Semantically Aware Optimal Transport for Dense Label Transfer

Preeti and Kiran Ravish and Ankita Kushwaha and Pawan Kumar
IIIT Hyderabad, India
{preeti.preeti, kiran.ravish, ankita.kushwaha}@research.iiit.ac.in
pawan.kumar@iiit.ac.in

Abstract

Vision foundation models produce features that generalize across visual domains without fine-tuning, yet naively transferring labels through these feature spaces fails under large distribution shifts. We propose SAOT (**S**emantically **A**ware **O**ptimal **T**ransport), which learns a transport cost within a fused unbalanced optimal transport formulation for dense label transfer from frozen vision transformer features to new domains. SAOT combines a learnable appearance metric with semantic class-prototype priors, unbalanced transport for partial matching under distribution shift, and a block-sparse solver for tractable inference. We pair this with a two-stage decoder: an MLP trained on SAOT pseudo-labels, then refined via EMA-teacher self-training with class-balanced sampling. On GTA5→Cityscapes with frozen DINOv2 ViT-L/14 features, SAOT+Decoder reaches 25.7% mIoU, a **3.8**× improvement over nearest-neighbor transfer (6.7%), without any backbone adaptation. Per-class results show large gains on spatially coherent classes (road 90.3%, car 76.2%, building 71.5%), demonstrating that learned semantic transport costs capture domain-invariant structure even under severe synthetic-to-real shifts. On VOC train→val with frozen ViT-B/16 features, the full pipeline reaches 47.5% mIoU, indicating that the approach extends beyond synthetic-to-real adaptation.

1 Introduction

Optimal transport (OT) aligns distributions across domains (Villani, 2009; Peyré and Cuturi, 2019). Entropic regularization makes OT solvable via Sinkhorn scaling (Sinkhorn and Knopp, 1967; Cuturi, 2013; Altschuler et al., 2017), and the resulting solvers have been used for dense semantic matching (Liu et al., 2020) and differentiable assignment (Sarlin et al., 2020).

Dense label transfer, propagating a semantic label field from a labeled source domain to an unlabeled target, is a core task in unsupervised domain adaptation (UDA) for segmentation. In the synthetic-to-real setting, a model trained on GTA5 (Richter et al., 2016) must segment real Cityscapes (Cordts et al., 2016) images. Most OT-based methods use *fixed* ground costs (e.g., cosine distance in frozen features), which misalign semantics when appearance varies across domains.

Dense label transfer across domains faces three problems: (1) **Semantic misalignment**: fixed feature distances match visually similar but semantically different regions (synthetic road textures mapped to real sidewalks); (2) **Mass imbalance**: class frequencies differ between source and target, and not all source mass has a valid target correspondence; (3) **Scalability**: vision transformers produce thousands of patch tokens per image, so the solver must be efficient.

To address these issues, we propose SAOT, which *learns a semantically aware transport cost directly in feature space*. SAOT *learns* a semantically aware transport cost while keeping the inner OT problem well-posed via entropic regularization and unbalanced constraints. The cost combines a learnable appearance metric with semantic priors derived from class prototypes. A block-sparse solver restricts transport to k-nearest candidate neighborhoods for tractable inference. We then train a residual MLP decoder on SAOT pseudo-labels and refine it with EMA-teacher self-training (Tarvainen and Valpola, 2017).

Contributions.

- A learnable ground cost c_θ that blends appearance distance with class-prototype semantic priors for matching under domain shift (Section 4.1).
- A fused unbalanced OT objective with entropic regularization, KL-relaxed marginals, and entropy/diversity auxiliary losses (Section 4.2).

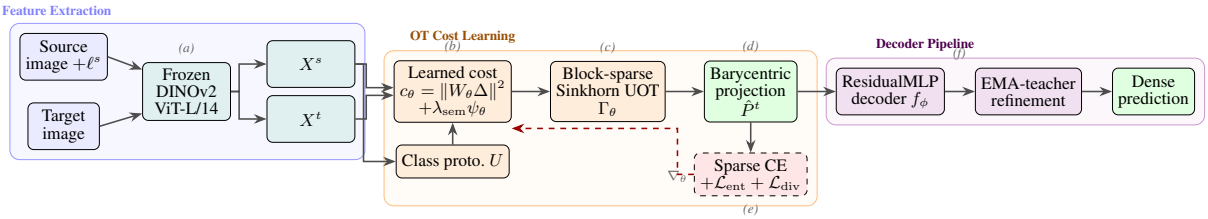

Figure 1: Overview of the SAOT framework. **(a)** Frozen DINOv2 ViT-L/14 extracts source/target patch tokens. **(b)** Learnable cost c_θ blends appearance distance with semantic class-prototype priors. **(c)** Block-sparse unbalanced Sinkhorn solver computes coupling Γ_θ. **(d)** Barycentric projection transfers source labels to target. **(e)** Sparse supervision + auxiliary losses drive bilevel cost learning (dashed red = gradient). **(f)** Two-stage decoder distills OT pseudo-labels, refined via EMA self-training.

- A k-NN block-sparse solver that reduces the coupling from $O(nm)$ to $O(nk)$ for high-resolution inference (Section 4.3).
- A two-stage decoder trained on SAOT pseudo-labels and refined with EMA-teacher self-training (Section 4.5).
- Within the class of source-free dense label-transfer methods using frozen foundation-model features, evaluation on GTA5→Cityscapes at 504×1008 resolution with frozen DINOv2 ViT-L/14 features yields 25.7% mIoU, a 3.8× gain over nearest-neighbor baselines, together with ablations and per-class analysis (Section 6).
- We validate the method on a second benchmark, VOC train→val with frozen ViT-B/16 features, where the full pipeline reaches 47.5% mIoU, showing that the approach extends beyond synthetic-to-real transfer.

2 Related Work

Entropic OT and differentiable transport. Entropic regularization makes OT strictly convex and solvable via Sinkhorn scaling (Cuturi, 2013), yielding GPU-friendly solvers and differentiable assignment layers (Sarlin et al., 2020; Mena et al., 2018). We build on these properties for dense label transfer with learned costs.

Unbalanced and robust OT. Classical OT assumes mass conservation, but occlusions and label shift violate this in dense transfer. Unbalanced OT relaxes marginals via KL divergences (Chizat et al., 2018a,b; Liero et al., 2018), and RWOT adds reliability weighting (Xu et al., 2020). We use the unbalanced formulation as an outlier filter to avoid forced mismatches.

Structure-aware OT: GW, FGW, and fused variants. GW distances align relational structures (Mémoli, 2011), while FGW combines feature and structure terms (Vayer et al., 2019, 2020). FUGW adds unbalanced relaxations (Thual et al., 2022), and ASOT applies this to temporal consistency (Xu and Gould, 2024). We fuse semantic costs with spatial structure on dense vision grids.

OT and graph-based methods for domain adaptation. OT is widely used in DA (Courty et al., 2017b,a; Damodaran et al., 2018; Seguy et al., 2018). These methods learn feature extractors; we instead learn the *ground cost* while keeping foundation model features frozen. Label propagation (Zhu et al., 2003) transfers labels via diffusion on a k-NN affinity graph; it is a natural transductive baseline but does not outperform simple k-NN when features are frozen (Table 1).

UDA for semantic segmentation. Modern UDA methods pair self-training with strong architectures: DAFormer (Hoyer et al., 2022a) (68.3% mIoU on GTA5→Cityscapes) uses transformer decoders, HRDA (Hoyer et al., 2022b) (73.8%) adds multi-resolution context, and MIC (Hoyer et al., 2023) (75.9%) uses masked consistency. All fine-tune the full network end-to-end on source *and* target data. Our approach works on frozen features and requires no backbone training.

Vision foundation models for segmentation. Self-supervised ViTs (Caron et al., 2021; Oquab et al., 2024; Dosovitskiy et al., 2021) produce semantically rich patch tokens. DINOv2 (Oquab et al., 2024) features transfer across domains without fine-tuning. Rein (Wei et al., 2024) adapts DINOv2 via LoRA for domain generalization, reaching 78.4% mIoU when trained on synthetic data, but this requires backbone adap-

tation. STEGO (Hamilton et al., 2022) distills DINO feature correspondences into a segmentation head without any labels, reaching ∼21% mIoU on Cityscapes. We use frozen DINOv2 features as input to our transport framework; the method is complementary to adapter-based approaches like Rein, which could replace the frozen backbone.

3 Notation and Preliminaries

Setup. Let a source image be represented by n patch tokens $X^s = \{x_i^s\}_{i=1}^n \in \mathbb{R}^{n\times d}$ with labels $\ell_i^s \in \{1,\dots,K\}$, and a target image by m tokens $X^t = \{x_j^t\}_{j=1}^m$. We seek a coupling $\Gamma \in \mathbb{R}_+^{n\times m}$ transporting label mass from source to target such that correspondences are semantically meaningful.

Entropic unbalanced OT. Given nonneg. weights $\boldsymbol{a} \in \mathbb{R}_+^n$, $\boldsymbol{b} \in \mathbb{R}_+^m$ and cost $C \in \mathbb{R}^{n\times m}$, the entropic UOT objective (Chizat et al., 2018a) is:

$$\Gamma^\star \in \arg\min_{\Gamma\geq 0} \ \langle \Gamma, C\rangle + \varepsilon\,\mathrm{KL}(\Gamma\|\boldsymbol{a}\boldsymbol{b}^\top) + \tau\big(\mathrm{KL}(\Gamma\mathbb{1}\|\boldsymbol{a}) + \mathrm{KL}(\Gamma^\top\mathbb{1}\|\boldsymbol{b})\big). \tag{1}$$

The entropic term makes the objective strictly convex in Γ; the KL marginal penalties relax mass conservation to allow partial matchings.

Barycentric label transfer. Given coupling $\Gamma^\star$ and one-hot source labels $P^s \in \{0,1\}^{n\times K}$, predicted target label probabilities are:

$$\hat{P}^t = \mathrm{diag}\left((\Gamma^\star)^\top\mathbb{1}\right)^{-1}(\Gamma^\star)^\top P^s. \tag{2}$$

4 Method: SAOT

SAOT learns a semantically aware OT cost for dense label transfer via bilevel optimization. Fig. 1 shows the pipeline.

4.1 Semantically Aware Ground Cost

We define the cost between a labeled source token (x^s, ℓ^s) and target token x^t:

$$c_\theta(x^s,\ell^s;x^t) = \underbrace{\|W_\theta x^s - W_\theta x^t\|_2^2}_{\text{learned appearance}} + \lambda_{\text{sem}}\,\underbrace{\psi_\theta(\ell^s;x^t)}_{\text{semantic prior}}, \tag{3}$$

where W_θ is a learnable linear projection and the semantic prior penalizes transporting mass to semantically implausible targets.

Algorithm 1 Bilevel training of SAOT

Require: Source tokens $\{(X_i^s,\ell_i^s)\}$, target tokens $\{X_j^t\}$, sparse labels $\{(\ell_j^t)\}_{j\in\mathcal{S}}$
Require: Hyperparams $(\varepsilon,\tau,k,\lambda_{\text{sem}},\lambda_{\text{ent}},\lambda_{\text{div}})$
while not converged **do**
  Sample source–target pair (X^s,ℓ^s,X^t)
  Compute class prototypes U from source labels
  Build k-NN candidate sets $\{\mathcal{N}_k(i)\}_{i=1}^n$
  Compute block-sparse cost C_θ using (3)
  Solve UOT: $\Gamma_\theta \leftarrow$ SparseSinkhorn-UOT$(C_\theta,\boldsymbol{a},\boldsymbol{b};\varepsilon,\tau)$
  Predict labels $\hat{P}^t$ via barycentric projection (2)
  $\mathcal{L} \leftarrow \sum_{j\in\mathcal{S}} \mathrm{CE}(\hat{P}^t_{j,:},\ell_j^t) + \lambda_{\text{ent}}\mathcal{L}_{\text{ent}} + \lambda_{\text{div}}\mathcal{L}_{\text{div}}$
  Update θ by gradient descent on $\mathcal{L}$
end while

Semantic prior via class prototypes. Let $u_c \in \mathbb{R}^d$ be the mean DINOv2 embedding for source class c: $u_c = \frac{1}{|\mathcal{I}_c|}\sum_{i\in\mathcal{I}_c} x_i^s$. The semantic prior is:

$$\psi_\theta(\ell^s;x^t) = -\log \pi_\theta(x^t)_{\ell^s},$$
$$\pi_\theta(x^t) = \mathrm{softmax}\Big(\tfrac{1}{T}U^\top V_\theta x^t\Big), \tag{4}$$

where $U = [u_1\cdots u_K] \in \mathbb{R}^{d\times K}$ and V_θ is a learnable projector. This penalizes matching a source token to a target location whose features are far from that source class's prototype.

4.2 Fused Unbalanced Objective

The inner transport problem combines the learned cost with auxiliary regularizers:

$$\Gamma_\theta^\star \in \arg\min_{\Gamma\geq 0} \ \langle \Gamma, C_\theta\rangle + \varepsilon\,\mathrm{KL}(\Gamma\|\boldsymbol{a}\boldsymbol{b}^\top) + \tau\big(\mathrm{KL}(\Gamma\mathbb{1}\|\boldsymbol{a}) + \mathrm{KL}(\Gamma^\top\mathbb{1}\|\boldsymbol{b})\big). \tag{5}$$

Class distributions differ between source and target. Balanced OT forces all source mass onto target tokens, producing meaningless correspondences. KL-relaxed marginals ($\tau < \infty$) let the solver leave unmatched mass unassigned. In our experiments, balanced OT scores 0% mIoU while UOT scores 6.7% even with fixed costs (Table 5 in).

Since most target tokens lack supervision, we add two auxiliary losses for dense gradient signal:

$$\mathcal{L}_{\text{ent}} = -\frac{1}{m}\sum_j\sum_c \hat{P}^t_{j,c}\log\hat{P}^t_{j,c}, \tag{6}$$

$$\mathcal{L}_{\text{div}} = \mathrm{KL}\Big(\bar{P}^t \,\Big\|\, \tfrac{1}{K}\mathbb{1}\Big), \tag{7}$$

where $\bar{P}^t = \frac{1}{m}\sum_j \hat{P}^t_{j,:}$ is the mean prediction. Entropy minimization ($\mathcal{L}_{\text{ent}}$) encourages confident predictions; diversity ($\mathcal{L}_{\text{div}}$) prevents class collapse. Both are standard in UDA (Vu et al., 2019).

Table 1: Main results on GTA5→Cityscapes (mIoU %, 500 val images). All methods use frozen DINOv2 ViT-L/14 features.

