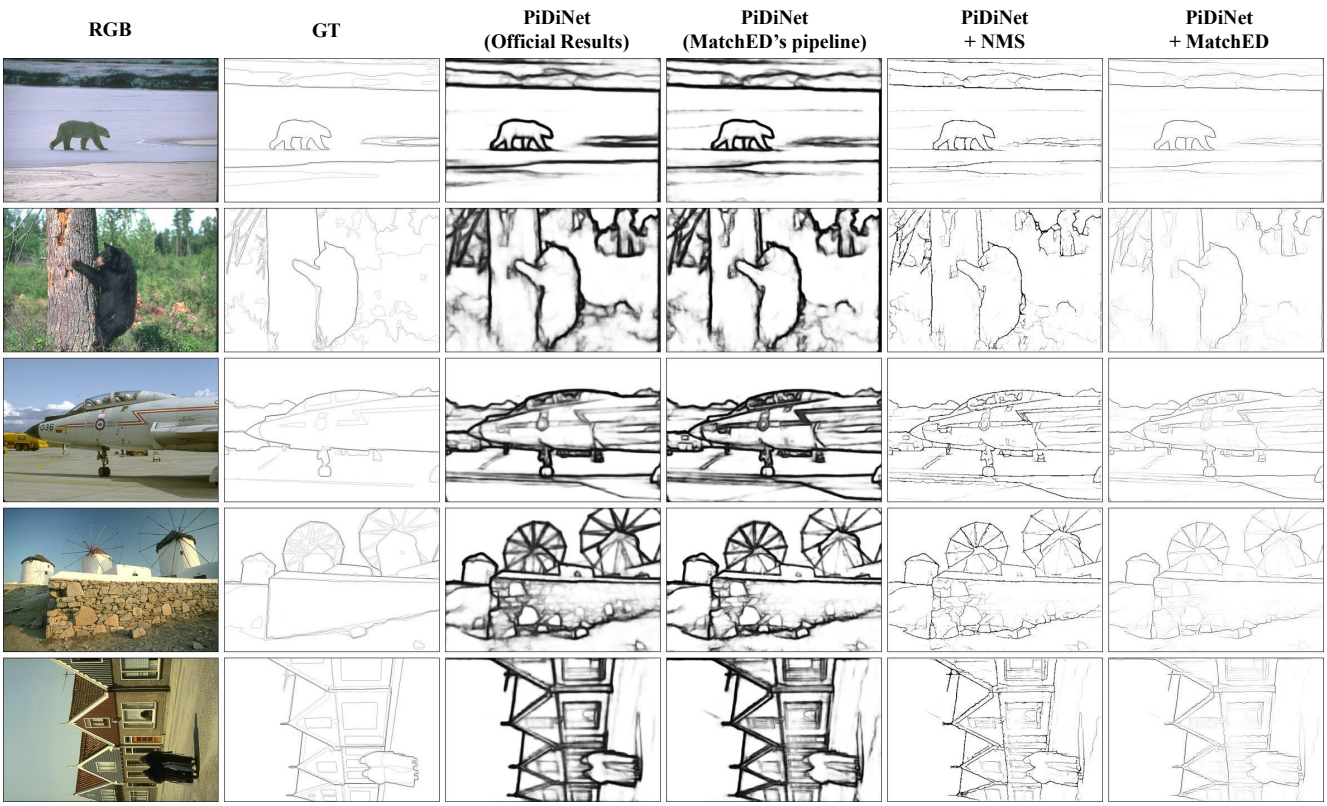


Figure S7. Qualitative comparisons on BSDS dataset using PiDiNet [11]. We show results from PiDiNet [11] (official checkpoint), raw outputs from our MATCHED pipeline, raw outputs after applying NMS, and their corresponding MATCHED integrated results, respectively. Best viewed zoomed-in.

