Supplementary for Foundation Model Priors Enhance Object Focus in Feature Space for Source-Free Object Detection

Contents

A.1. Algorithm of FALCON-SFOD

Algorithm 1 Training Loop of the proposed method

Require: Teacher h^{te}, student h^{st}; target images $\mathcal{X}_t$
Require: optimizer $\mathrm{Opt}(\cdot)$, hyperparameters
Pre-compute binary foreground masks A_{G}
for each mini-batch $\mathcal{B} \subset \mathcal{X}^t$ **do**
  Augment: obtain weak/strong views $(\tilde{x}_i^t, \bar{x}_i^t)$ for every $x_i^t \in \mathcal{B}$
  Teacher forward: $\tilde{\mathcal{Y}}_i^t \leftarrow h^{\mathrm{te}}(\tilde{x}_i^t)$
  Pseudo-labels: $\hat{\mathcal{Y}}_i^t = \{(\hat{b}_{ij}, \hat{c}_{ij})\} \leftarrow \mathrm{Filter}(\tilde{\mathcal{Y}}_i^t)$
  Student forward: $(\mathbf{p}_i, \mathbf{b}_i) \leftarrow h^{\mathrm{st}}(\bar{x}_i^t)$
  Apply Eq. 4 to $\mathbf{p}_i$ to obtain $\mathbf{p}'_i$
  $w_{\hat{c}} \leftarrow \begin{cases} w_{\mathrm{fg}}, & \hat{c} \text{ is foreground,} \\ w_{\mathrm{bg}}, & \text{otherwise} \end{cases}$
  Compute $\mathcal{L}_{\mathrm{IRPL}}$ via Eq. 5
  Compute student mean maps $A_{\mathrm{S}}(x_i^t) = \text{mean-channel}\big(g^{\mathrm{st}}(x_i^t)\big)$
  Compute $\mathcal{L}_{\mathrm{SPAR}}$ via Eq. 3
  Compute standard detection localization loss $\mathcal{L}_{reg}$
  Aggregate and update: $\theta^{\mathrm{S}} \leftarrow \mathrm{Opt}\big(\theta^{\mathrm{st}}, \nabla_{\theta^{\mathrm{st}}}(\mathcal{L}_{\mathrm{IRPL}} + \mathcal{L}_{\mathrm{SPAR}} + \mathcal{L}_{reg})\big)$
end for
Update teacher: $\theta^{te} \leftarrow \delta\theta^{te} + (1-\delta)\theta^{st}$
return Adapted teacher h^{te}

A.2. Architecture-generality of FALCON-SFOD

To assess the architectural generality of our framework, we integrate FALCON-SFOD with multiple representative detection methods, including both conventional Faster R-CNN-based pipelines [20, 30] and transformer-based detectors such as Deformable DETR [11]. As shown in Table A.1, incorporating our proposed SPAR and IRPL modules consistently improves performance across architectures and domain shift settings. Notably, the observed gains are obtained without modifying the detector structure or adding inference-time cost, underscoring the plug-and-play nature of our design. These results demonstrate that FALCON-SFOD generalizes well beyond a single detector family, enhancing adaptation robustness across both convolutional and transformer architectures.

Table A.1. Performance comparison of our method when integrated into different detection architectures on *Cityscapes*→ *Foggy Cityscapes* (C→F), *Sim10k*→ *Cityscapes* (S→C), and *Kitti*→ *Cityscapes* (K→C). Note that FALCON-SFOD consistently improves perforamance across architectures and domain shifts.