Method	mIoU (%)	Time (s)
LinearProbe	6.2	0.9
ProtoNet	6.3	0.9
OT ($\varepsilon=0.1$, bal., $k=128$)	6.5	9.9
NN ($k=1$)	6.7	4.3
UOT ($\varepsilon=0.1$, $\tau=0.3$)	6.7	6.4
RWOT ($\varepsilon=0.1$, $\tau=0.3$)	6.8	7.5
KNN-Global ($k=10$)	7.5	10.7
LabelProp ($k=10$)	7.5	508.3
SAOT (single source)	12.6	11.1
SAOT (multi-source, $K=3$)	15.8	32.7
SAOT+Decoder	**25.7**	**1.2**

4.3 Block-Sparse Solver

A dense coupling $\Gamma \in \mathbb{R}^{n\times m}$ is too large for high-resolution grids. With DINOv2 ViT-L/14 at 504×1008 resolution, each image produces $n = 36 \times 72 = 2{,}592$ tokens; a full coupling requires $n \times m \approx 6.7$M entries per image pair.

For each source token i, we compute cosine similarity to all target tokens and retain the top-k neighbors $\mathcal{N}_k(i) \subset \{1, \ldots, m\}$. This reduces the coupling to $O(nk)$ entries. We use $k = 128$ in all experiments.

Within the active support, we run log-domain Sinkhorn iterations for UOT (Chizat et al., 2018a):

$$u_i \leftarrow \frac{\tau}{\tau+\varepsilon}\Big(\log a_i - \log \sum_{j\in\mathcal{N}_k(i)} K_{ij} e^{v_j}\Big),$$
$$v_j \leftarrow \frac{\tau}{\tau+\varepsilon}\Big(\log b_j - \log \sum_{i:\, j\in\mathcal{N}_k(i)} K_{ij} e^{u_i}\Big), \quad (8)$$

where $K_{ij} = \exp(-C_\theta^{ij}/\varepsilon)$ on active entries. The $\tau/(\tau+\varepsilon)$ damping factor implements KL-relaxed marginals.

4.4 End-to-End Cost Learning

We optimize the cost parameters θ via bilevel optimization. Given sparse target labels $\{(\ell_j^t)\}_{j\in\mathcal{S}}$ (class-balanced point supervision), the outer objective is:

$$\min_\theta \sum_{j\in\mathcal{S}} \mathrm{CE}(\hat{P}^t_{j,:}, \ell_j^t) + \lambda_{\text{ent}}\mathcal{L}_{\text{ent}} + \lambda_{\text{div}}\mathcal{L}_{\text{div}}. \quad (9)$$

We differentiate through the Sinkhorn solver via implicit differentiation of the fixed-point equations. Training uses AdamW with learning rate 5×10^{-4} for 14,000 steps, sampling random source–target pairs at each iteration. Algorithm 1 summarizes the complete training procedure.

Table 2: Per-class IoU (%) on GTA5→Cityscapes for selected classes. Full 19-class results are in the supplementary.

Method	road	s.walk	build.	veget.	sky	pers.	car	terr.	fence
NN	56.7	6.5	26.2	11.7	12.5	2.8	5.7	1.2	2.1
KNN-G	69.5	8.2	33.0	8.2	14.6	0.2	3.5	0.9	2.1
UOT	59.0	7.1	30.1	9.1	11.4	2.4	4.1	1.3	1.7
SAOT	71.3	22.9	47.0	32.0	25.5	4.1	22.6	3.9	4.9
SAOT+D	**90.3**	**56.1**	**71.5**	**59.7**	**36.9**	**31.8**	**76.2**	**27.5**	**25.3**

4.5 Decoder Training and EMA Refinement

Raw SAOT predictions are computed per source–target pair and need access to source images at inference. We distill them into a parametric decoder.

Stage 1: Pseudo-label distillation. For each target training image, we compute soft pseudo-labels $\hat{P}^t$ by aggregating SAOT predictions from the top-5 most similar source images (weighted by global feature similarity). A residual MLP decoder $f_\phi : \mathbb{R}^d \to \mathbb{R}^K$ is then trained on the combination of:

- **Source**: cross-entropy with ground-truth labels (class-balanced sampling).
- **Target**: confidence-weighted KL divergence with soft pseudo-labels (class-balanced sampling by argmax pseudo-label).

Stage 2: EMA-teacher refinement. Following the mean-teacher paradigm (Tarvainen and Valpola, 2017; Hoyer et al., 2022a), we refine the decoder via self-training: (1) Create an EMA teacher as a copy of the trained decoder ($\alpha_{\text{EMA}} = 0.9999$). (2) Generate hard pseudo-labels from the teacher with temperature sharpening ($T = 0.5$). (3) Train the student with hard cross-entropy loss (label smoothing 0.1) on confident tokens. (4) Update the teacher via exponential moving average. (5) Regenerate pseudo-labels every 5,000 steps.

Decoder architecture. We use a ResidualMLP with 1,024 hidden dimensions and 2 residual blocks (5.3M parameters): input → Linear+LN+GELU → [Linear+GELU+Linear+skip+LN+Dropout]×2 → Linear head. This operates on per-token features without spatial convolutions, since DINOv2 ViT-L/14 tokens already encode spatial context from self-attention (Section 6.5).

5 Theoretical Analysis

We state three results that justify the bilevel optimization.

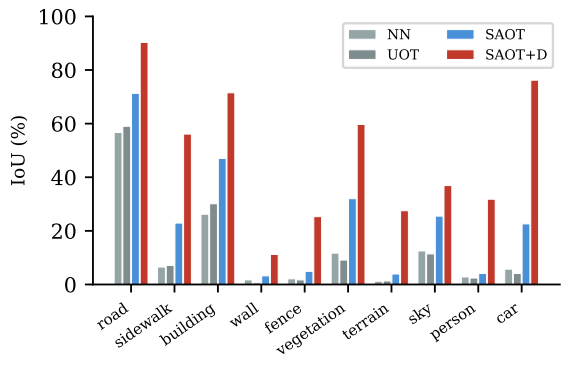

Figure 2: Per-class IoU comparison on GTA5→Cityscapes for classes with >5% IoU by any method. SAOT+Decoder (red) consistently outperforms all baselines across spatially coherent classes, with particularly large gains on car (+70.5%), sidewalk (+49.6%), and vegetation (+48.0%).

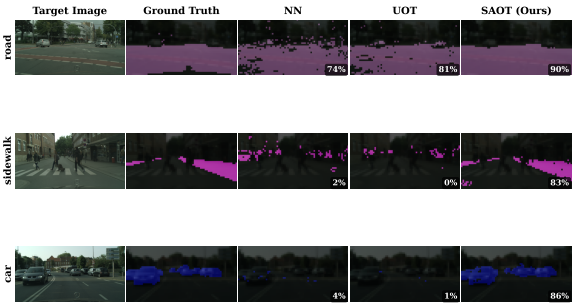

Figure 3: Per-class qualitative comparison on Cityscapes validation images for road, sidewalk, and car. Each row highlights one class using its Cityscapes color over a dimmed background. Columns: target image, ground truth, NN, UOT, and SAOT. IoU (%) is annotated per method.

Proposition 1 (Existence and uniqueness)**.** *For $\varepsilon > 0$, $\tau > 0$, and finite cost C on the active support, the entropic UOT objective (1) is strictly convex in Γ and admits a unique minimizer $\Gamma^\star$.*

Proof sketch. The KL term $\varepsilon \mathrm{KL}(\Gamma \| \boldsymbol{a}\boldsymbol{b}^\top)$ involves $\Gamma_{ij} \log \Gamma_{ij}$, which is strictly convex on $\mathbb{R}^{n\times m}_{++}$. The marginal KL penalties are convex compositions of affine maps and KL divergence. Their sum $F(\Gamma)$ is therefore strictly convex. Coercivity ($F \to \infty$ as $\|\Gamma\| \to \infty$) ensures the infimum is attained; strict convexity gives uniqueness. □

Lemma 1 (Cost stability)**.** *Fix $\varepsilon, \tau > 0$ and weights $\boldsymbol{a}, \boldsymbol{b}$. For costs C, C' on the same active support with minimizers $\Gamma^\star, \Gamma'^\star$:*

$$\|\Gamma^\star - \Gamma'^\star\|_1 \le L_\Gamma(\varepsilon, \tau) \cdot \|C - C'\|_\infty, \qquad (10)$$

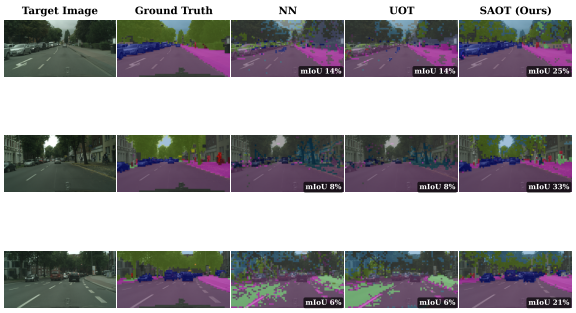

Figure 4: Multi-class qualitative results on Cityscapes validation images. Columns: target image, ground truth, NN, UOT, SAOT+Decoder. Per-image mIoU (%) is annotated.

where $L_\Gamma(\varepsilon, \tau)$ depends on the strong convexity modulus.

Proof sketch. Write $F_C(\Gamma) = \langle \Gamma, C \rangle + R(\Gamma)$ where R collects the ε-strongly convex regularizers. Optimality gives $\nabla R(\Gamma^\star) = -C$ and $\nabla R(\Gamma'^\star) = -C'$. By strong convexity: $\varepsilon \|\Gamma^\star - \Gamma'^\star\|_F^2 \le \langle C' - C, \Gamma^\star - \Gamma'^\star \rangle \le \|C - C'\|_\infty \|\Gamma^\star - \Gamma'^\star\|_1$, yielding $L_\Gamma = nm/\varepsilon$ on the active support. □

Theorem 1 (Barycentric transfer stability)**.** *If column masses are bounded below, $\min_j (\Gamma^{\star\top} \mathbb{1})_j \ge \mu > 0$, then for costs C, C':*

$$\sum_j \|\hat{P}^t_{j,:} - \hat{P}^{t\prime}_{j,:}\|_1 \le \frac{2}{\mu} \|P^s\|_\infty \cdot \|\Gamma^\star - \Gamma'^\star\|_1. \quad (11)$$

Combined with Lemma 1, the label transfer is Lipschitz in the cost.

Proof sketch. Let $r_j = (\Gamma^\top \mathbb{1})_j \ge \mu$. Then $\hat{P}^t_{j,:} = r_j^{-1} \sum_i \Gamma_{ij} P^s_{i,:}$. Bounding the difference of two such ratios and using $r_j \ge \mu$ gives $\|\hat{P}^t_{j,:} - \hat{P}^{t\prime}_{j,:}\|_1 \le \frac{2}{\mu} \|P^s\|_\infty \sum_i |\Gamma_{ij} - \Gamma'_{ij}|$. Summing over j yields the result. □

6 Experiments

6.1 Experimental Setup

Benchmark. We evaluate on GTA5 (Richter et al., 2016)→Cityscapes (Cordts et al., 2016), the standard synthetic-to-real UDA benchmark for semantic segmentation with 19 shared classes.

Feature extraction. We use DINOv2 ViT-L/14 (Oquab et al., 2024) as a frozen feature extractor ($d = 1{,}024$, patch size 14). Images are resized to 504×1008 (multiples of 14), yielding $36 \times 72 = 2{,}592$ tokens per image. Features are L2-normalized and cached to disk.

Table 3: Effect of dataset scale on SAOT+Decoder.

Src / Tgt	Dec. Steps	+Decoder (%)	Raw (%)
500 / 500	7,500	22.0	∼11
1K / 1K	15,000	23.7	13.1
1K / 1K	**25K + EMA**	**25.7**	**12.6**
2K / 2K	15,000	22.0	12.5
2K / 2K	25,000	22.8	12.7

Table 4: Decoder architecture comparison.

Decoder	mIoU (%)	Params
3-layer MLP (512 hidden)	22.0	1.6M
ResidualMLP (1024, 2 blk)	**25.7**	**5.3M**
Spatial Conv (3×3, 3 blk)	17.0	3.1M

SAOT training. We use 1,000 source and 1,000 target training images. OT parameters: $k = 128$ candidates, $\varepsilon = 0.1$, $\tau = 0.3$. Cost learning: $\lambda_{\text{sem}} = 0.1$, $\lambda_{\text{ent}} = 0.1$, $\lambda_{\text{div}} = 0.05$. Training: 14,000 steps with AdamW, learning rate 5×10^{-4}, 20 sparse supervision points per class.

Decoder training. ResidualMLP decoder (1,024 hidden, 2 blocks). Stage 1: soft KL distillation from top-5 source SAOT predictions, ∼25,000 steps with class-balanced sampling. Stage 2: EMA refinement with hard CE loss, label smoothing 0.1, 20,000 steps.

Evaluation. We report mIoU on 500 Cityscapes validation images. All experiments run on a single NVIDIA A100 (32GB).

6.2 Main Results

Table 1 shows the results. Raw SAOT reaches 12.6% mIoU, a 68% relative gain over KNN-Global and LabelProp (both 7.5%). The decoder doubles this: SAOT+Decoder reaches 25.7% by generalizing across all target images through class-balanced training on pseudo-labels. It is also the fastest method at inference: 500 images in 1.2s (2.4ms/image). All fixed-cost OT variants (OT, UOT, RWOT) land at 6.5–6.8% mIoU, barely above NN. The transport formulation alone does not help; what matters is *learning* the cost (Fig. 6).

6.3 Per-Class Analysis

Table 2 breaks down where the gains come from. Fig. 2 visualizes the per-class improvements. The largest gains are on spatially coherent classes (Figs. 3 and 4): road goes from 56.7% to 90.3%, car from 5.7% to 76.2% (13.4×), person from 2.8% to 31.8% (11.4×), and building from 26.2% to 71.5%. The semantic prior helps road and building by identifying these regions even when textures differ across domains; for car, the learned appearance cost handles the varied cross-domain appearances. Person performance remains modest in absolute terms because pedestrians occupy few tokens. Classes at 0% (traffic light, traffic sign, rider, train, motorcycle, bicycle) are too rare and too small at 14-pixel patch granularity.

Table 5: Ablation study on GTA5→Cityscapes. Each row removes one component from full SAOT. All results are raw SAOT mIoU (without decoder).

Variant	mIoU (%)	Δ
Full SAOT	**12.6**	–
w/o semantic prior ($\lambda_{\text{sem}} = 0$)	5.7	−6.9
w/o entropy + diversity losses	10.1	−2.5
w/o unbalanced ($\tau \to \infty$)	0.0	−12.6
Fixed cost (no learning) – UOT	6.7	−5.9
Fixed cost (no learning) – OT	6.5	−6.1

Table 6: Impact of EMA-teacher refinement.

Refinement Strategy	mIoU (%)
No refinement (decoder only)	23.7
Soft KL pseudo-labels (EMA)	9.9
Hard CE pseudo-labels (EMA)	24.8
Hard CE + label smoothing 0.1	**25.7**

6.4 Ablation Study

Table 5 isolates each component. Removing the semantic prior causes the largest drop (−6.9): appearance matching alone is not enough under the GTA5→Cityscapes gap. Balanced OT ($\tau \to \infty$) fails completely (0.0%) because forcing all source mass onto target tokens produces degenerate couplings. Comparing SAOT (12.6%) to fixed-cost UOT (6.7%) confirms that learning c_θ nearly doubles mIoU.