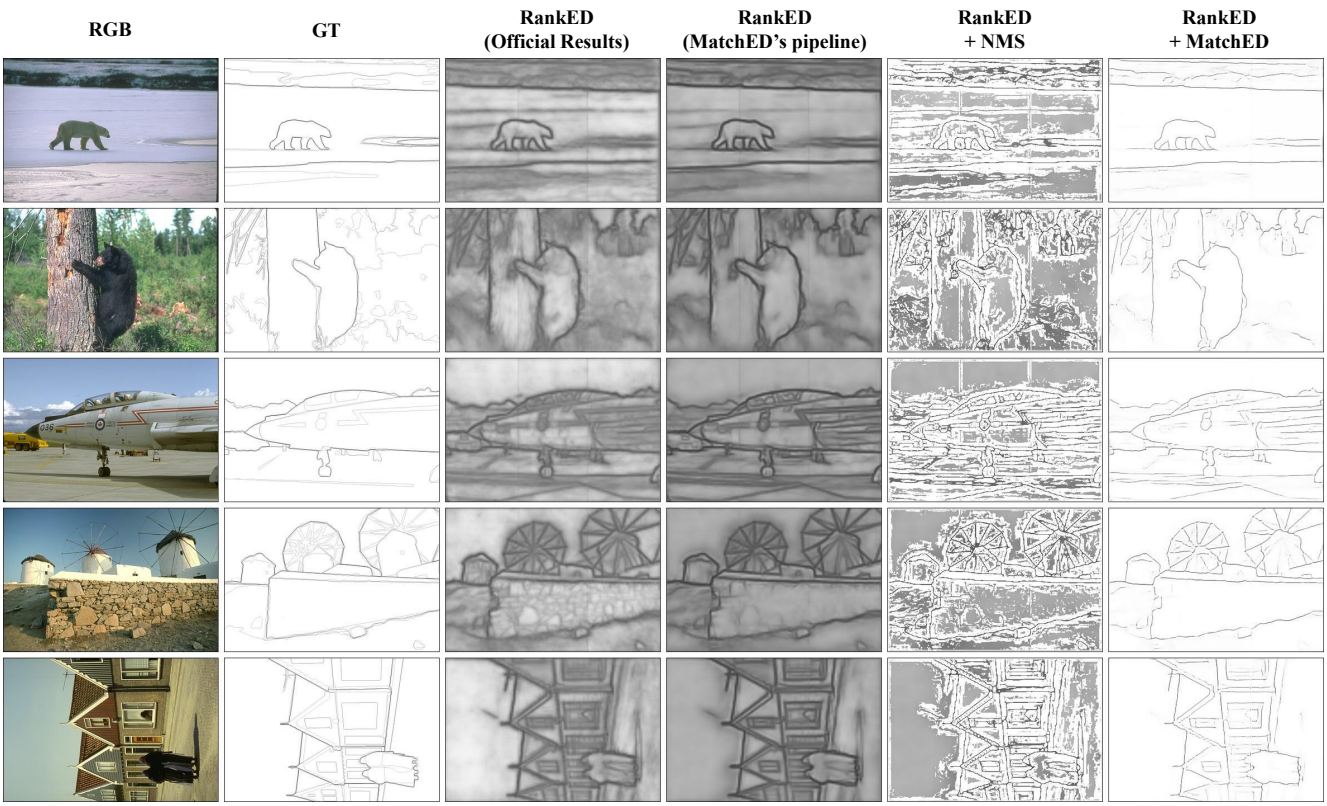


Figure S8. Qualitative comparisons on BSDS dataset using RankED [2]. We show results from RankED [2] (official checkpoint), raw outputs from our MATCHED pipeline, raw outputs after applying NMS, and their corresponding MATCHED integrated results, respectively. Best viewed zoomed-in.