	C→F									S→C	K→C
Method	prsn	rider	car	truck	bus	train	mcycle	bicycle	**mAP**	**AP Car**	**AP Car**
IRG [30] (CVPR'23)	37.4	45.2	51.9	24.4	39.6	25.2	31.5	41.6	37.1	45.2	46.9
+ (Ours)	37	45.9	51.7	30.2	44.7	30	32.9	40.6	**39.0**	**49.1**	**49.8**
PETS [20] (ICCV'23)	42.0	48.7	56.3	19.3	39.3	5.5	34.2	41.6	35.9	57.8	47.0
+ (Ours)	46.2	52.9	63.2	24	49.1	10.4	40.5	48.6	**41.9**	**59.1**	**48.9**
DRU [11] (ECCV'24)	48.3	51.5	62.5	26.2	43.2	34.1	34.2	48.6	43.7	58.7	45.1
+ (Ours)	49.1	52.0	63.7	29.4	45.2	36.7	37.5	49.8	**45.4**	**60.5**	**48.2**

A.3. Reproducibility and Implementation Details

As described in Section A.2, FALCON-SFOD can be seamlessly integrated into a variety of detector architectures, consistently yielding performance gains across settings. For a fair comparison, we preserve all training parameters such as batch size, number of epochs, learning rate, optimization schedule, and data preprocessing pipeline from the corresponding baseline models, applying FALCON-SFOD as an additional adaptation module without altering the underlying training configuration. For the newly introduced hyperparameters, Section A.3.1 demonstrates that a single configuration performs consistently across benchmarks, indicating that FALCON-SFOD exhibits strong robustness and low sensitivity to hyperparameter choices.

A.3.1. Hyperparameter Analysis

Spatial Prior-Aware Regularization. SPAR enforces object-focused feature learning by aligning the student's channel-mean activation map with class-agnostic foreground priors. It achieves this by combining mean ℓ_1 and Dice terms with default weights $\lambda_1 = 1$ and $\lambda_2 = 2$ selected via a coarse sweep (see Table A.2) and kept fixed across all experiments. The ℓ_1 loss enforces pixel-wise agreement, ensuring accurate correspondence in activation magnitudes, while the Dice term complements it by emphasizing overlap and boundary coherence, preventing degenerate solutions where the map matches in average value but misses the overall object shape. Empirically, this combination leads to more stable optimization and consistently higher performance across domain shifts compared to using either term alone. As shown in Table A.2, the mAP varies by less than 1 point across the entire $(\lambda_1, \lambda_2) \in (0, 4)$ range, indicating that SPAR is largely insensitive to its weighting coefficients. We therefore fix a single setting $(\lambda_1{=}1, \lambda_2{=}2)$ for all experiments, confirming that the method does not depend on careful tuning for stable performance.

Table A.2. SPAR λ_1/λ_2 sweep on C→F. Best mAP in **bold**. **Note**: With mAP variation under one point, SPAR demonstrates strong stability with respect to its weighting coefficients.

	λ_2				
λ_1	0	1	2	3	4
0	45	45.6	45.6	45.3	45.5
1	45.4	45.8	**46.1**	45.7	45.6
2	45.3	45.4	45.7	45.6	45.9
3	45.4	45.5	45.4	45.7	45.6
4	45.2	45.3	45.3	45.4	45.6

Imbalance-aware Noise-Robust Pseudo-Labeling. IRPL addresses the dual challenges of label noise and foreground-background imbalance inherent in source-free object detection. Unlike the standard cross-entropy objective, which can be dominated by mislabeled high-confidence predictions, IRPL utilizes a *peak-adjust transform* that moderates the student's logits by adding a large margin m to the most confident class and renormalizing, thereby dampening gradients for easy, clean samples while preserving full corrective signals for uncertain or mislabeled ones. This mechanism provides intrinsic