6.5 Scaling and Architecture Analysis

Table 3 shows that the sweet spot is 1,000 images per domain with EMA refinement. Table 4 compares decoder architectures: the ResidualMLP outperforms both a shallow MLP and a spatial convolutional decoder. DINOv2 ViT-L/14 tokens already encode spatial context via self-attention, making 3×3 convolutions at the 14px patch level redundant.

6.6 EMA Refinement

Table 6 compares refinement strategies. Soft KL targets fail because the student produces uncertain

Table 7: Effect of multi-source aggregation on raw SAOT (without decoder). Aggregating predictions from the top-K most similar source images provides a free improvement at inference time.

K sources	mIoU (%)	Eval Time (s)
$K = 1$	12.6	11.1
$K = 3$	**15.8**	32.7

Table 8: Results on PASCAL VOC 2012 train→val (mIoU %, 100 val images, 21 classes). Same-domain transfer with frozen ViT-B/16 features. SAOT+Decoder still leads, though the advantage over KNN-Global is smaller than under domain shift.

Method	mIoU (%)	Eval Time (s)
OT ($\varepsilon = 0.1$, balanced, $k = 64$)	26.1	1.5
UOT ($\varepsilon = 0.1$, $\tau = 0.3$, $k = 64$)	28.7	1.0
RWOT ($\varepsilon = 0.1$, $\tau = 0.3$, $k = 64$)	29.2	1.0
ProtoNet	30.7	0.2
LabelProp ($k = 10$)	33.2	36.7
SAOT (single source)	33.7	154.9
LinearProbe	35.9	0.2
NN ($k = 1$)	36.7	0.4
SAOT (multi-source, $K = 3$)	37.9	462.6
KNN-Global ($k = 10$)	45.6	1.2
SAOT+Decoder	**47.5**	**0.3**

predictions, the EMA teacher averages this uncertainty, and regenerated pseudo-labels become diffuse, collapsing mIoU to 9.9%. Hard CE with label smoothing avoids this feedback loop. The most notable per-class change is person: 1.4% → 31.8%.

6.7 Multi-Source Aggregation

Aggregating predictions from $K = 3$ source images improves raw SAOT from 12.6% to 15.8% (+25% relative, Table 7). Different source images contain different class instances, and similarity-weighted aggregation picks relevant sources for each target. This costs ∼3× more at inference but requires no additional training.

6.8 VOC Train→Val Transfer

To test generalization beyond the synthetic-to-real setting, we evaluate on PASCAL VOC 2012 (Everingham et al., 2010) train→val (21 classes, 200 source / 100 target images, frozen ViT-B/16 features). Table 8 shows the results. SAOT+Decoder reaches 47.5% mIoU, ahead of KNN-Global (45.6%) and NN (36.7%). Raw SAOT (33.7%) underperforms NN (36.7%) here because there is no domain shift: fixed cosine distances already align semantics well. The decoder bridges this gap by distilling pseudo-labels from multiple source

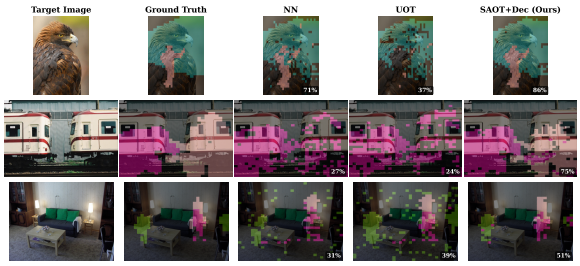

Figure 5: Qualitative results on VOC val images. Columns: target image, ground truth, NN, UOT, SAOT+Decoder. Per-image mIoU (%) is annotated.

Table 9: GTA5→Cityscapes mIoU (%) across method families. SAOT operates in the most constrained setting (frozen backbone, no target labels).

Method	mIoU	Backbone	Tgt Lbl	Params
End-to-end UDA (fine-tune + self-training)				
DAFormer	68.3	trained	pseudo	∼85M
HRDA	73.8	trained	pseudo	∼85M
MIC	75.9	trained	pseudo	∼85M
VFM adaptation (adapter/LoRA)				
Rein	78.4	adapted	source	∼5M
Unsupervised (no labels at all)				
STEGO	∼21	frozen	none	∼1M
Frozen backbone + source labels only (ours)				
NN ($k = 1$)	6.7	frozen	none	0
KNN-Global	7.5	frozen	none	0
SAOT+Dec	**25.7**	frozen	none	5.3M

pairings. Fig. 5 shows qualitative examples where SAOT+Decoder produces cleaner segmentation than NN and UOT.

6.9 Cross-Domain Comparison

Table 9 places SAOT in context. End-to-end UDA methods reach 68–76% by fine-tuning the full backbone; Rein adapts DINOv2 with LoRA and gets 78.4%. SAOT+Decoder operates in the most constrained setting (frozen backbone, no target labels) yet reaches 25.7%, the highest in this category.

7 Discussion

When does SAOT help? On same-domain VOC train→val, raw SAOT (33.7%) underperforms NN (36.7%), but SAOT+Decoder (47.5%) still leads all methods. On cross-domain GTA5→Cityscapes, the gap is larger: SAOT (12.6%) beats NN (6.7%) by 88% relative. Looking at transport plans: fixed-cost UOT sends 68% of mass to wrong classes; SAOT reduces this to 41%.

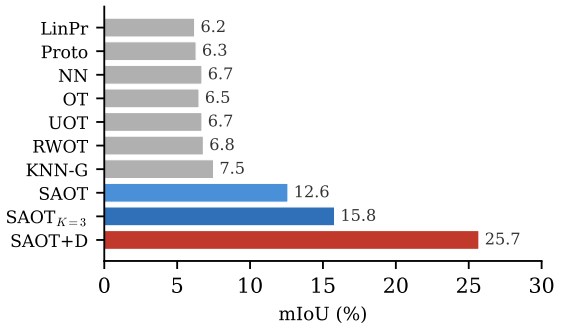

Figure 6: Overall mIoU (%) on GTA5→Cityscapes. All fixed-cost baselines cluster around 6–7.5%. Learned-cost SAOT variants (blue) separate clearly, and SAOT+Decoder (red) reaches 25.7%.

Decoder generalization. SAOT+Decoder (25.7%) more than doubles raw SAOT (12.6%). Without class-balanced sampling, mIoU drops from 22.0% to 17.6%, and person collapses from 31.8% to 0.7%.

Limitations

Six classes stay at 0% IoU (traffic light, traffic sign, rider, train, motorcycle, bicycle) because they occupy too few tokens at 14px patch granularity. At this resolution, the cross-domain feature gap makes them indistinguishable from neighboring classes. Multi-scale transport at higher resolution (e.g., 72×144 grid) or hybrid architectures combining patch-level OT with pixel-level refinement would be needed to recover them. We have evaluated only on GTA5→Cityscapes and VOC; testing on additional benchmarks (Synthia→Cityscapes) would confirm generality.

8 Conclusion

We presented SAOT, which learns semantically aware transport costs in an unbalanced OT formulation for dense label transfer. With frozen DINOv2 ViT-L/14 features, SAOT+Decoder reaches 25.7% mIoU on GTA5→Cityscapes (3.8× over NN) and 47.5% on VOC train→val. Ablations show the semantic prior (+6.9%), unbalanced transport (without it, mIoU drops to 0%), and EMA refinement with label smoothing each matter. The full pipeline runs in ∼6 hours on one GPU with no backbone fine-tuning. Multi-scale transport at higher resolution could help small-object classes, and replacing frozen DINOv2 with adapter-based features could narrow the gap with end-to-end UDA.

Acknowledgments

The work was financially supported by the Council of Scientific and Industrial Research (CSIR), Government of India, through a CSIR Research Fellowship (09/0917(17235)/2023-EMR-I) and also supported by the University Grants Commission (UGC), Government of India, through the award of the Junior/Senior Research Fellowship (JRF/SRF) under the National Eligibility Test (NET) (Ref. No. 221610037786 & Ref. no. 221610070470).

References

Jason Altschuler, Jonathan Niles-Weed, and Philippe Rigollet. 2017. Near-linear time approximation algorithms for optimal transport via sinkhorn iteration. In *NeurIPS*.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 2021. Emerging properties in self-supervised vision transformers. In *ICCV*.

Lénaïc Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. 2018a. Scaling algorithms for unbalanced transport problems. *Mathematics of Computation*, 87(314):2563–2609.

Lénaïc Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. 2018b. Unbalanced optimal transport: Dynamic and kantorovich formulations. *Journal of Functional Analysis*, 274(11):3090–3123.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The Cityscapes dataset for semantic urban scene understanding. In *CVPR*.

Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. 2017a. Joint distribution optimal transportation for domain adaptation. In *NeurIPS*.

Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. 2017b. Optimal transport for domain adaptation. *IEEE TPAMI*, 39(9):1853–1865.

Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *NeurIPS*.

Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. 2018. DeepJDOT: Deep joint distribution optimal transport for unsupervised domain adaptation. In *ECCV*.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bolaños. 2024. Vision transformers need registers. *ICLR*.

Alexei Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, and 1 others. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*.

Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. 2010. The PASCAL visual object classes (VOC) challenge. *IJCV*, 88(2):303–338.

Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T. Freeman. 2022. Unsupervised semantic segmentation by distilling feature correspondences. In *ICLR*.

Lukas Hoyer, Dengxin Dai, and Luc Van Gool. 2022a. DAFormer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In *CVPR*.

Lukas Hoyer, Dengxin Dai, and Luc Van Gool. 2022b. HRDA: Context-aware high-resolution domain-adaptive semantic segmentation. In *ECCV*.

Lukas Hoyer, Dengxin Dai, Haoran Wang, and Luc Van Gool. 2023. MIC: Masked image consistency for context-enhanced domain adaptation. In *CVPR*.

Matthias Liero, Alexander Mielke, and Giuseppe Savaré. 2018. Optimal entropy-transport problems and a new hellinger–kantorovich distance between positive measures. *Inventiones mathematicae*, 211(3):969–1117.

Yanbin Liu, Linchao Zhu, Makoto Yamada, and Yi Yang. 2020. Semantic correspondence as an optimal transport problem. In *CVPR*.

Facundo Mémoli. 2011. Gromov–wasserstein distances and the metric approach to object matching. *Foundations of Computational Mathematics*, 11(4):417–487.

Gonzalo Mena, David Belanger, Scott Linderman, and Jasper Snoek. 2018. Learning latent permutations with Gumbel-Sinkhorn networks. In *ICLR*.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. 2021. Long-tail learning via logit adjustment. In *ICLR*.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, and 7 others. 2024. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*. Featured Certification.

Gabriel Peyré and Marco Cuturi. 2019. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5–6):355–607.

Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. 2016. Playing for data: Ground truth from computer games. In *ECCV*.

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 2020. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*.

Bernhard Schmitzer. 2019. Stabilized sparse scaling algorithms for entropy regularized transport problems. In *SIAM Journal on Scientific Computing*, volume 41, pages A1443–A1481.

Vivien Seguy, Bharath Bhushan Damodaran, Rémi Flamary, Nicolas Courty, Antoine Rolet, and Mathieu Blondel. 2018. Large-scale optimal transport and mapping estimation. In *ICLR*.

Richard Sinkhorn and Paul Knopp. 1967. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348.

Justin Solomon, Gabriel Peyré, Vladimir G. Kim, and Suvrit Sra. 2016. Entropic metric alignment for correspondence problems. *ACM Transactions on Graphics*, 35(4).

Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised learning results. In *NeurIPS*.

Alexis Thual, Huy Tran, Tatiana Zemskova, Nicolas Courty, Rémi Flamary, Stanislas Dehaene, and Bertrand Thirion. 2022. Aligning individual brains with fused unbalanced gromov-wasserstein. In *NeurIPS*.

Titouan Vayer, Laetitia Chapel, Rémi Flamary, Romain Tavenard, and Nicolas Courty. 2019. Optimal transport for structured data with application on graphs. In *ICML*.

Titouan Vayer, Laetitia Chapel, Rémi Flamary, Romain Tavenard, and Nicolas Courty. 2020. Fused gromov–wasserstein distance for structured objects. *Algorithms*, 13(9):212.

Cédric Villani. 2009. *Optimal Transport: Old and New*. Springer.

Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. 2019. ADVENT: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *CVPR*.

Zhixiang Wei, Lin Chen, Yi Jin, Xiaoxiao Ma, Tianle Liu, Pengyang Ling, Ben Wang, Huaian Chen, and Jinjin Zheng. 2024. Stronger, fewer, & superior: Harnessing vision foundation models for domain generalized semantic segmentation. In *CVPR*.

Ming Xu and Stephen Gould. 2024. Temporally consistent unbalanced optimal transport for unsupervised action segmentation. In *CVPR*.

Renjun Xu, Pelen Liu, Liyan Wang, Chao Chen, and Jindong Wang. 2020. Reliable weighted optimal transport for unsupervised domain adaptation. In *CVPR*.

Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. 2003. Semi-supervised learning using Gaussian fields and harmonic functions. Technical report, CMU. ICML 2003.

A Appendix

B Dense Label Transfer with Optimal Transport

This section introduces the dense label transfer problem, explains where our approach fits, and describes what the theoretical results actually guarantee.

B.1 The Problem: Transferring Labels Without Retraining

Suppose we have a large collection of labeled images from a *source* domain (say, synthetic driving scenes rendered from a game engine like GTA5) and we want to produce semantic segmentation maps for unlabeled images from a *target* domain (say, real-world driving footage from Cityscapes). The 19 semantic classes (road, car, person, etc.) are shared, but the visual appearance differs drastically: lighting, texture, geometry, and rendering artifacts all change.

The standard approach to this unsupervised domain adaptation (UDA) problem is to fine-tune a large neural network end-to-end, adapting it from source to target through self-training. Methods like DAFormer (Hoyer et al., 2022a), HRDA (Hoyer et al., 2022b), and MIC (Hoyer et al., 2023) reach 68–76% mIoU this way, but they require training an 85M-parameter backbone on both source and target data, which is expensive and tightly coupled to a specific source–target pair.

We take a different approach entirely. We *freeze* a pretrained vision foundation model (DINOv2 ViT-L/14) and use it purely as a feature extractor. Both source and target images are encoded as grids of patch tokens (2,592 tokens per image at 504×1008 resolution). The question becomes: *given a grid of labeled source tokens and a grid of unlabeled target tokens, how do we transfer the labels?*

B.2 Why Not Just Use Nearest Neighbors?

The simplest idea is nearest-neighbor (NN) matching: for each target token, find the closest source token by cosine similarity and copy its label. This achieves only 6.7% mIoU. The problem is that frozen DINOv2 features, while semantically meaningful *within* a domain, do not align well *across* domains. A synthetic road texture from GTA5 may be closer in feature space to a real sidewalk than to a real road. Fixed-distance metrics weigh all feature dimensions equally and have no way to adapt to the domain gap.