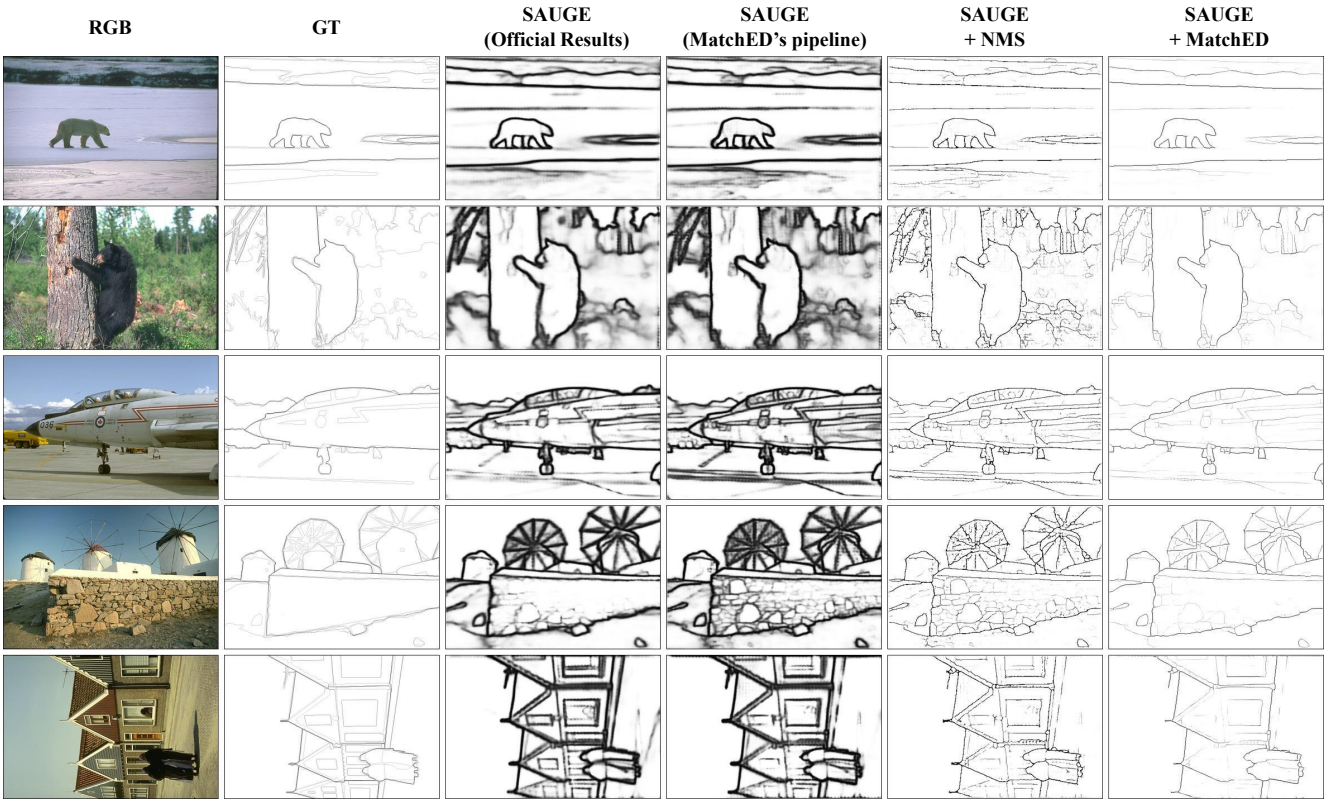


Figure S9. Qualitative comparisons on BSDS dataset using SAUGE [5]. We show results from SAUGE [5] (official checkpoint), raw outputs from our MATCHED pipeline, raw outputs after applying NMS, and their corresponding MATCHED integrated results, respectively. Best viewed zoomed-in.