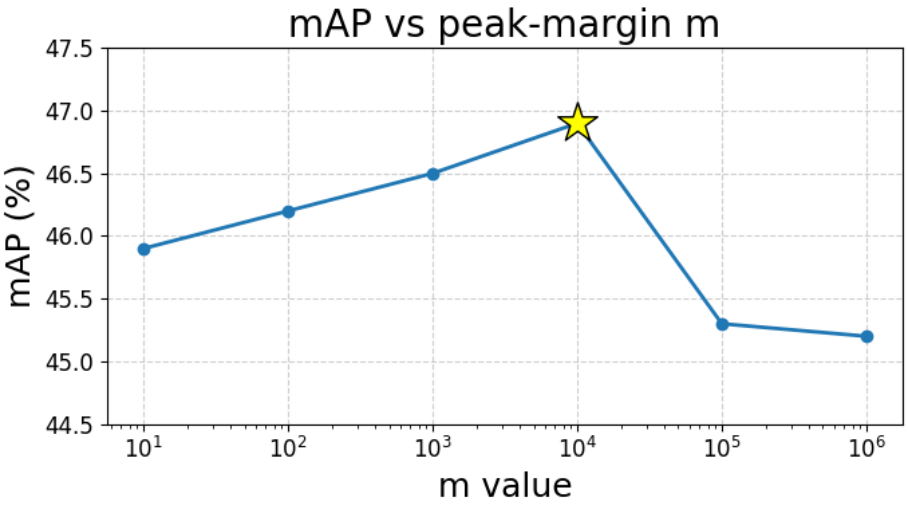

Figure A.1. Performance on C → F with different m values used in the IRPL loss.

robustness to noisy pseudo-labels. To further stabilize learning under imbalance, IRPL combines this with *foreground-background weighting* ($w_{\text{fg}}, w_{\text{bg}}$) and a mild *entropy regularization* term weighted by γ, which discourages over-confident head-class predictions. Following the theoretical insights from [32], we treat the coefficients α and β in

$$L = \alpha L_\epsilon + \beta L_{\text{symmetric}}$$

as scaling factors rather than critical hyperparameters. Lemma 3 in [32] proves that the excess-risk bound of $\alpha L_\epsilon + \beta L_{\text{symmetric}}$ is identical to that of αL_ϵ alone, implying that α and β affect only the gradient magnitude rather than the theoretical robustness. Hence, we fix them following [32] instead of tuning.

- $\alpha = 0.1$**,** $\beta = 2$**.** These values balance the stronger gradients from the Cross-Entropy term and the weaker but noise-robust gradients from the symmetric component, keeping both contributions numerically comparable during training.
- $m = 10^4$**.** We swept $m \in \{10, 10^2, 10^3, 10^4, 10^5, 10^6\}$ (see Figure A.1) and found 10^4 to yield the best trade-off between clean-data fitting and robustness to noisy pseudo-labels, consistent with the analysis that larger m produces more one-hot-like outputs but can lead to underfitting.
- $w_{fg} = 2$, $w_{bg} = 1$. Foreground regions are twice-weighted to emphasize object features over background context during imbalance-aware reweighting. However, we observe that excessively increasing w_{fg} does not further improve performance, as it can overemphasize limited foreground samples and hinder domain alignment with background features.
- $\gamma = 0.01$**.** A mild entropy regularizer; larger values caused over-smoothing, while smaller values provided negligible regularization.

All values were chosen for stability and consistency across benchmarks rather than dataset-specific optimization. This configuration was empirically robust across all experiments without further tuning.

A.3.2. Results across Random Seeds

Statistical Summary. As shown in Table A.3, the mAP values for the three random seeds are 46.9, 46.7, and 47.4. The mean and standard deviation are computed as:

$$\text{Mean} = 47.0, \quad \text{Std} = 0.29.$$

Discussion. While existing SFOD works largely report single-run results, Table A.3 provides performance across three random seeds to evaluate robustness and stability. When integrated into Simple-SFOD [7], our FALCON-SFOD consistently improves performance on the challenging *Cityscapes→Foggy Cityscapes* domain shift. Across seeds 42, 123, and 9999, our approach achieves an average mAP of **47.0** $\pm$ **0.29**, demonstrating both a clear improvement over the Simple-SFOD baseline (45.0 mAP) and strong reproducibility. The small variance across runs indicates that FALCON-SFOD's gains are stable and not sensitive to random initialization.