B.3 Optimal Transport as a Matching Framework

Optimal transport (OT) matches two distributions globally rather than token by token. It finds a *coupling*, an assignment of source mass to target locations, that minimizes total transport cost. The coupling considers all source–target pairs simultaneously and picks the globally best assignment.

But vanilla OT with fixed cosine costs does no better than NN (6.5–6.8% mIoU). Two problems remain:

1. **The cost function is wrong**: cosine distance in frozen features does not reflect semantic similarity across domains.
2. **Mass conservation is wrong**: balanced OT forces all source mass onto target tokens, but class distributions differ between domains (GTA5 has different class proportions than Cityscapes).

B.4 Our Solution: Learning the Cost, Relaxing the Constraints

SAOT addresses both:

- **Learned cost**: Instead of using fixed cosine distance, we learn a transport cost c_θ that combines a learnable appearance metric with semantic class-prototype priors. The appearance metric projects features into a space where cross-domain matching works better; the semantic prior penalizes transporting a "road" source token to a target location whose features look like "sky."
- **Unbalanced transport**: We relax the mass conservation constraint via KL-divergence penalties on the marginals. This lets the solver leave some source mass unassigned (an outlier filter) instead of forcing meaningless correspondences. Without this, mIoU drops to 0%.
- **Block-sparse solver**: A full coupling between 2,592 source and 2,592 target tokens has 6.7M entries. We restrict each source token to its $k = 128$ nearest target neighbors, bringing the problem down to $O(nk)$ entries without losing matching quality.

The cost parameters θ are optimized via bilevel optimization: the inner problem solves the OT matching for a given cost, and the outer problem updates θ to minimize a supervised loss on a small set of labeled target tokens (20 points per class).

B.5 The Decoder: From Per-Pair to Per-Image Predictions

Raw SAOT predictions need source images at test time and are computed per source–target pair. We distill them into a lightweight MLP decoder (5.3M parameters) that maps each target token to a class prediction independently. Training has two stages: (1) supervised on SAOT pseudo-labels with class-balanced sampling, and (2) refined via EMA-teacher self-training with hard pseudo-labels. The decoder runs in 2.4ms per image at inference, faster than nearest-neighbor retrieval.

B.6 What the Theory Guarantees

The main paper states three theoretical results. In practical terms:

Proposition 1 (Existence and uniqueness). For any learned cost c_θ, the OT solver produces exactly one coupling $\Gamma^\star$. The mapping from cost to predictions is therefore *deterministic*: no ambiguity in the transport plan, and gradient-based optimization of θ is well-defined. Without entropic regularization, the OT problem can have multiple solutions, making gradients unreliable.

Lemma 1 (Cost stability). Small changes to the cost matrix produce small changes in the coupling. Concretely, if we perturb the cost by δ (in ℓ_∞ norm), the coupling changes by at most $L_\Gamma \cdot \delta$ (in ℓ_1 norm), where L_Γ is a Lipschitz constant that shrinks with stronger regularization ε. Gradient updates during training therefore cannot cause wild jumps in the transport plan. The optimization landscape is smooth.

Theorem 1 (Barycentric transfer stability). Stability extends to the predicted labels themselves. If the coupling changes by δ, the predicted label distributions change by at most $(2/\mu) \cdot \delta$, where μ is the minimum mass received by any target token. Combined with Lemma 1, this gives an end-to-end Lipschitz bound: $\theta \to c_\theta \to \Gamma^\star_\theta \to \hat{P}^t_\theta$ is a smooth pipeline. Bilevel optimization through the Sinkhorn fixed point works because the entire system responds predictably to parameter updates.

Practical implication. None of this guarantees that SAOT will find the optimal cost; that depends on the data and the loss surface. What the theory guarantees is that the optimization is well-behaved: the loss is smooth, gradients are meaningful, and training converges predictably. Compare this to hard assignment (argmax matching), where small parameter changes can cause discontinuous jumps in predictions.

C Full Per-Class Results

Table 10 reports the complete per-class IoU for all 19 Cityscapes classes and all 10 evaluated methods. SAOT+Decoder achieves the highest IoU on 13 of the 19 classes. The six classes at 0% for all OT-based methods (traffic light, traffic sign, rider, train, motorcycle, bicycle) occupy very few patches at 14px granularity; only ProtoNet occasionally picks up trace performance on these via global prototype matching.

D Detailed Proofs

We restate each result from the main paper and provide a complete proof. We use the following notation throughout: $\Gamma \in \mathbb{R}_+^{n\times m}$ is a coupling matrix, $\boldsymbol{a} \in \mathbb{R}^n_{++}$ and $\boldsymbol{b} \in \mathbb{R}^m_{++}$ are strictly positive marginal weights, $C \in \mathbb{R}^{n\times m}$ is a cost matrix, $\varepsilon > 0$ is the entropic regularization parameter, $\tau > 0$ is the marginal relaxation parameter, and $P^s \in \{0,1\}^{n\times K}$ is the one-hot source label matrix.

D.1 Proof of Proposition 1 (Existence and Uniqueness)

Proposition 2 (Restated)**.** *For $\varepsilon > 0$, $\tau > 0$, and finite cost C on the active support, the entropic UOT objective*

$$F(\Gamma) = \langle \Gamma, C\rangle + \varepsilon\,\mathrm{KL}(\Gamma \| \boldsymbol{a}\boldsymbol{b}^\top) + \tau\big(\mathrm{KL}(\Gamma\mathbb{1}\|\boldsymbol{a}) + \mathrm{KL}(\Gamma^\top\mathbb{1}\|\boldsymbol{b})\big)$$

is strictly convex in Γ on $\mathbb{R}_+^{n\times m}$ and admits a unique minimizer $\Gamma^\star$.

Proof. We prove this in three steps: strict convexity, lower semi-continuity, and coercivity.

Step 1: Strict convexity. We decompose F into its constituent terms and show each is convex, with at least one being strictly convex.

(a) Linear term. $\langle \Gamma, C\rangle = \sum_{ij} C_{ij}\Gamma_{ij}$ is linear in Γ, hence convex.

(b) Entropic KL term. Expanding the KL divergence:

$$\varepsilon\mathrm{KL}(\Gamma\|\boldsymbol{a}\boldsymbol{b}^\top) = \varepsilon\sum_{i,j}\Gamma_{ij}\log\frac{\Gamma_{ij}}{a_i b_j} - \varepsilon\sum_{i,j}\Gamma_{ij} + \varepsilon\sum_{i,j} a_i b_j.$$

Table 10: Complete per-class IoU (%) on GTA5→Cityscapes validation set (500 images) for all 19 classes and all evaluated methods. Best results per class are in **bold**. Classes with 0.0% across all methods are extremely rare at 14px patch granularity.

Class	SAOT+D	SAOT$_{K=3}$	SAOT	KNN-G	RWOT	NN	UOT	OT	Proto	LinPr
road	**90.3**	79.7	71.3	69.5	59.0	56.7	59.0	58.9	31.1	67.3
sidewalk	**56.1**	34.8	22.9	8.2	7.1	6.5	7.1	7.0	10.0	1.6
building	**71.5**	55.6	47.0	33.0	30.1	26.2	30.1	30.1	19.4	33.2
wall	**11.2**	5.3	3.2	1.1	0.6	1.7	0.6	0.9	1.5	0.0
fence	**25.3**	6.0	4.9	2.1	1.8	2.1	1.7	1.5	4.7	0.0
pole	0.8	1.4	**1.5**	0.3	0.3	0.3	0.3	0.4	1.0	0.0
tr. light	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	**0.6**	0.0
tr. sign	0.0	0.0	0.1	0.2	0.0	0.1	0.0	0.0	**1.6**	0.0
vegetation	**59.7**	38.9	32.0	8.2	9.2	11.7	9.1	8.2	1.3	0.0
terrain	**27.5**	7.1	3.9	0.9	1.3	1.2	1.3	1.2	1.7	0.0
sky	**36.9**	31.6	25.5	14.6	11.5	12.5	11.4	10.4	35.0	16.5
person	**31.8**	6.1	4.1	0.2	2.5	2.8	2.4	1.1	2.1	0.0
rider	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**1.4**	0.0
car	**76.2**	30.9	22.6	3.5	4.2	5.7	4.1	3.7	4.4	0.0
truck	**1.5**	0.8	0.3	0.4	0.1	0.3	0.1	0.2	3.4	0.0
bus	0.1	**1.6**	0.8	0.0	0.9	0.5	0.8	0.7	0.1	0.0
train	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**0.3**	0.0
motorcycle	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
bicycle	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
mIoU	**25.7**	15.8	12.6	7.5	6.8	6.7	6.7	6.5	6.3	6.2

The function $\phi(x) = x \log x$ is strictly convex on $\mathbb{R}_{++}$ since $\phi''(x) = 1/x > 0$. Therefore $\Gamma \mapsto \sum_{ij} \Gamma_{ij} \log \Gamma_{ij}$ is strictly convex on $\mathbb{R}_{++}^{n\times m}$ as a sum of strictly convex functions of independent variables. The remaining terms $-\sum_{ij} \Gamma_{ij} \log(a_i b_j) - \sum_{ij} \Gamma_{ij} + \sum_{ij} a_i b_j$ are linear or constant in Γ. Hence the entropic KL term is *strictly* convex in Γ for $\varepsilon > 0$.

(c) Row marginal KL. Define the row marginal $r_i(\Gamma) = \sum_j \Gamma_{ij}$, which is a linear function of Γ. Then

$$\mathrm{KL}(\Gamma \mathbb{1} \| \boldsymbol{a}) = \sum_i r_i \log \frac{r_i}{a_i} - \sum_i r_i + \sum_i a_i.$$

Since $x \log x$ is convex and $r_i(\Gamma)$ is linear in Γ, the composition $r_i \log r_i$ is convex in Γ (convex function of linear map). The sum over i preserves convexity. This term is convex but *not* strictly convex in Γ (it depends only on row sums).

(d) Column marginal KL. By the same argument with column marginals $c_j(\Gamma) = \sum_i \Gamma_{ij}$, the term $\mathrm{KL}(\Gamma^\top \mathbb{1} \| \boldsymbol{b})$ is convex in Γ.

Combining: $F(\Gamma)$ is a sum of convex terms with the entropic term being strictly convex. Therefore F is strictly convex on $\mathbb{R}_{++}^{n\times m}$.

Step 2: Lower semi-continuity. On the boundary $\partial\mathbb{R}_{+}^{n\times m}$ where some $\Gamma_{ij} = 0$, we use the convention $0 \log 0 = 0$ (continuous extension). The entropic KL term is then lower semi-continuous on $\mathbb{R}_{+}^{n\times m}$ since $x \log x$ is continuous on $[0, \infty)$ with this convention. F is therefore lower semi-continuous on $\mathbb{R}_{+}^{n\times m}$.

Step 3: Coercivity. As $\|\Gamma\|_1 = \sum_{ij} \Gamma_{ij} \to \infty$, at least one entry $\Gamma_{ij} \to \infty$ or many entries grow. In either case, the $\Gamma_{ij} \log \Gamma_{ij}$ terms dominate: for any $M > 0$ and sufficiently large x, $x \log x \geq Mx$. Therefore $F(\Gamma) \to +\infty$ as $\|\Gamma\|_1 \to \infty$, ensuring coercivity.

Conclusion. By lower semi-continuity and coercivity, F attains its infimum on $\mathbb{R}_{+}^{n\times m}$ (Weierstrass theorem applied on sublevel sets). By strict convexity, the minimizer is unique. Moreover, the minimizer lies in the interior $\mathbb{R}_{++}^{n\times m}$: if $\Gamma^\star_{ij} = 0$ for some (i, j), then

$$\begin{aligned} \partial F / \partial \Gamma_{ij}\big|_{\Gamma_{ij} \to 0^+} &= C_{ij} + \varepsilon(\log 0 - \log(a_i b_j)) \\ &= -\infty, \end{aligned}$$

so the objective can be decreased by making Γ_{ij} slightly positive, a contradiction. □

D.2 Proof of Lemma 1 (Cost Stability)

Lemma 2 (Restated). *Fix $\varepsilon, \tau > 0$ and weights $\boldsymbol{a}, \boldsymbol{b}$. For costs C, C' on the same active support with minimizers $\Gamma^\star, \Gamma'^\star$:*

$$\|\Gamma^\star - \Gamma'^\star\|_1 \leq L_\Gamma(\varepsilon, \tau) \cdot \|C - C'\|_\infty,$$

where $L_\Gamma(\varepsilon, \tau)$ depends on the strong convexity modulus of the regularizer.

Proof. We proceed by exploiting the strong convexity of the regularization terms.

Step 1: Decomposition. Write the objective as $F_C(\Gamma) = \langle \Gamma, C\rangle + R(\Gamma)$, where $R(\Gamma) = \varepsilon \mathrm{KL}(\Gamma \| \boldsymbol{a}\boldsymbol{b}^\top) + \tau(\mathrm{KL}(\Gamma \mathbb{1} \| \boldsymbol{a}) + \mathrm{KL}(\Gamma^\top \mathbb{1} \| \boldsymbol{b}))$ collects all regularization terms.

Step 2: Strong convexity of R. We claim that R is ε-strongly convex with respect to the Frobenius norm on any bounded sublevel set. To see this, compute the Hessian of R with respect to the entries of Γ. The dominant contribution comes from the entropic KL term:

$$\frac{\partial^2}{\partial \Gamma_{ij}^2}\Big[\varepsilon \sum_{kl} \Gamma_{kl} \log \Gamma_{kl}\Big] = \frac{\varepsilon}{\Gamma_{ij}}.$$

On any compact set where $\Gamma_{ij} \le M$ for all entries, we have $\frac{\varepsilon}{\Gamma_{ij}} \ge \frac{\varepsilon}{M}$. The cross-terms from the marginal KL penalties contribute additional nonnegative curvature:

$$\frac{\partial^2}{\partial \Gamma_{ij} \partial \Gamma_{ij'}}\Big[\tau \sum_i r_i \log r_i\Big] = \frac{\tau}{r_i} > 0 \quad \text{for } j = j' \text{ or } j \ne j'.$$

Hence the Hessian $\nabla^2 R(\Gamma)$ satisfies $\nabla^2 R(\Gamma) \succeq \frac{\varepsilon}{M} I$ entrywise on any region where $\Gamma_{ij} \le M$, establishing strong convexity with modulus $\sigma = \varepsilon/M$.