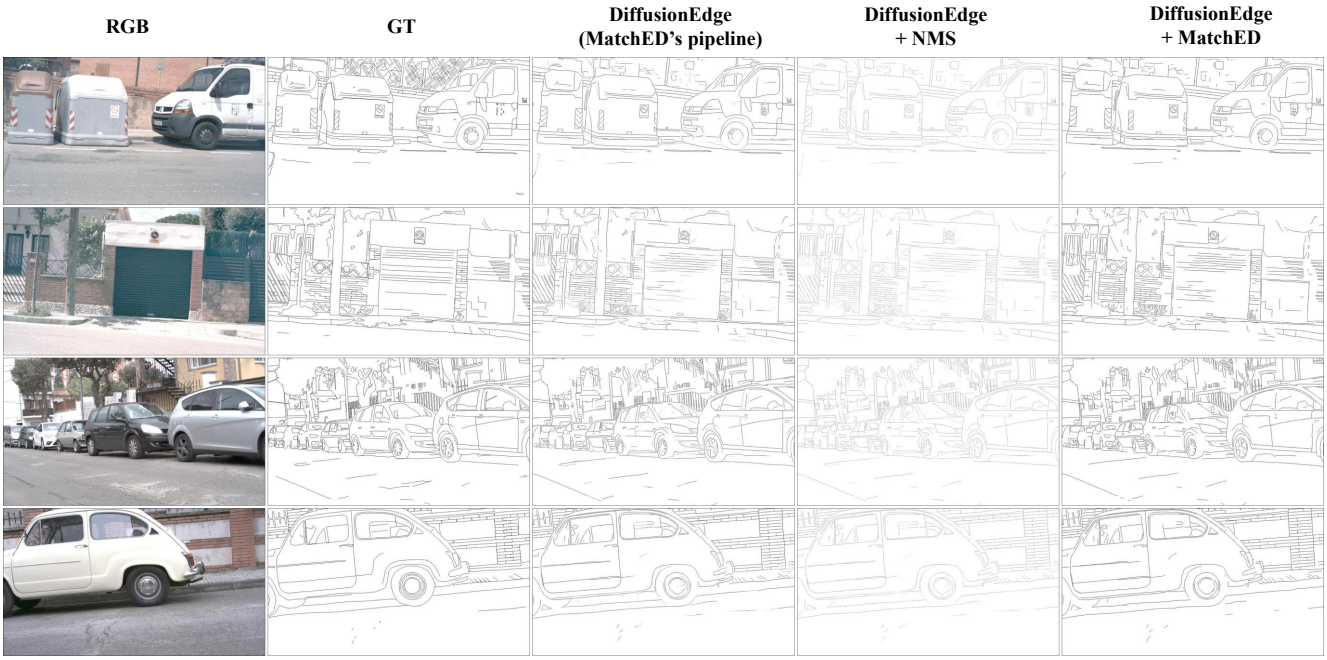


Figure S10. Qualitative comparisons on BIPED dataset using DiffusionEdge [14]. We show results of raw outputs from our MATCHED pipeline, raw outputs after applying NMS, and their corresponding MATCHED integrated version, respectively. Best viewed zoomed-in.