Table A.3. Performance comparison of our method (three random seeds) when integrated into Simple-SFOD [7] on *Cityscapes*→ *Foggy Cityscapes* (C→F). FALCON-SFOD shows consistent improvements and robustness across different runs.

	C→F								
Method	prsn	rider	car	truck	bus	train	mcycle	bicycle	**mAP**
Simple-SFOD [7] (ECCV'24)	40.9	48.0	58.9	29.6	51.9	50.2	36.2	44.1	45.0
+ (Ours, Seed 42)	41.0	48.3	58.7	33.6	54.8	54.3	38.6	46.2	46.9
+ (Ours, Seed 123)	41.3	48.1	59.2	32.7	54.2	53.5	38.4	46.3	46.7
+ (Ours, Seed 9999)	41.6	48.6	59.5	34.1	55.1	54.9	39.1	46.8	47.4

A.4. Proofs for the Lemmas and Theorems

Proof for Lemma 1.

Proof. For any x and any true class $j \in \{0, \dots, K\}$ define the non-negative loss vector $\boldsymbol{\ell}(x) = \big(\ell_0(x), \dots, \ell_K(x)\big)^\top$ with $\ell_i(x) = -\log p_i(x) \geq 0$. Since $T_{jj} \geq \lambda$ we have

$$\begin{aligned} \ell_j(x) &= \frac{T_{jj}}{T_{jj}}\,\ell_j(x) \leq \frac{1}{\lambda}\,T_{jj}\,\ell_j(x) \\ &\leq \frac{1}{\lambda}\sum_{i=0}^{K} T_{ji}\,\ell_i(x). \end{aligned} \tag{A.1}$$

Taking expectation under the joint (x, c) and using $\Pr[\hat{c} = i \mid c = j, x] = T_{ji}$ yields

$$\begin{aligned} R^{cls}_{\mathcal{D}^T,\,clean}(f^{st}_c) &= \mathbb{E}_{(x,c)}\big[\ell_c(x)\big] \\ &\leq \frac{1}{\lambda}\,\mathbb{E}_{(x,c)}\Big[\sum_i T_{ci}\,\ell_i(x)\Big] \\ &= \frac{1}{\lambda}\,R^{cls}_{\mathcal{D}^T,\,noisy}(f^{st}_c). \end{aligned} \tag{A.2}$$

This completes the proof of Lemma 1. □

Proof for Lemma 2.

Proof. We write the clean risk as the expectation over the two disjoint events $M = 1$ and $M = 0$:

$$\begin{aligned} \|f^{st}_r(x) - b\|_1 &= M\,\|f^{st}_r(x) - b\|_1 \\ &\quad +\ (1 - M)\,\|f^{st}_r(x) - b\|_1. \end{aligned} \tag{A.3}$$

Case $M = 1$ (teacher matched the box): When $M = 1$ there exists the pseudo-box $\hat{b}$ and by the triangle inequality

$$\|f^{st}_r(x) - b\|_1 \leq \|f^{st}_r(x) - \hat{b}\|_1 + \|\hat{b} - b\|_1. \tag{A.4}$$

Case $M = 0$ (teacher missed the box): With normalised coordinates $\|f^{st}_r(x) - b\|_1 \leq 2$ for all x, b, hence

$$(1 - M)\,\|f^{st}_r(x) - b\|_1 \leq 2\,(1 - M). \tag{A.5}$$

Taking expectations over $\mathcal{D}^T$ and summing the two cases gives

$$\begin{aligned} R^{reg}_{\mathcal{D}^T,\,clean}(f^{st}_r) &\leq \mathbb{E}\big[M\,\|f^{st}_r(x) - \hat{b}\|_1\big] \\ &\quad + \mathbb{E}\big[M\,\|\hat{b} - b\|_1\big] + 2\,\mathbb{E}[1 - M]. \end{aligned} \tag{A.6}$$

Using the definitions $R^{reg}_{\mathcal{D}^T,\,noisy} = \mathbb{E}[M\,\|f^{st}_r(x) - \hat{b}\|_1]$, $\eta_{reg} = \mathbb{E}[M\,\|\hat{b} - b\|_1]$, and $\zeta = \mathbb{E}[1 - M]$, we recover (9). □

Proof for Theorem 1.