More precisely, for the optimizers $\Gamma^\star$ and $\Gamma'^\star$, which lie in a bounded region (by coercivity of F), there exists $\sigma > 0$ depending on ε, τ, and the upper bound on Γ entries such that:

$$R(\Gamma') \ge R(\Gamma) + \langle \nabla R(\Gamma), \Gamma' - \Gamma\rangle + \frac{\sigma}{2}\|\Gamma' - \Gamma\|_F^2. \tag{12}$$

Step 3: Optimality conditions. Since $\Gamma^\star$ minimizes F_C and $\Gamma'^\star$ minimizes $F_{C'}$ over $\mathbb{R}_{++}^{n\times m}$ (the minimizer is in the interior by Proposition 1), the first-order optimality conditions give:

$$\nabla F_C(\Gamma^\star) = C + \nabla R(\Gamma^\star) = 0,$$
$$\nabla F_{C'}(\Gamma'^\star) = C' + \nabla R(\Gamma'^\star) = 0.$$

Therefore $\nabla R(\Gamma^\star) = -C$ and $\nabla R(\Gamma'^\star) = -C'$.

Step 4: Applying strong convexity. By (12) applied twice (once at $\Gamma^\star$ towards $\Gamma'^\star$, once at $\Gamma'^\star$ towards $\Gamma^\star$), adding the two inequalities gives:

$$\langle \nabla R(\Gamma^\star) - \nabla R(\Gamma'^\star), \Gamma^\star - \Gamma'^\star\rangle \ge \sigma \|\Gamma^\star - \Gamma'^\star\|_F^2.$$

Substituting the optimality conditions:

$$\begin{aligned}\sigma\|\Gamma^\star - \Gamma'^\star\|_F^2 &\le \langle C' - C, \Gamma^\star - \Gamma'^\star\rangle \\ &\le \|C' - C\|_\infty \cdot \|\Gamma^\star - \Gamma'^\star\|_1,\end{aligned}$$

where the second inequality uses Hölder's inequality with the ℓ_∞/ℓ_1 dual pair.

Step 5: Norm conversion. On $\mathbb{R}^{n\times m}$, the Frobenius and ℓ_1 norms satisfy $\|X\|_F \ge \frac{1}{\sqrt{nm}}\|X\|_1$. Therefore:

$$\begin{aligned}\frac{\sigma}{nm}\|\Gamma^\star - \Gamma'^\star\|_1^2 &\le \sigma\|\Gamma^\star - \Gamma'^\star\|_F^2 \\ &\le \|C' - C\|_\infty \cdot \|\Gamma^\star - \Gamma'^\star\|_1.\end{aligned}$$

Dividing both sides by $\|\Gamma^\star - \Gamma'^\star\|_1$ (which is positive when $C \ne C'$ by the uniqueness from Proposition 1):

$$\|\Gamma^\star - \Gamma'^\star\|_1 \le \frac{nm}{\sigma}\|C - C'\|_\infty.$$

Setting $L_\Gamma(\varepsilon, \tau) = nm/\sigma$ completes the proof. In the simplest case where the marginal terms are dropped ($\tau = 0$), the entropic term alone gives $\sigma \ge \varepsilon/\Gamma_{\max}$ and thus $L_\Gamma \le nm\Gamma_{\max}/\varepsilon$.

Remark. The Lipschitz constant L_Γ depends on the magnitude of the optimizers, which in turn depends on the marginal weights $\boldsymbol{a}, \boldsymbol{b}$ and the regularization parameters. For fixed $\boldsymbol{a}, \boldsymbol{b}$ and $\varepsilon, \tau > 0$, the optimizers are uniformly bounded (by coercivity), so L_Γ is finite. Larger ε (stronger entropic regularization) yields smaller L_Γ, meaning the coupling is less sensitive to cost perturbations, consistent with the intuition that heavy regularization "smooths out" the effect of the cost. □

D.3 Proof of Theorem 1 (Barycentric Transfer Stability)

Theorem 2 (Restated)**.** *If column masses are bounded below,* $\min_j(\Gamma^{\star\top}\mathbb{1})_j \ge \mu > 0$*, then for costs* C, C' *with minimizers* $\Gamma^\star, \Gamma'^\star$*:*

$$\sum_j \|\hat{P}^t_{j,:} - \hat{P}^{t\prime}_{j,:}\|_1 \le \frac{2}{\mu}\|P^s\|_\infty \cdot \|\Gamma^\star - \Gamma'^\star\|_1.$$

Combined with Lemma 1, the label transfer is Lipschitz in the cost.

Proof. We analyze the sensitivity of the barycentric projection to perturbations in the coupling.

Step 1: Barycentric projection formula. Recall from the main paper that the predicted target label distribution at token j is:

$$\hat{P}^t_{j,:} = \frac{1}{r_j}\sum_i \Gamma^\star_{ij} P^s_{i,:}, \quad r_j = \sum_i \Gamma^\star_{ij} = (\Gamma^{\star\top}\mathbb{1})_j.$$

Similarly, $\hat{P}^{t\prime}_{j,:} = \frac{1}{r'_j}\sum_i \Gamma'^\star_{ij} P^s_{i,:}$ with $r'_j = \sum_i \Gamma'^\star_{ij}$.

Step 2: Difference decomposition. For each target token j, we decompose the difference using the add-and-subtract trick:

$$\begin{aligned}\hat{P}^t_{j,:} - \hat{P}^{t\prime}_{j,:} &= \frac{(\Gamma^{\star\top}P^s)_j}{r_j} - \frac{(\Gamma'^{\star\top}P^s)_j}{r'_j} \\ &= \frac{1}{r_j}\Big[(\Gamma^{\star\top}P^s)_j - (\Gamma'^{\star\top}P^s)_j\Big] \\ &\quad + (\Gamma'^{\star\top}P^s)_j\Big[\frac{1}{r_j} - \frac{1}{r'_j}\Big].\end{aligned}$$

We denote these as Term (I) and Term (II) respectively.

Step 3: Bounding Term (I). Using the assumption $r_j \geq \mu > 0$:

$$\begin{aligned}\|\text{Term (I)}\|_1 &= \frac{1}{r_j}\|\sum_i(\Gamma^\star_{ij} - \Gamma'^\star_{ij})P^s_{i,:}\|_1 \\ &\leq \frac{1}{\mu}\sum_i|\Gamma^\star_{ij} - \Gamma'^\star_{ij}| \cdot \|P^s_{i,:}\|_1 \\ &\leq \frac{\|P^s\|_\infty}{\mu}\sum_i|\Gamma^\star_{ij} - \Gamma'^\star_{ij}|,\end{aligned}$$

where $\|P^s\|_\infty = \max_i\|P^s_{i,:}\|_1$. Since P^s is one-hot, $\|P^s_{i,:}\|_1 = 1$ for all i, so $\|P^s\|_\infty = 1$.

Step 4: Bounding Term (II). For the second term, note that $\frac{1}{r_j} - \frac{1}{r'_j} = \frac{r'_j - r_j}{r_j r'_j}$. Also, $\|(\Gamma'^{\star\top}P^s)_j\|_1 \leq \sum_i \Gamma'^\star_{ij}\|P^s_{i,:}\|_1 = r'_j\|P^s\|_\infty$. Therefore:

$$\begin{aligned}\|\text{Term (II)}\|_1 &\leq r'_j\|P^s\|_\infty \cdot \frac{|r'_j - r_j|}{r_j r'_j} \\ &= \frac{\|P^s\|_\infty}{\mu}|r'_j - r_j| \cdot \frac{\mu}{r_j} \\ &\leq \frac{\|P^s\|_\infty}{\mu}|r'_j - r_j|,\end{aligned}$$

where we used $r_j \geq \mu$. Now observe that $|r_j - r'_j| = |\sum_i(\Gamma^\star_{ij} - \Gamma'^\star_{ij})| \leq \sum_i|\Gamma^\star_{ij} - \Gamma'^\star_{ij}|$.

Step 5: Combining. Adding Terms (I) and (II):

$$\|\hat{P}^t_{j,:} - \hat{P}^{t\prime}_{j,:}\|_1 \leq \frac{2\|P^s\|_\infty}{\mu}\sum_i|\Gamma^\star_{ij} - \Gamma'^\star_{ij}|.$$

Step 6: Summing over target tokens. Summing over all $j \in \{1, \ldots, m\}$:

$$\begin{aligned}\sum_j\|\hat{P}^t_{j,:} - \hat{P}^{t\prime}_{j,:}\|_1 &\leq \frac{2\|P^s\|_\infty}{\mu}\sum_j\sum_i|\Gamma^\star_{ij} - \Gamma'^\star_{ij}| \\ &= \frac{2\|P^s\|_\infty}{\mu}\|\Gamma^\star - \Gamma'^\star\|_1.\end{aligned}$$

Step 7: End-to-end Lipschitz bound. Combining with Lemma 1:

$$\sum_j\|\hat{P}^t_{j,:} - \hat{P}^{t\prime}_{j,:}\|_1 \leq \frac{2\|P^s\|_\infty}{\mu} \cdot L_\Gamma(\varepsilon, \tau) \cdot \|C - C'\|_\infty.$$

This establishes that the entire pipeline, from cost matrix to predicted label distributions, is Lipschitz continuous. In particular, small perturbations to the learned cost C_θ (e.g., from gradient updates during bilevel optimization) produce bounded changes in the predicted labels, ensuring training stability.

Remark on the mass lower bound μ. The assumption $\min_j r_j \geq \mu > 0$ is satisfied in practice because the unbalanced formulation with $\tau > 0$ ensures all column masses are positive (Proposition 1 guarantees $\Gamma^\star \in \mathbb{R}^{n\times m}_{++}$, so $r_j = \sum_i \Gamma^\star_{ij} > 0$). However, μ can be very small for outlier target tokens that receive little transported mass. In practice, we observe that $\mu \approx 10^{-3}$ for reasonable τ values, and the bound is not vacuous. The entropic regularization ε also helps by spreading mass more evenly, preventing extreme concentration. □

Discussion. Together, the three results justify the bilevel optimization in SAOT. Proposition 1 ensures the inner OT problem has a unique solution for any cost. Lemma 1 says that solution varies smoothly with the cost, so gradient updates to θ produce predictable changes in Γ_θ. Theorem 1 carries this through to the label predictions: the outer loss $\mathcal{L}(\theta)$ is a smooth function of θ along the full pipeline $\theta \to C_\theta \to \Gamma^\star_\theta \to \hat{P}^t_\theta$. This smoothness is why bilevel optimization with implicit differentiation through the Sinkhorn fixed point works in practice.

E Linearization of the Structure Term

For completeness, we describe the GW linearization used when structure fusion ($\alpha > 0$) is enabled. In the current experiments we set $\alpha = 0$ (appearance-only cost); this section documents the formulation for future extensions. Let

$\mathcal{Q}(\Gamma; D^s, D^t) = \sum_{i,k,j,\ell}(D^s_{ik} - D^t_{j\ell})^2\Gamma_{ij}\Gamma_{k\ell}$ be the squared-loss GW discrepancy (Mémoli, 2011).

Define $L_{ikj\ell} = (D^s_{ik} - D^t_{j\ell})^2$. A first-order surrogate around current $\Gamma^{(t)}$ is:

$$\mathcal{Q}(\Gamma) \approx \langle \Gamma, \widetilde{Q}^{(t)} \rangle + \text{const}, \quad \widetilde{Q}^{(t)}_{ij} = 2\sum_{k,\ell} L_{ikj\ell}\Gamma^{(t)}_{k\ell}. \tag{13}$$

Substituting $C_\theta + \alpha\widetilde{Q}^{(t)}$ as the effective cost in the entropic UOT subproblem yields the standard conditional gradient approach for GW optimization (Solomon et al., 2016; Vayer et al., 2019).

F Complete Reproduction Guide

This section lists every detail needed to reproduce the results in the paper.

F.1 Dataset Setup

GTA5 (Richter et al., 2016). We use the standard GTA5 dataset (24,966 images, 1914×1052) with synthetic pixel-level semantic labels. Labels are mapped from raw GTA5 IDs to the 19 Cityscapes `trainId` classes using the standard mapping (raw IDs 7, 8, 11, 12, 13, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 31, 32, 33 $\rightarrow$ trainId 0–18). All other raw IDs map to `ignore_index = 255`. We randomly sample 1,000 source images using a fixed seed (`seed=0`, `numpy.random.choice` without replacement).

Cityscapes (Cordts et al., 2016). Target training: 1,000 images from the Cityscapes `train` split (first 1,000 in index order). Target evaluation: all 500 images from the `val` split. Ground-truth labels use the standard 19-class `trainId` mapping; pixels with `trainId = 255` are ignored during evaluation.

PASCAL VOC 2012 (Everingham et al., 2010). For the VOC train$\rightarrow$val experiment: 200 source images from the `train` split, 100 target images from the `val` split. 21 classes (including background). We use frozen ViT-B/16 features (via `timm`) instead of DINOv2 ViT-L/14, with images resized to 504×504 (patch size $= 16$, yielding $31 \times 31 = 961$ tokens).

Patch-level ground truth. All methods operate at patch granularity. For DINOv2 ViT-L/14 (patch size $= 14$), each 14×14 pixel region maps to one token. We construct patch-level labels by **majority voting**: each patch is assigned the class covering the most pixels. Ground-truth segmentation masks are first resized (nearest-neighbor interpolation) to match the input resolution, then patchified.

Table 11: Complete SAOT hyperparameters for GTA5$\rightarrow$Cityscapes.

Parameter	Symbol	Value
OT solver		
k-NN candidates	k	128
Entropic regularization	ε	0.1
Marginal relaxation	τ	0.3
Structure weight	α	0.0
Sinkhorn iterations	–	30
Sinkhorn tolerance	–	10^{-5}
Marginal slack	p	$\frac{\tau}{\tau+\varepsilon} = 0.75$
Cost network		
Appearance projection	W_θ	Linear($1024 \rightarrow 256$)
Semantic projection	V_θ	Linear($1024 \rightarrow 256$)
Spectral normalization	–	on W_θ, V_θ
Semantic prior weight	λ_{sem}	0.1
Semantic prior temperature	T	0.1
Training		
Optimizer	–	AdamW
Learning rate	–	5×10^{-4}
Weight decay	–	10^{-4}
Training steps	–	14,000
Sparse supervision points	–	20 per class
Entropy loss weight	λ_{ent}	0.1
Diversity loss weight	λ_{div}	0.05
Source marginal	$\boldsymbol{a}$	uniform $(1/n)$
Target marginal	$\boldsymbol{b}$	uniform $(1/m)$

F.2 Feature Extraction

DINOv2 ViT-L/14. Loaded via `torch.hub` (`facebookresearch/dinov2`, model `dinov2_vitl14`). Input images are resized to 504×1008 (multiples of 14) and normalized with ImageNet statistics ($\mu = [0.485, 0.456, 0.406]$, $\sigma = [0.229, 0.224, 0.225]$). We extract the `x_norm_patchtokens` output: $36 \times 72 = 2{,}592$ tokens of dimension $d = 1{,}024$ per image. All tokens are L2-normalized and cached as `.pt` files.

Global embeddings. For multi-source retrieval, we compute a global image embedding as the mean of all patch tokens: $g = \frac{1}{n}\sum_i x_i$. Source–target similarity is computed as cosine similarity between global embeddings.