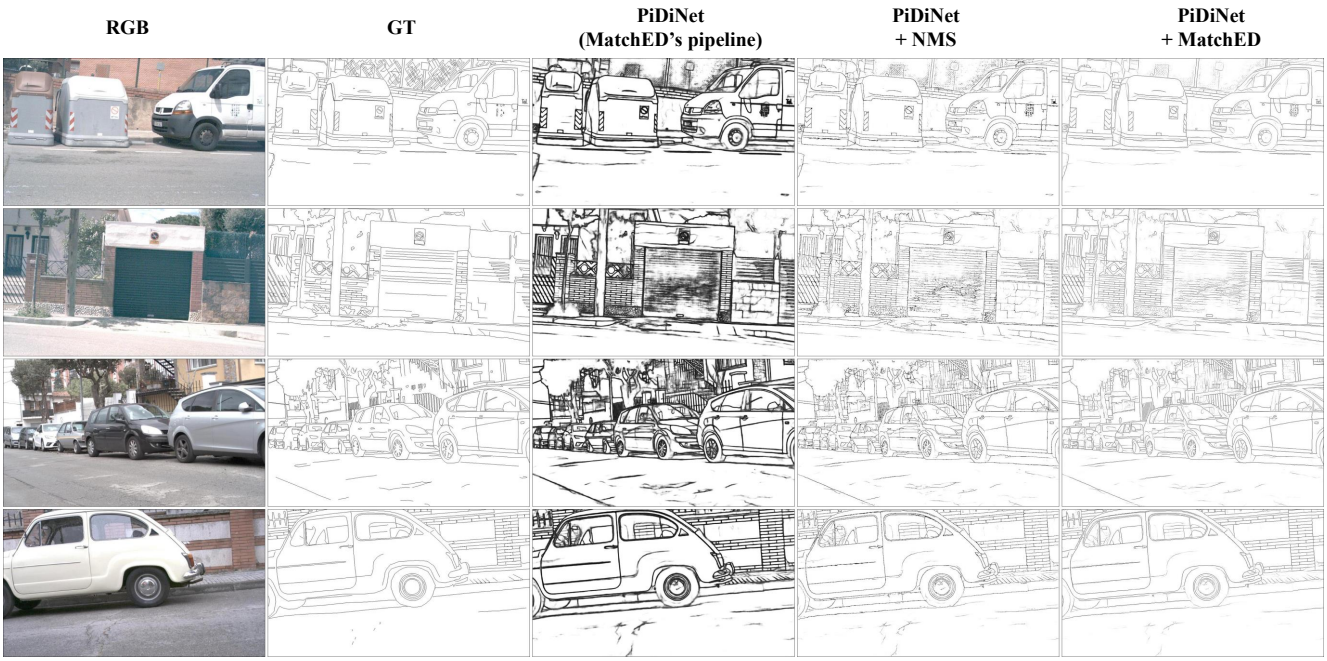


Figure S11. Qualitative comparisons on BIPED dataset using PiDiNet [11]. We show results of raw outputs from our MATCHED pipeline, raw outputs after applying NMS, and their corresponding MATCHED integrated version, respectively. Best viewed zoomed-in.