Proof. Starting from the decomposition

$$R^{det}_{\mathcal{D}^T} = R^{cls}_{\mathcal{D}^T,clean} + R^{reg}_{\mathcal{D}^T,clean}, \tag{A.7}$$

(cf. Eq. 6), we apply Lemma 1 to the classification term and Lemma 2 to the regression term:

$$R^{cls}_{\mathcal{D}^T,clean}(f^{st}_c) \leq \tfrac{1}{\lambda}\, R^{cls}_{\mathcal{D}^T,noisy}(f^{st}_c), \tag{A.8}$$

$$R^{reg}_{\mathcal{D}^T,clean}(f^{st}_r) \leq R^{reg}_{\mathcal{D}^T,noisy}(f^{st}_r) + \eta_{reg} + 2\,\zeta. \tag{A.9}$$

Adding the two inequalities gives (10). □

Proof for Theorem 2.

Proof. Inspired from [32], we can write

$$R^{\text{cls}}_{\mathcal{D}^T,\text{clean}}\big(f^*_\eta\big) \leq 2\delta + \frac{2w\,\delta}{a}. \tag{A.10}$$

Combining this with the regression bound that accommodates missed boxes (Lemma 2),

$$R^{\text{reg}}_{\mathcal{D}^T,\text{clean}}(f^{st}_r) \leq R^{\text{reg}}_{\mathcal{D}^T,\text{noisy}}(f^{st}_r) + \eta_{\text{reg}} + 2\zeta, \tag{A.11}$$

and using the decomposition $R^{\text{det}} = R^{\text{cls}}_{\text{clean}} + R^{\text{reg}}_{\text{clean}}$ yields inequality (11). □

A.5. Run time and Memory Analysis

Tables A.4 and A.5 report the cost of integrating G-SAM [24] into our pipeline. The time overhead is marginal: the offline G-SAM pass completes in 1 050 s (∼17 min), which is only 3.8% of the 28 000 s required for our model training (and only 5.2% relative to the baseline). When added to the full pipeline, the end-to-end wall-clock increases only slightly (28 084 s → 29 134 s), well within the typical run-to-run variability of large-scale training. The memory overhead is also short-lived: G-SAM peaks at 18 GB only during its 17-minute preprocessing, while training itself never exceeds 9.6 GB. Since these stages do not overlap, the entire procedure fits comfortably on a single 24-48 GB GPU without any modification to training. Moreover, the amortization cost is small: G-SAM masks are generated once per target split and can be cached for reuse in all subsequent experiments. The extraction step is fully parallel across images, so on a multi-GPU node the elapsed time approaches standard data-loading latency. *For completeness, we note that ESC-Net [14] and OV-SAM [36] are both lighter than Grounded-SAM in parameter size, and therefore require less compute and memory*; we report G-SAM values here since it represents the most demanding case among the three. Overall, even with G-SAM enabled, the complete adaptation run finishes in under 8 hours and < 18 GB peak memory on a single RTX A6000, confirming that the footprint remains modest.

Table A.4. Run time comparison between baseline and our method on C → F domain shift. Note that the time taken for offline generation of foreground-masks is negligible compared to the training time. Also, the inference time remains the same as the baseline.

Setting	Offline G-SAM time (1000s)	Training time (1000s)	Test time (s)	End-to-end time (1000s)
IRG [30]	–	20	84	20.08
IRG + Ours	–	28	84	28.08
IRG + Ours + G-SAM pre-processing cost	1.050	28	84	29.13

A.6. Additional Qualitative Results

Figure A.2 presents additional qualitative examples comparing the baseline method [7] and our proposed method on the Foggy Cityscapes dataset. Each row corresponds to one scene, with the first two columns illustrating the baseline's thresholded mean-channel feature maps and predicted detections, and the last two columns showing the same representations from our method. The mean-channel map is obtained by taking the mean along the channel dimension of the last layer of the backbone

Table A.5. Memory usage comparison between baseline and our method on C → F domain shift. Note that the offline foreground-mask generation is short-lived and both the train-time and test-time memory remains same as the baseline.