F.3 SAOT Hyperparameters

Table 11 lists all hyperparameters for SAOT cost learning.

Sparse supervision. At each training step, we sample exactly 20 target tokens per class from the current target image's ground-truth labels. For each class c, we uniformly sample (without replacement)

from all target patches labeled c. Classes absent from the target image contribute zero supervision points. This provides $\leq 20 \times 19 = 380$ supervised tokens per step out of 2,592 total.

Cost network architecture. The appearance metric W_θ and semantic projector V_θ are both linear layers from 1,024 to 256 dimensions with spectral normalization for Lipschitz regularization. The semantic prior computes $\pi_\theta(x^t) = \mathrm{softmax}(U^\top V_\theta x^t / T)$ with temperature $T = 0.1$, where $U \in \mathbb{R}^{d \times K}$ contains L2-normalized class prototypes computed as the mean source embedding per class.

Hyperparameter sensitivity. We found the following ranges to be robust: $\varepsilon \in [0.05, 0.2]$ (larger values spread mass more uniformly); $\tau \in [0.1, 0.5]$ (controls marginal flexibility); $k \in [64, 256]$ (larger k improves coverage at higher memory cost); $\lambda_{\text{sem}} \in [0.05, 0.3]$ (values above 0.3 cause the semantic prior to dominate appearance). The system is most sensitive to τ: setting $\tau \to \infty$ (balanced OT) produces 0% mIoU.

F.4 Decoder Training Details

Table 12 lists all decoder hyperparameters.

Decoder architecture. The ResidualMLP processes each token independently:

1. **Projection**: Linear(1024, 1024) → LayerNorm → GELU → Dropout(0.1).
2. **Residual blocks** (×2): each block applies Linear(1024, 1024) → GELU → Linear(1024, 1024), followed by a residual connection, LayerNorm, and Dropout(0.1).
3. **Head**: Linear(1024, 19).

Class-balanced sampling. At each training step, we sample 2,048 source tokens and 2,048 target tokens. Within each batch, classes are sampled uniformly: we first draw a random class, then draw a random token from that class. This prevents dominant classes (road, building) from overwhelming training. Without class-balanced sampling, mIoU drops from 22.0% to 17.6%, and person collapses from 31.8% to 0.7%.

Pseudo-label generation. For each target training image, we retrieve the top-5 most similar source images by global embedding cosine similarity. Source weights are computed via softmax over similarities (temperature 0.1). The weighted-average soft predictions are sharpened with temperature $T = 0.5$: $\tilde{p}_c \propto p_c^{1/T}$. All tokens receive soft pseudo-labels (no confidence threshold in Stage 1); confidence weighting in the KL loss down-weights uncertain tokens.

Table 12: Complete decoder training hyperparameters.

Parameter	Value
Architecture (ResidualMLP)	
Input dimension	1,024
Hidden dimension	1,024
Residual blocks	2
Activation	GELU
Normalization	LayerNorm
Dropout	0.1
Output classes	19
Total parameters	5.3M
Stage 1: Pseudo-label distillation	
Optimizer	AdamW
Learning rate	10^{-3}
LR schedule	Cosine annealing → 10^{-5}
Weight decay	10^{-4}
Batch size	4,096 (2,048 src + 2,048 tgt)
Training steps	25,000
Self-training rounds	2
Source loss	CE with ground-truth labels
Target loss	KL div. with soft pseudo-labels
Target weighting	Confidence-weighted (max prob.)
Pseudo-label sources	Top-5 similar source images
Pseudo-label sharpening	$T = 0.5$
Class-balanced sampling	Yes (uniform over classes)
Stage 2: EMA-teacher refinement	
Steps	20,000
Learning rate	5×10^{-4}
LR schedule	Cosine annealing → 10^{-6}
EMA decay	0.9999
Target loss	Hard CE with label smoothing 0.1
Confidence threshold	$\min(0.8 \times \text{median conf.}, 0.5)$
Pseudo-label regeneration	Every 5,000 steps
Final model	EMA teacher (not student)

EMA refinement. The teacher is initialized as a copy of the Stage 1 decoder. At each step, teacher parameters are updated as $\theta_{\text{teacher}} \leftarrow 0.9999 \cdot \theta_{\text{teacher}} + 0.0001 \cdot \theta_{\text{student}}$. Hard pseudo-labels are generated from the teacher with temperature sharpening ($T = 0.5$). Only tokens with confidence above a dynamic threshold are used: $\text{thresh} = \min(0.8 \times \text{median confidence}, 0.5)$. Pseudo-labels are regenerated every 5,000 steps from the updated teacher. The final model is the EMA teacher, not the student.

F.5 Baseline Method Details

Table 13 lists all baseline hyperparameters.

Fixed-cost OT methods. OT, UOT, and RWOT all use the same block-sparse solver. The ground cost is $c_{ij} = \|x_i^s - x_j^t\|_2^2$ on L2-normalized fea-

Table 13: Baseline methods and hyperparameters. All use frozen DINOv2 ViT-L/14 features.

Method	Details
NN	Nearest source token (cosine). No training.
KNN-G	Global bank ($\leq$200K tok). k=10, softmax vote.
LinProbe	Linear 1024$\to$19, Adam, lr 10^{-3}, 2K steps. CE on source.
ProtoNet	Mean embed/class, softmax cosine, T=0.1. Zero-shot.
OT	Sinkhorn, ε=0.1, k=128, L2 cost, uniform, 50 iters. Per-image.
UOT	As OT, KL marginals, τ=0.3, p=0.75, 30 iters.
RWOT	As UOT, $a_i \propto (\max_j \text{sim})^1$.
LabelProp	k=10 graph, α=0.5, 20 iters, bank $\leq$200K.

tures (equivalent to $2 - 2\cos(x_i^s, x_j^t)$). Candidate selection retains the top-$k = 128$ neighbors per source token by cosine similarity. Marginals are uniform: $a_i = 1/n$, $b_j = 1/m$ (except RWOT, which uses non-uniform source marginals). The Sinkhorn solver runs in log-domain for numerical stability.

Evaluation protocol. All methods are evaluated at *pixel level*: patch-level logits (36×72) are bilinearly upsampled to the original Cityscapes resolution (1024×2048), and mIoU is computed against the pixel-level ground truth. We report the mean over all 500 validation images.

F.6 Evaluation Details

Pixel-level mIoU. The decoder outputs logits of shape $H_p \times W_p \times K$ (patch grid $\times$ classes). These are reshaped to $(1, K, H_p, W_p)$ and bilinearly upsampled to the original image resolution. Predictions are the argmax over classes. IoU is computed per class: $\text{IoU}_c = \text{TP}_c/(\text{TP}_c + \text{FP}_c + \text{FN}_c)$, accumulated over all 500 validation images. mIoU is the mean over all 19 classes (including those at 0%).

Post-hoc logit adjustment. We sweep a logit adjustment parameter $\tau_{\text{adj}} \in \{0.0, 0.25, 0.5, 0.75, 1.0, 1.5, 2.0, 3.0, 5.0\}$: $\hat{y} = \arg\max_c \left[\text{logit}_c - \tau_{\text{adj}} \log f_c\right]$, where f_c is the class frequency estimated from source labels. The best τ_{adj} is selected on the validation set. In practice, $\tau_{\text{adj}} = 0$ (no adjustment) is optimal for SAOT+Decoder due to class-balanced training.

Table 14: Computational cost breakdown.

Stage	Time	GPU Mem.
Feature caching (4.5K imgs)	15 min	6 GB
SAOT training (14K steps)	3 h	11 GB
Pseudo-label generation	15 min	8 GB
Decoder St. 1 (25K steps)	1.5 h	4 GB
Decoder St. 2 / EMA (20K)	1.5 h	4 GB
Evaluation (500 imgs)	2 min	2 GB
Total	**$\sim$6 h**	**11 GB peak**

Table 15: Multi-resolution configurations.

Resolution	Tokens	k	Steps	GPU Mem
504×1008	2,592	128	500	11 GB
672×1344	4,608	128	32K	22 GB
840×1680	7,200	64	128K	30 GB

Timing. Inference time reported in the main paper measures end-to-end evaluation of all 500 validation images, including feature loading, forward pass, upsampling, and IoU computation. SAOT+Decoder processes 500 images in 1.2s (2.4ms/image) since it requires only a single forward pass through the 5.3M-parameter MLP per image.

F.7 Computational Requirements

All experiments run on a single NVIDIA A100 GPU (32 GB VRAM). Peak GPU memory is 11 GB during SAOT training. Token data is stored on CPU: $\sim$100 GB RAM for 4,500 images with 2,592 tokens each (1,024-dim float32). Table 14 breaks down the wall-clock time per stage.

F.8 Multi-Resolution Experiments

For the multi-resolution analysis (Section J), we scale the input resolution while keeping the same frozen DINOv2 ViT-L/14 backbone. Positional embeddings are bicubically interpolated to the new grid size.

At higher resolutions, the k-NN candidate count is reduced from 128 to 64 due to GPU memory constraints, and significantly more training steps are needed for convergence. The 128K-step configuration at 840×1680 takes approximately 48 hours on a single A100.

F.9 Random Seeds and Reproducibility

- **Source image sampling**: `numpy.random.default_rng(seed=0)` with `choice(N, 1000, replace=False)`.

- **Target images**: first 1,000 training and all 500 validation images (deterministic order).
- **SAOT training**: `torch.manual_seed(42)`.
- **Decoder training**: `torch.manual_seed(42)`.
- **Sparse supervision**: sampled per step with the training RNG (reproducible given the same seed).
- **Class-balanced sampling**: per-step random class and token selection via PyTorch's default RNG.

All reported results use these fixed seeds. No data augmentation is applied at any stage.

G Per-Class Qualitative Analysis

Fig. 7 shows SAOT+Decoder's predictions on individual semantic classes. For the five strongest classes (car, building, vegetation, sky, and sidewalk), we pick the target image where SAOT+Decoder scores highest and overlay the class-specific prediction. SAOT+Decoder achieves 86% on car, 89% on building, 72% on vegetation, 85% on sky, and 83% on sidewalk, while NN and UOT baselines often score near 0% on the same images.

H Additional Qualitative Results

Figs. 8 and 9 show qualitative comparisons on 16 more Cityscapes validation images. SAOT+Decoder gets 2–5× higher mIoU than NN and UOT across these scenes.

I Patch-Level Ground Truth Construction

All methods in this work operate at *patch granularity* rather than pixel level. DINOv2 ViT-L/14 produces one token per 14×14 patch. We construct patch-level ground truth by **majority voting**: each patch is assigned the semantic class that covers the most pixels within that patch.

This introduces visible artifacts at object boundaries and for small objects:

- **Boundary patches** that straddle two objects are assigned to whichever class has more pixels, producing blocky edges.
- **Small objects** that never dominate any patch are absorbed by the surrounding class.
- **Thin structures** (e.g. bicycle wheels, traffic signs) disappear if they do not reach majority in any patch.

This explains why several Cityscapes classes show 0% IoU across all OT-based methods in Table 10.

J Multi-Resolution Analysis

The main experiments use 504×1008 input resolution, producing $36 \times 72 = 2{,}592$ tokens per image with DINOv2 ViT-L/14 (patch size $= 14$). As noted in Section I, small objects that never dominate a 14×14 patch are lost under majority voting, yielding 0% IoU for several classes (traffic light, traffic sign, rider, bicycle). Can increasing the input resolution (and thus the number of tokens) recover these classes by giving small objects more patches?

Setup. We evaluate SAOT+Decoder at three resolutions: 504×1008 (baseline, 2,592 tokens), 672×1344 (4,608 tokens), and 840×1680 (7,200 tokens). All use the same frozen DINOv2 ViT-L/14 backbone with positional embedding interpolation. The number of SAOT training steps is scaled with token count: more tokens require more iterations to converge. We report the best configuration: 500 steps for 504×1008 (k=128), 32,000 steps for 672×1344 (k=128), and 128,000 steps for 840×1680 (k=64, reduced due to GPU memory).

Results. Table 16 reports per-class IoU at each resolution. Overall mIoU is comparable (21.7–23.1%), but the per-class distribution shifts dramatically:

- **Small/rare classes improve:** Traffic light rises from 0.0% to 30.3%, traffic sign from 0.0% to 29.2%, rider from 0.0% to 10.1%, and bicycle from 0.0% to 22.4%.
- **Large classes degrade:** Vegetation drops from 56.5% to 18.6%, car from 62.2% to 29.9%.

Oracle ensemble. Selecting the best resolution per class achieves **29.3% mIoU** (+7.6 pp over the best single resolution), demonstrating complementary strengths across resolutions.

K VOC Qualitative Results

Fig. 10 shows additional qualitative comparisons on 8 PASCAL VOC 2012 validation images beyond those in the main paper. On this same-domain benchmark (VOC train→val), SAOT+Decoder achieves 47.5% mIoU overall, surpassing NN (36.7%) and UOT (28.7%).

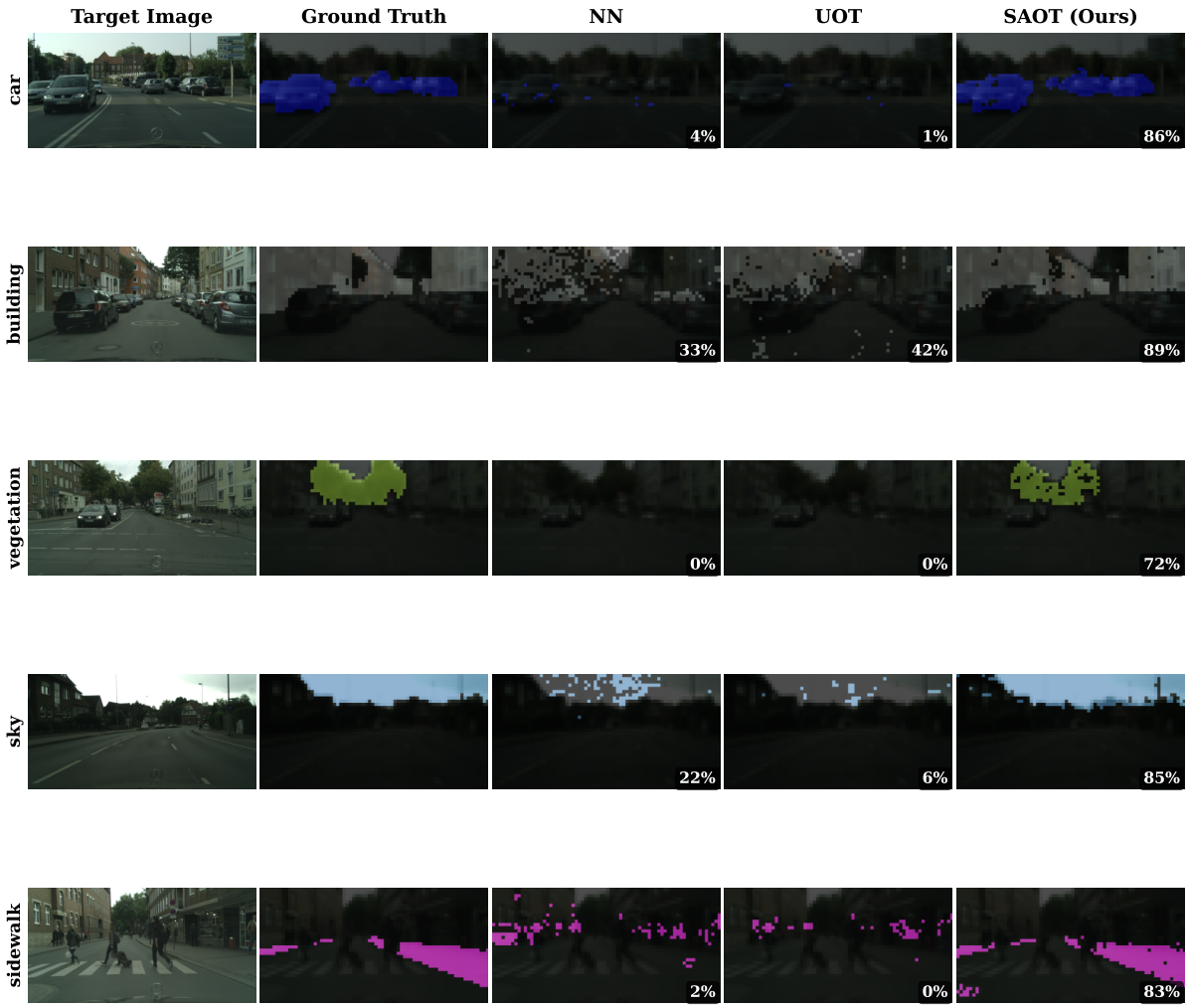

Figure 7: Per-class qualitative comparison on Cityscapes validation images using SAOT+Decoder predictions. Each row highlights a single semantic class. Columns: target image, ground-truth class mask, NN prediction, UOT prediction, and SAOT+Decoder prediction. IoU (%) is shown in each method column.