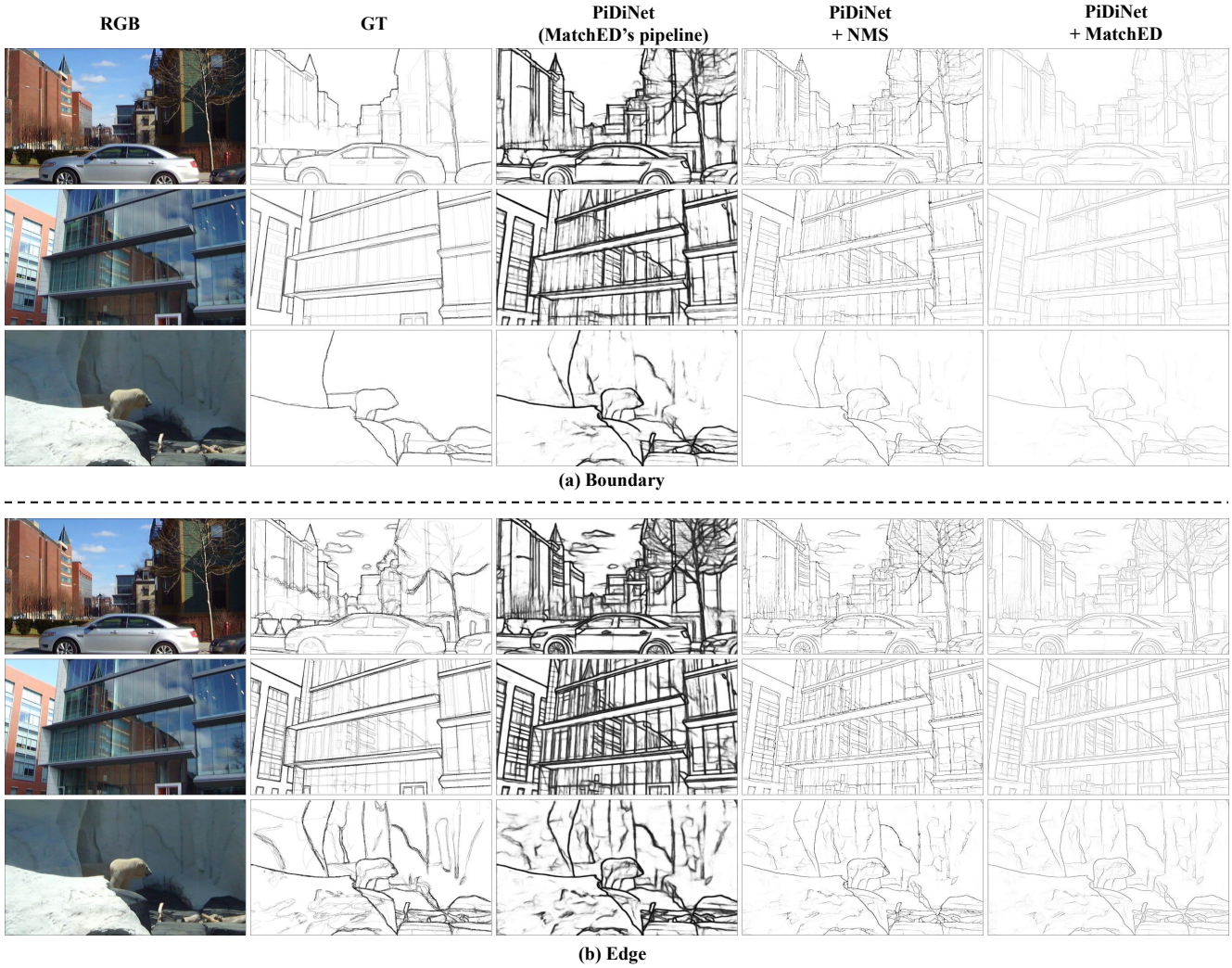


Figure S12. Qualitative comparisons on Multi-Cue dataset using PiDiNet [11]. We show results of raw outputs from our MATCHED pipeline, raw outputs after applying NMS, and their corresponding MATCHED integrated version, respectively. Best viewed zoomed-in. While the upper part shows the boundary detection results, the lower part shows the edge detection results.