Setting	Offline G-SAM peak mem (GB)	Training peak mem (GB)	Stage-wise peak mem (GB)
IRG [30]	–	6.9	6.9
IRG + Ours	–	9.6	9.6
IRG + Ours + G-SAM pre-processing cost	18.4	9.6	18.4

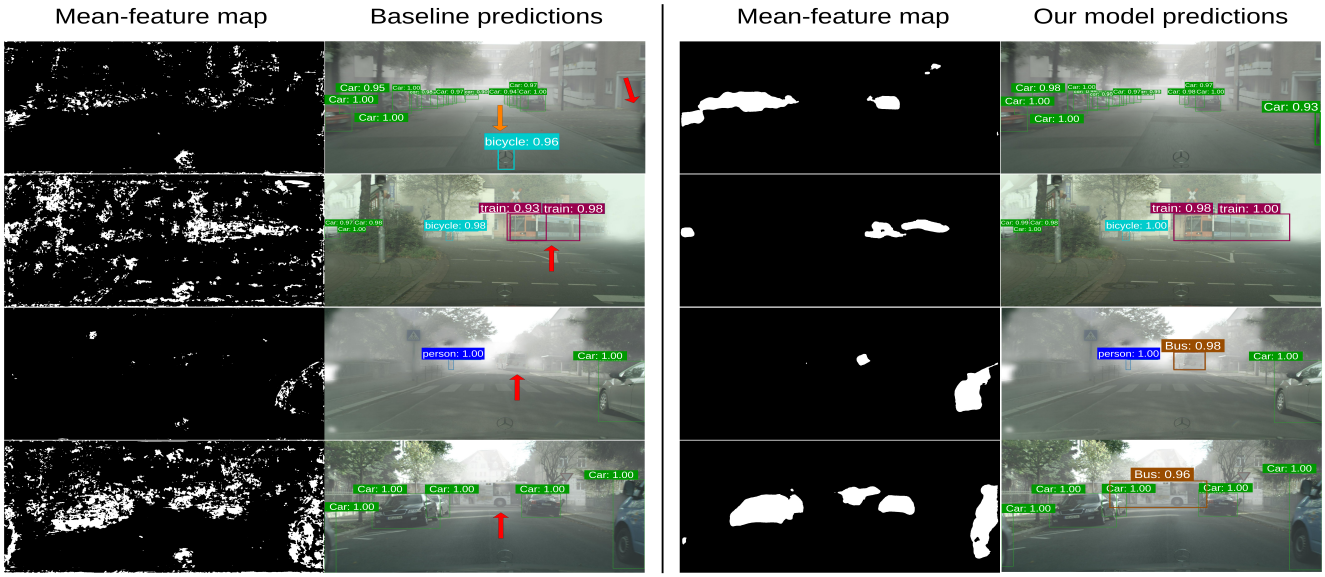

Figure A.2. **Additional qualitative results.** Four examples from the Foggy Cityscapes [26] target set. Mean-feature map is obtained from taking the channel-mean from the last layer of the student's backbone and thresholding at 0.6.**Left**: Baseline model [7] produces spurious background activations, leading to missed detections or localization errors (red arrows) and false positives (orange arrows). **Right**: Our method effectively suppresses both feature-space confusion and class-label noise, resulting in clear activations and more accurate classification and object localization. *Zoom in for best view*.

and thresholding at 0.6, which is then upsampled to the image dimension for visualization. The baseline consistently exhibits dispersed activations, causing inaccuracies such as false positives (bicycle in the first image) and missed detections (car in the first image, incomplete train localization in the second, and missed buses in the third and fourth images). In contrast, our method significantly reduces the irrelevant background activations, producing cleaner, focused activations that accurately highlight relevant objects and recover detections missed by the baseline. Overall, these qualitative results illustrate that our method addresses both the pseudo-label noise and feature-space confusion that occurs due to the domain shift, resulting in improved detection accuracy and reliability.