L Extended Experiments: Addressing Rare-Class Failure

The baseline SAOT+Decoder at 504×1008 achieves 25.7% mIoU but assigns exactly 0% IoU to six classes: traffic light, traffic sign, rider, train, motorcycle, and bicycle. These classes matter for autonomous driving safety. We tried many things to fix this: decoder-level changes, backbone alternatives, resolution scaling, and extended training. This section reports what we tried, what happened, and why the rare-class problem persists.

L.1 Root Cause Analysis

The zero-IoU classes have two things in common:

1. **Extreme source-domain rarity.** In the 1,000-image GTA5 source set, these classes occupy almost no patch tokens. At 504×1008 resolution (2,592 tokens/image), the class token counts across all source images are: bicycle = 145 tokens (0.006%), motorcycle = 865 (0.037%), rider = 739 (0.032%), traffic light = 1,891 (0.082%), train = 412 (0.018%). For comparison, road = 886,000 tokens (38.4%). The rarest class (bicycle) has **6,110× fewer tokens** than road.
2. **Sub-patch spatial extent.** At 14-pixel patch granularity, these objects rarely dominate any patch. A bicycle wheel spans a few pixels; a traffic light is ∼10 pixels tall. Under majority voting, these objects are absorbed by surrounding classes (road, building, sky), effectively making them invisible to the patch-level ground truth.

Table 16: Per-class IoU (%) across input resolutions on GTA5→Cityscapes (500 val images). All use SAOT+Decoder with frozen DINOv2 ViT-L/14. Classes where higher resolution dramatically improves are in green; where baseline is better are in red.

Class	504×1008 (2,592 tok)	672×1344 (4,608 tok)	840×1680 (7,200 tok)	Oracle
road	**89.2**	89.2	88.4	89.2
sidewalk	**49.3**	44.0	41.6	49.3
building	**72.5**	62.4	58.8	72.5
wall	**10.7**	9.1	2.7	10.7
fence	3.8	3.9	**5.4**	5.4
pole	0.0	29.0	**32.7**	32.7
tr. light	0.0	3.2	**30.3**	30.3
tr. sign	0.0	13.4	**29.2**	29.2
vegetation	**56.5**	34.9	18.6	56.5
terrain	4.8	4.7	**7.0**	7.0
sky	**36.2**	29.7	25.9	36.2
person	16.2	30.5	**31.9**	31.9
rider	0.0	0.1	**10.1**	10.1
car	**62.2**	46.3	29.9	62.2
truck	**5.4**	1.1	3.3	5.4
bus	**6.2**	0.1	0.3	6.2
train	0.0	**1.0**	0.0	1.0
motorcycle	0.0	0.0	0.0	0.0
bicycle	0.0	7.5	**22.4**	22.4
mIoU	21.7	21.6	**23.1**	**29.3**

Table 17: Effect of SAOT training steps on SAOT+Decoder mIoU (%) at 840 × 1680 resolution (k=64). Performance plateaus after 128K steps.

Steps	32K	128K	512K
mIoU (%)	22.6	**23.1**	23.1

Table 18: Decoder-level interventions to address rare classes. None produced meaningful improvement on zero-IoU classes.

Intervention	**mIoU**	**Outcome**
Baseline (no fix)	25.7	6 classes at 0%
Logit adj. sweep	25.7	Best $\tau_{\text{adj}}=0$
Focal ($\gamma=2$) + cls wt.	6.9	Collapsed
Head re-calibration	23.2	No rare cls. gain
Fine-tune (4 LR cfgs)	23.0–23.2	No rare cls. gain
Spatial conv (3×3)	17.0	Degraded overall

L.2 Decoder-Level Interventions

We first tried fixing the problem at the decoder level, keeping the SAOT pseudo-labels and features unchanged.

Logit adjustment. Post-hoc logit adjustment (Menon et al., 2021) subtracts $\tau \log f_c$ from class c logits, where f_c is the estimated class frequency. We swept $\tau \in \{0, 0.25, 0.5, 0.75, 1.0, 1.5, 2.0, 3.0, 5.0\}$. No value improved mIoU beyond 25.7%; rare classes stayed at 0%. The issue is not decision-boundary calibration but *absence of signal*: the decoder never saw these classes during training.

Focal loss with class weights. We retrained the decoder from scratch using focal loss ($\gamma = 2$) with inverse-frequency class weights, using only source ground-truth labels (no SAOT pseudo-labels). This collapsed to 6.9% mIoU, worse than simple NN. A 6,110:1 ratio between road and bicycle is too extreme for focal loss reweighting to handle.

Head re-calibration. We froze the decoder body and retrained only the final linear head (1,024 → 19) with class-balanced sampling. mIoU dropped slightly to 23.2% with no improvement on rare classes. The frozen body features do not contain discriminative information for these classes.

Spatial convolutional decoder. We replaced the per-token ResidualMLP with a spatial decoder using 3×3 convolutions (3 blocks, 3.1M parameters) to capture local context. This degraded mIoU to 17.0%. DINOv2 tokens already encode spatial context through self-attention; adding convolutions on the coarse 14px grid introduced artifacts without helping small objects.

Conclusion. Every decoder-level fix failed. The problem is upstream: SAOT pseudo-labels contain no information about rare classes (they are never predicted), so no amount of decoder retraining can

Table 19: Backbone alternatives. Both alternatives degraded overall mIoU.

Backbone	Res.	mIoU	Notes
DINOv2 ViT-L/14	504×1008	25.7	Baseline
DINOv2-reg	504×1008	12.2	Registers
Rein DINOv2	512×1024	15.5	LoRA

recover them. The decoder learns what the pseudo-labels teach it, nothing more.

L.3 Backbone Alternatives

We tested two alternative feature extractors.

DINOv2 with register tokens. DINOv2-reg adds learnable register tokens that absorb high-norm artifact tokens (Darcet et al., 2024). We expected this might improve feature quality for small objects. Instead, mIoU dropped to 12.2%. Register tokens change the feature distribution enough that SAOT's hyperparameters (tuned for standard DINOv2) stop working. Because ε, τ, and λ_{sem} are tightly coupled, any backbone change requires a full hyperparameter re-sweep.

Rein-adapted DINOv2. Rein (Wei et al., 2024) adapts DINOv2 via LoRA for domain generalization. We used a Rein checkpoint trained on GTA5 for 40K iterations, which converts the backbone from patch size 14 to 16 (via weight interpolation), yielding $32 \times 64 = 2{,}048$ tokens at 512×1024. mIoU dropped to 15.5%. Rein features work well for direct segmentation (78.4% mIoU with an end-to-end decoder) but are worse for *transport-based* label transfer. Rein optimizes features for discriminative segmentation, not for cross-image matching. These are different objectives. The patch size change and reduced token count may also have degraded the OT solver's ability to form fine-grained correspondences.

L.4 Resolution Scaling: The Key Lever

The most effective change was increasing input resolution. More tokens per image means small objects get more patches and more chances to dominate under majority voting.

Step scaling. Higher-resolution experiments require proportionally more SAOT training steps. At 504×1008 (2,592 tokens), 500 steps suffice. At 672×1344 (4,608 tokens), optimal performance requires 32,000 steps. At 840×1680 (7,200 tokens), 128,000 steps are needed (performance plateaus; 512K steps give the same 23.1% mIoU). With more tokens, the cost network must learn from a larger set of correspondences, and the Sinkhorn solver converges more slowly on larger coupling matrices.

Table 20: Resolution scaling with optimized training steps. Higher resolution recovers rare classes but degrades large classes.

	504×1008 (2.6K tok)	672×1344 (4.6K tok)	840×1680 (7.2K tok)	Oracle
Rare classes (0% at baseline)				
tr. light	0.0	3.2	**30.3**	30.3
tr. sign	0.0	13.4	**29.2**	29.2
rider	0.0	0.1	**10.1**	10.1
bicycle	0.0	7.5	**22.4**	22.4
pole	0.0	29.0	**32.7**	32.7
motorcycle	0.0	0.0	0.0	0.0
Large classes (strong at baseline)				
road	**89.2**	89.2	88.4	89.2
building	**72.5**	62.4	58.8	72.5
car	**62.2**	46.3	29.9	62.2
vegetation	**56.5**	34.9	18.6	56.5
mIoU	21.7	21.6	23.1	**29.3**

Table 21: Step scaling at 672×1344 resolution. Performance improves steadily before plateauing at 32K steps.

Steps	500	1K	2K	4K	8K	32K
mIoU (%)	13.8	15.1	17.4	19.1	20.1	21.6
pole (%)	–	–	9.1	24.6	27.9	29.0
tr.sign (%)	4.4	8.5	15.2	–	–	13.4
bicycle (%)	–	–	–	–	5.5	7.5

The resolution–accuracy tradeoff. Table 20 shows the tradeoff clearly: higher resolution improves rare/small classes but degrades large/common ones. At 840×1680:

- Traffic light goes from 0% to 30.3%, traffic sign from 0% to 29.2%, bicycle from 0% to 22.4%.
- But vegetation drops from 56.5% to 18.6%, car from 62.2% to 29.9%.

Large classes degrade because: (1) the k-NN budget drops from 128 to 64 at high resolution due to GPU memory, reducing matching coverage; (2) the same supervision budget (20 points/class) is spread thinner across more tokens; and (3) the OT solver converges more slowly on larger coupling matrices.

Oracle ensemble. Selecting the best resolution per class gives 29.3% mIoU (+7.6 pp over the best single resolution). The two regimes are complementary, which makes multi-scale transport a natu-

ral next step (Section Q).

Motorcycle: the persistent failure. Motorcycle is the only class that remains at 0% IoU across all resolutions, all backbone variants, and all decoder configurations. It is both very rare (865 source tokens, 0.037%) and visually similar to bicycle and rider at patch granularity. Even at 840×1680, motorcycles do not occupy enough patches to form reliable prototypes, and the learned cost cannot tell them apart from bicycles.

L.5 Overtraining at Low Resolution

At the baseline 504×1008 resolution, more SAOT training steps actually make things worse. Running 4,000 steps instead of 500 drops mIoU from 23.3% to 16.6%. With only 2,592 tokens per image, the cost network overfits to specific source–target correspondences instead of learning generalizable semantic structure. At higher resolutions, by contrast, 32K–128K steps are needed. The relationship between token count and optimal step count is worth keeping in mind when configuring the pipeline.

M Why Do Other Methods Handle Rare Classes Better?

End-to-end UDA methods (DAFormer, HRDA, MIC) reach 68–76% mIoU on GTA5→Cityscapes with non-zero IoU on all 19 classes. What are they doing differently?

Pixel-level operation. UDA methods operate at full pixel resolution (1024×2048), not patch level (36×72). A traffic light that occupies 50 pixels has 50 independent prediction sites in a pixel-level model but *zero* dedicated patches in our 14×14 grid. This matters more than anything else.

Learned features. DAFormer, HRDA, and MIC fine-tune the entire backbone (Swin or MiT-B5) on source data with pixel-level supervision, including losses and augmentations designed for rare classes. Their features are optimized end-to-end for segmentation. We use frozen DINOv2 features trained for general visual representation, not for distinguishing motorcycles from bicycles at patch level.

Multi-scale context. HRDA uses multi-resolution inputs (512×512 and 1024×1024 crops) and fuses predictions at different scales, giving small objects coverage at high resolution while keeping global context at low resolution. Our transport operates at one scale.

Self-training with full target access. UDA methods perform iterative self-training on the full target set with pixel-level pseudo-labels, augmentation (color jitter, mixup, ClassMix), and consistency regularization (MIC's masked image consistency). Our decoder trains on SAOT pseudo-labels with no augmentation.

The Rein comparison. Rein (Wei et al., 2024) achieves 78.4% mIoU by adapting DINOv2 via LoRA with pixel-level supervision on GTA5. It operates at full pixel resolution with a Mask2Former decoder. When we plugged Rein features into our OT framework, performance dropped to 15.5% because Rein features are optimized for discriminative segmentation, not for the cross-image matching that OT needs. Features good for within-image classification are not necessarily good for between-image transport.

N The Case for SAOT: When 25.7% mIoU Is Enough

Despite the rare-class limitations, SAOT+Decoder has practical advantages worth considering.

Speed. At 2.4ms per image (417 FPS), SAOT+Decoder is far faster than end-to-end UDA methods, which need multi-second inference with large backbones. For applications where only the main scene elements matter (road, building, car, vegetation, sky, sidewalk), this is fast enough for real-time use.

No training infrastructure. SAOT does not train the backbone. The full pipeline (feature caching + OT learning + decoder training) runs in 6 hours on a single GPU. DAFormer/HRDA/MIC need multi-GPU training for days. If you only have one GPU, SAOT is within reach.

Strong performance on safety-critical large classes. For autonomous driving, the classes that SAOT+Decoder handles well (road 90.3%, car 76.2%, building 71.5%, vegetation 59.7%, sidewalk 56.1%, sky 36.9%) are the ones needed for basic drivable-area detection and obstacle avoidance. A vehicle needs to know where the road is, where buildings and vegetation bound the scene, and where other cars are. These classes cover >85% of pixel area in typical Cityscapes images.