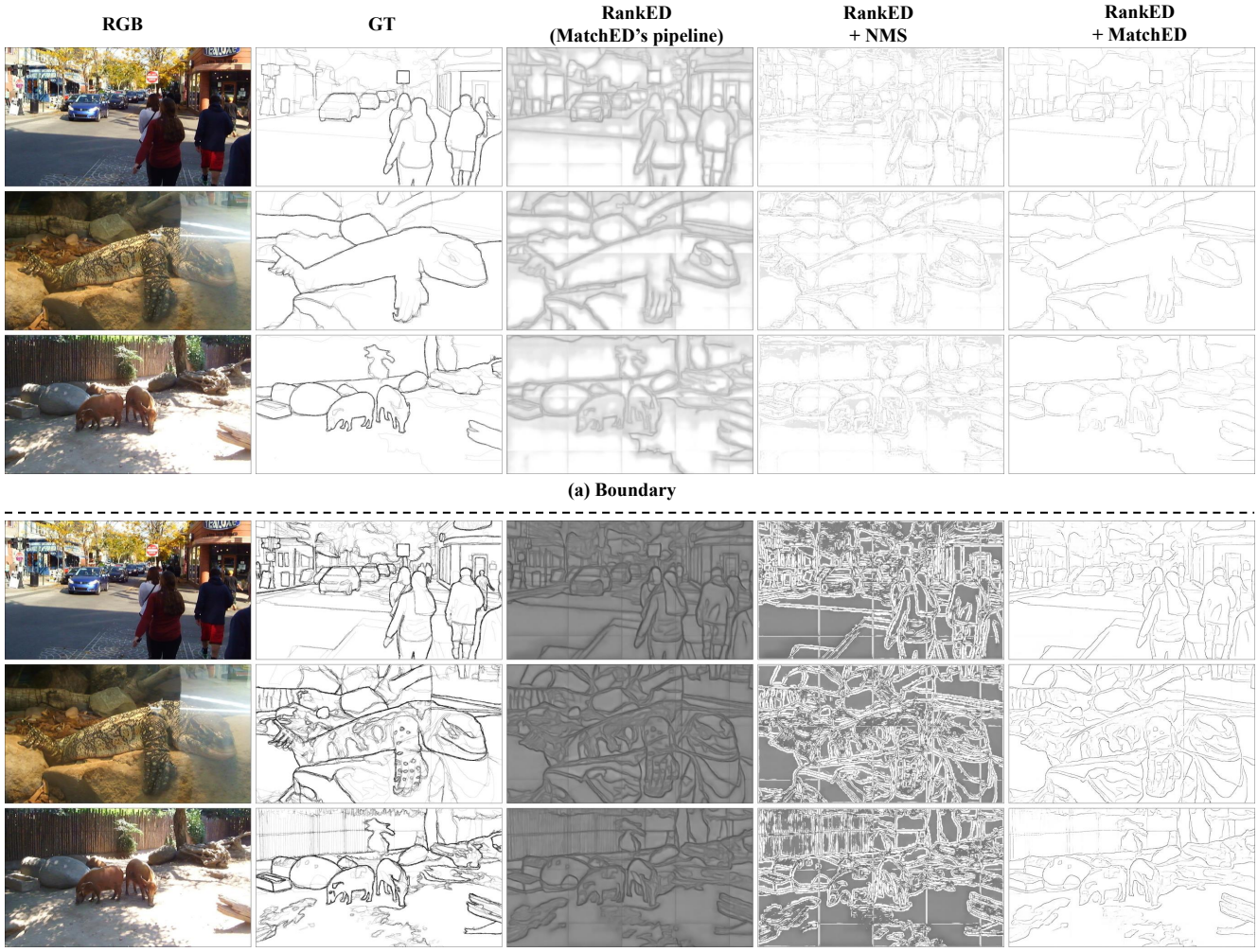


Figure S13. Qualitative comparisons on Multi-Cue dataset using RankED [2]. We show results of raw outputs from our MATCHED pipeline, raw outputs after applying NMS, and their corresponding MATCHED integrated version, respectively. Best viewed zoomed-in. While the upper part shows the boundary detection results, the lower part shows the edge detection results.

References

[1] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2010. 1, 2, 4

[2] Bedrettin Cetinkaya, Sinan Kalkan, and Emre Akbas. Ranked: Addressing imbalance and uncertainty in edge detection using ranking-based losses. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3239–3249, 2024. 2, 3, 4, 7, 11, 15

[3] Jianzhong He, Shiliang Zhang, Ming Yang, Yanhu Shan, and Tiejun Huang. Bi-directional cascade network for perceptual edge detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3828–3837, 2019. 3

[4] Yun Liu, Ming-Ming Cheng, Xiaowei Hu, Kai Wang, and Xiang Bai. Richer convolutional features for edge detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3000–3009, 2017. 3

[5] Xing Liufu, Chaolei Tan, Xiaotong Lin, Yonggang Qi, Jinxuan Li, and Jian-Fang Hu. Sauge: Taming sam for uncertainty-aligned multi-granularity edge detection. *arXiv preprint arXiv:2412.12892*, 2024. 2, 4, 8, 12

[6] David A Mély, Junkyung Kim, Mason McGill, Yuliang Guo, and Thomas Serre. A systematic comparison between visual cues for boundary detection. *Vision Research*, 120:93–107, 2016. 1, 2, 3, 4

[7] Mengyang Pu, Yaping Huang, Yuming Liu, Qingji Guan, and Haibin Ling. Edter: Edge detection with transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1402–1412, 2022. 3

[8] Nathan Silberman and Rob Fergus. Indoor scene segmentation using a structured light sensor. In *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, pages 601–608. IEEE, 2011. 4

[9] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pages 746–760. Springer, 2012. 2

[10] Xavier Soria, Angel Sappa, Patricio Humanante, and Arash Akbarinia. Dense extreme inception network for edge detection. *Pattern Recognition*, 139:109461, 2023. 2, 4

[11] Zhuo Su, Wenzhe Liu, Zitong Yu, Dewen Hu, Qing Liao, Qi Tian, Matti Pietikäinen, and Li Liu. Pixel difference networks for efficient edge detection. In *IEEE/CVF International Conference on Computer Vision*, pages 5117–5127, 2021. 2, 3, 4, 6, 10, 13, 14

[12] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *IEEE/CVF International Conference on Computer Vision*, pages 1395–1403, 2015. 3

[13] Jiawei Yang, Katie Z Luo, Jiefeng Li, Congyue Deng, Leonidas Guibas, Dilip Krishnan, Kilian Q Weinberger, Yonglong Tian, and Yue Wang. Denoising vision transformers. In *European Conference on Computer Vision*, pages 453–469. Springer, 2024. 2

[14] Yunfan Ye, Kai Xu, Yuhang Huang, Renjiao Yi, and Zhiping Cai. Diffusionedge: Diffusion probabilistic model for crisp edge detection. In *AAAI Conference on Artificial Intelligence*, pages 6675–6683, 2024. 2, 3, 4, 5, 9, 12

[15] Caixia Zhou, Yaping Huang, Mengyang Pu, Qingji Guan, Li Huang, and Haibin Ling. The treasure beneath multiple annotations: An uncertainty-aware edge detector. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15507–15517, 2023. 3

[16] Caixia Zhou, Yaping Huang, Mengyang Pu, Qingji Guan, Ruoxi Deng, and Haibin Ling. Muge: Multiple granularity edge detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25952–25962, 2024. 2, 3