A.7. More details on Extreme shifts

Table A.6. SFOD baselines on extreme shifts (mAP).

Method	V→CL	FI→FV	FV→CO
IRG (CVPR'23)	31.5	56.5	18.6
PETS (ICCV'23)	32.0	56.9	18.8
Simple-SFOD (ECCV'24)	33.6	56.7	19.4
DRU (ECCV'24)	33.9	57.4	19.3
FALCON-SFOD	**35.5**	**58.5**	**20.9**

Tab. A.6 compares the performance of FALCON-SFOD with prior SFOD methods on the extreme shift scenarios. Existing works largely do not report stress-test transfers; we ran IRG/PETS/Simple-SFOD/DRU on these domain shifts for comparison. FALCON-SFOD consistently achieves the best performance across the shifts.

A.8. Progressive pseudo-label quality.

Fig. A.3 tracks pseudo-label precision and recall over training epochs on Foggy Cityscapes, showing progressive improvement during adaptation.

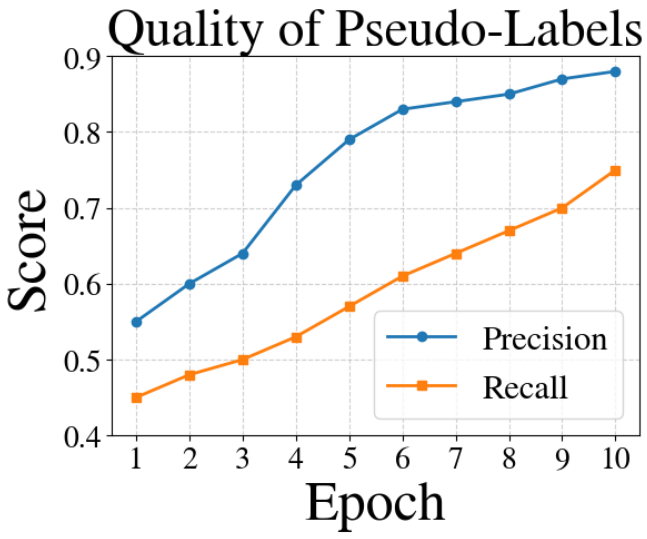

Figure A.3. Pseudo-label metrics improve over epochs.

A.9. Quantitative Feature Focus Diagnostics and Prior Failures

On FoggyCityscapes, scoring pixels by student channel-mean activation $A(p)$ and comparing to ground truth box masks, we report pixel ROC-AUC and $\text{FAM} = \frac{\sum_{p\in\text{FG}} A(p)}{\sum_p A(p)}$: Baseline (AUC/FAM)$= 0.61/23.7\%$; **+SPAR**$= 0.84/76.3\%$; **Full**$=$ $0.87/80.6\%$ ($> 3\times$ FAM). mAP improves $45.0 \rightarrow 46.1$ with +SPAR; +IRPL alone barely changes focus (FAM $23.7 \rightarrow 24.4\%$) with mAP $45.0 \rightarrow 45.8$; *Full method* is best (mAP 46.9, FAM 80.6%), indicating SPAR drives focus while IRPL complements via residual noise/imbalance handling.

We simulate segmentation mask-failure by (i) missing-object noise: dropping OV-SAM FG pixels (20/40/60%) gives mAP $46.8/46.4/45.9$, and (ii) over-segmentation noise: dilating FG into background pixels (20/40/60%) gives mAP $46.7/46.5/45.8$ (vs. 46.9 with full masks; baseline 45.0), showing SPAR is not brittle to prior failures.