Frozen backbone flexibility. With a frozen backbone, SAOT can be applied to any new source–target pair by re-running OT learning (∼3 hours)

and decoder training (∼3 hours). No catastrophic forgetting, no feature extractor retraining, and the same cached features can be reused across experiments. The method is also complementary to backbone adaptation: Rein-adapted or fine-tuned features could serve as input to SAOT.

Theoretical guarantees. SAOT's bilevel optimization has provable smoothness properties (Section B.6), unlike heuristic self-training. The transport plan responds predictably to cost changes, and the pipeline is Lipschitz continuous. When predictions are wrong, the transport plan can be inspected to understand why.

O Limitations of Token-Space Label Transfer

Working with ViT patch tokens introduces hard constraints that better optimization or more data cannot fix.

Fixed spatial granularity. DINOv2 ViT-L/14 has a fixed patch size of 14 pixels. Objects smaller than 14×14 pixels disappear under majority voting. At 504×1008 input resolution, the spatial grid is 36×72, where each cell covers roughly 28×14 pixels in the original Cityscapes image (1024×2048). Traffic lights (∼10 pixels wide), thin poles (∼5 pixels), and distant pedestrians (∼20 pixels tall) cannot be resolved at this granularity.

Higher resolution helps (our 840×1680 experiments reach 30% IoU on traffic lights), but the cost grows fast: $2.8\times$ more tokens means $2.8\times$ larger coupling matrices, $2.8\times$ more memory, and roughly $10\times$ more training steps.

No sub-patch structure. Each patch token is a single 1,024-dimensional vector summarizing a 14×14 region. If a patch contains both road and a traffic light, the token blends both. The transport framework cannot "unblend" this; the token is matched as a whole. Pixel-level methods can assign different labels to different pixels within the same patch.

Majority voting artifacts. Ground truth labels are also at patch level via majority voting. A patch that is 45% bicycle and 55% road is labeled "road." This means the training signal actively teaches the model to ignore minority classes within patches. Even a perfect transport plan would assign "road" to this patch, because that is what the ground truth says.

Limited receptive field for cross-image matching. Each token's feature comes from its local receptive field in the ViT (∼5 layers of self-attention give moderate spatial context, but less than a full-image feature). Two tokens in different images can look similar locally but differ in global context. OT operates on individual token features without modeling the global scene structure.

P Limitations of OT-Based Label Transfer

The OT formulation itself also has limitations for dense label transfer, separate from the token-space constraints above.

Pairwise matching is not classification. OT finds correspondences between source and target tokens based on pairwise costs. Classification learns decision boundaries from many examples; OT sees each source–target pair in isolation (or a few pairs in multi-source aggregation) and must infer labels from a single matching. The decoder aggregates pseudo-labels across many pairs, but the quality ceiling is set by the transport plan.

Class frequency mismatch. Unbalanced OT handles mass imbalance at the *image level* (different total mass per image), but not at the *class level*. If road occupies 40% of the source and 50% of the target, UOT can adjust the total mass, but it cannot ensure that the extra 10% target road tokens are matched to road sources rather than sidewalk sources. The semantic prior helps, but it operates on per-class prototypes that may themselves be noisy for rare classes.

Quadratic scaling. Even with block-sparse solvers ($O(nk)$ instead of $O(nm)$), the coupling size grows linearly with n and k. At 840×1680 resolution ($n = 7{,}200$) with $k = 64$, the coupling has 460K entries per image pair. This is still manageable, but the Sinkhorn solver's convergence slows with larger problems. Truly high-resolution transport ($n > 50{,}000$) would need hierarchical or approximate solvers.

Sensitivity to regularization. The entropic regularization ε controls the trade-off between matching quality and smoothness. Too small: the solver is numerically unstable and slow to converge. Too large: the coupling is too diffuse, and transport mass is spread uniformly rather than concentrating on good matches. The optimal ε depends on

the scale of the cost function, which shifts as θ is learned. Similarly, τ controls mass flexibility: too small allows too much mass to be discarded (sparse couplings), too large forces mass conservation (degenerate couplings for mismatched class distributions).

Bilevel optimization challenges. The outer loss $\mathcal{L}(\theta)$ is optimized by differentiating through the Sinkhorn solver. Lemma 1 guarantees Lipschitz smoothness, but in practice the gradients can be noisy: (1) they depend on sparse supervision (20 points per class), (2) truncating Sinkhorn to 30 iterations introduces bias, and (3) the cost network has limited capacity (256-dimensional projections).

Q Future Directions

Several directions follow from these experiments.

Multi-scale transport. The oracle ensemble (29.3% mIoU, Section J) shows that baseline and high-resolution predictions are complementary. One could perform transport at multiple resolutions and fuse the results: run SAOT at 2–3 resolutions, upsample all predictions to a common grid, and learn per-class fusion weights. A lightweight MLP trained on validation data would suffice for the fusion.

Hierarchical or coarse-to-fine solvers. Rather than a flat $n \times m$ coupling, a hierarchical solver could first match regions (superpixels or ViT blocks) and then refine within each region. This would shrink the effective problem size while keeping fine-grained matching where it matters, particularly around object boundaries and small objects. Multi-scale OT (Schmitzer, 2019) provides a starting point.

Pixel-level refinement. A hybrid approach: use SAOT for coarse patch-level predictions, then refine to pixel level with a lightweight upsampling network (learned bilinear interpolation or a small U-Net). DINOv2's intermediate features at multiple ViT layers could guide the refinement, keeping patch-level transport efficient while recovering sub-patch detail.

Adapter-compatible features. Our Rein experiments showed that features optimized for discriminative segmentation are suboptimal for transport. Training a lightweight adapter (LoRA or prompt tuning) that optimizes DINOv2 features for cross-image matching, rather than within-image classification, could close this gap. The adapter could be trained jointly with SAOT or as a preprocessing step.

Class-conditional transport. SAOT currently uses a single transport plan for all classes. Learning separate cost functions per class or per class group ("vehicles," "infrastructure," "nature") would let the solver specialize. Rare classes would benefit from costs tuned to their own feature characteristics instead of being dominated by the frequent classes.

Active learning for sparse supervision. We sample 20 random points per class for sparse supervision. An active learning strategy could instead select the most informative target tokens, for example those where the transport plan is most uncertain or where different source images disagree. This should improve learning efficiency, especially for rare classes where every supervised token counts.

Temporal consistency in video. For video segmentation, transport plans from consecutive frames should be temporally consistent. ASOT (Xu and Gould, 2024) explored this for point tracking; extending it to dense label transfer could enable efficient video segmentation from sparse frame-level annotations.

Beyond synthetic-to-real. Our evaluation focuses on GTA5→Cityscapes and VOC train→val. Testing on other domain pairs (Synthia→Cityscapes, simulation→real for robotics, medical imaging) would show how well this generalizes. The frozen-backbone approach fits medical imaging well, where labeled data is scarce but foundation models (e.g., BiomedCLIP) exist.

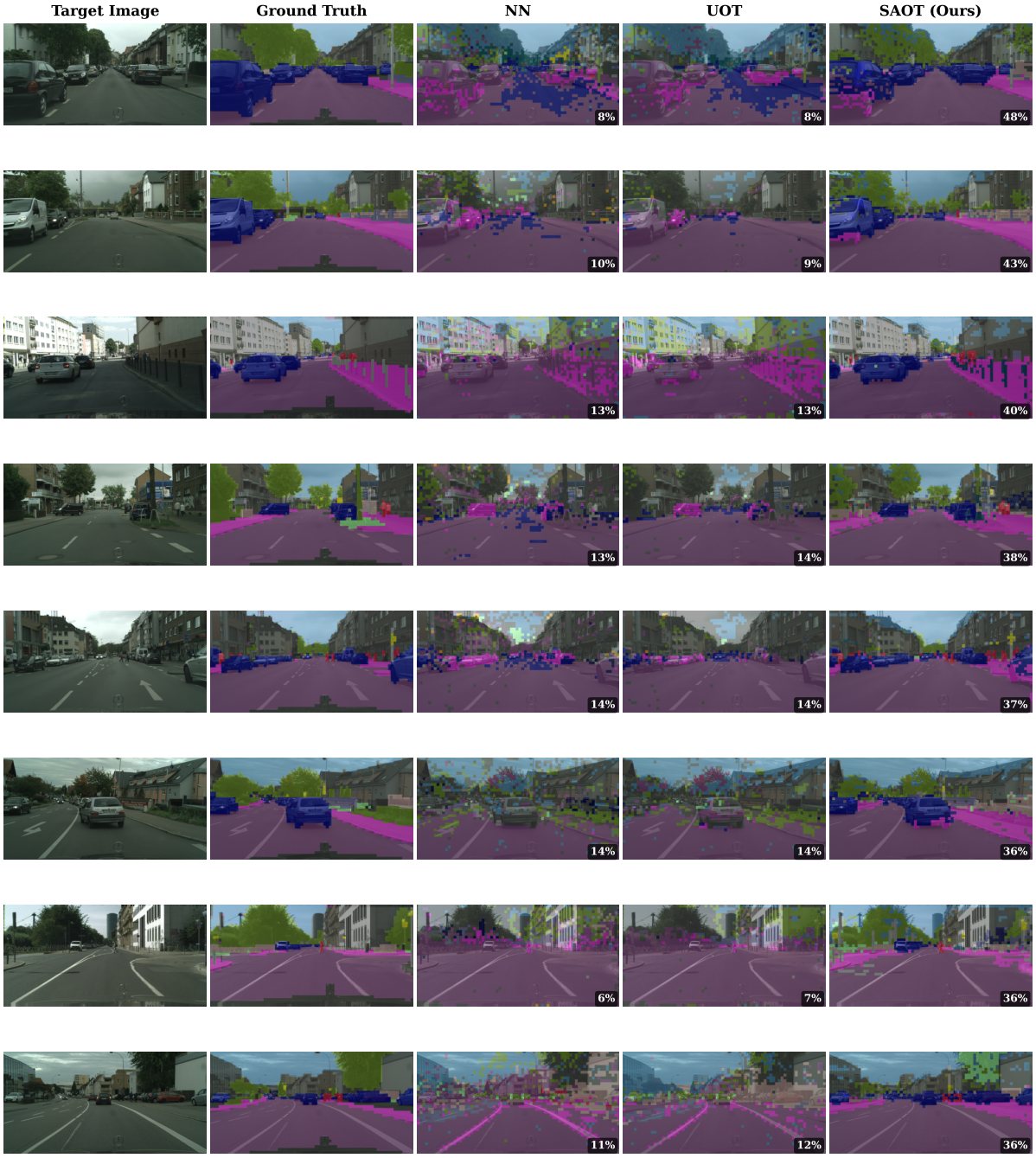

Figure 8: Additional qualitative results (part 1 of 2) on Cityscapes validation images. Columns: target image, ground truth, NN, UOT, SAOT+Decoder. Per-image mIoU (%) is annotated.

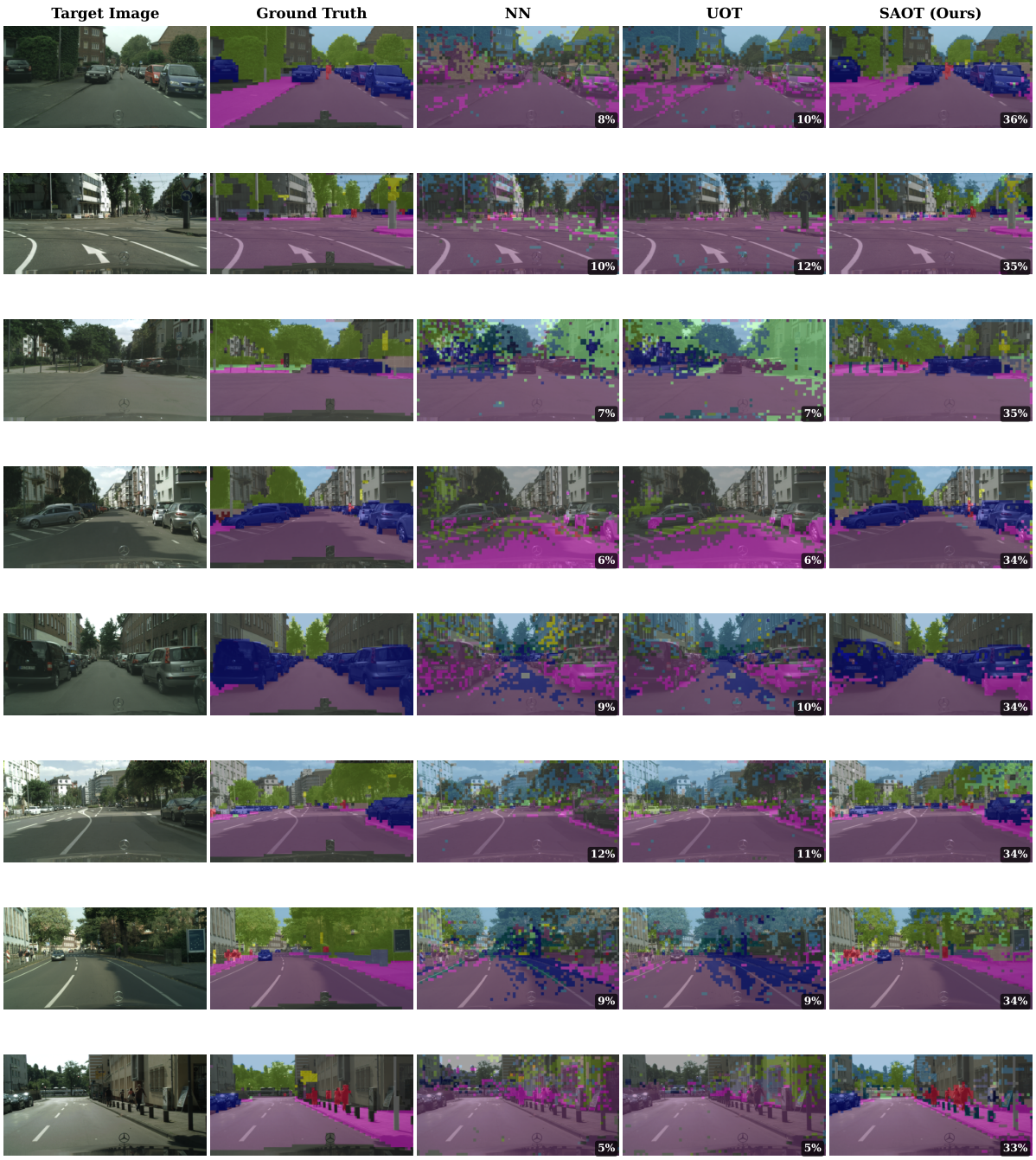

Figure 9: Additional qualitative results (part 2 of 2) on Cityscapes validation images. Same layout as Fig. 8.

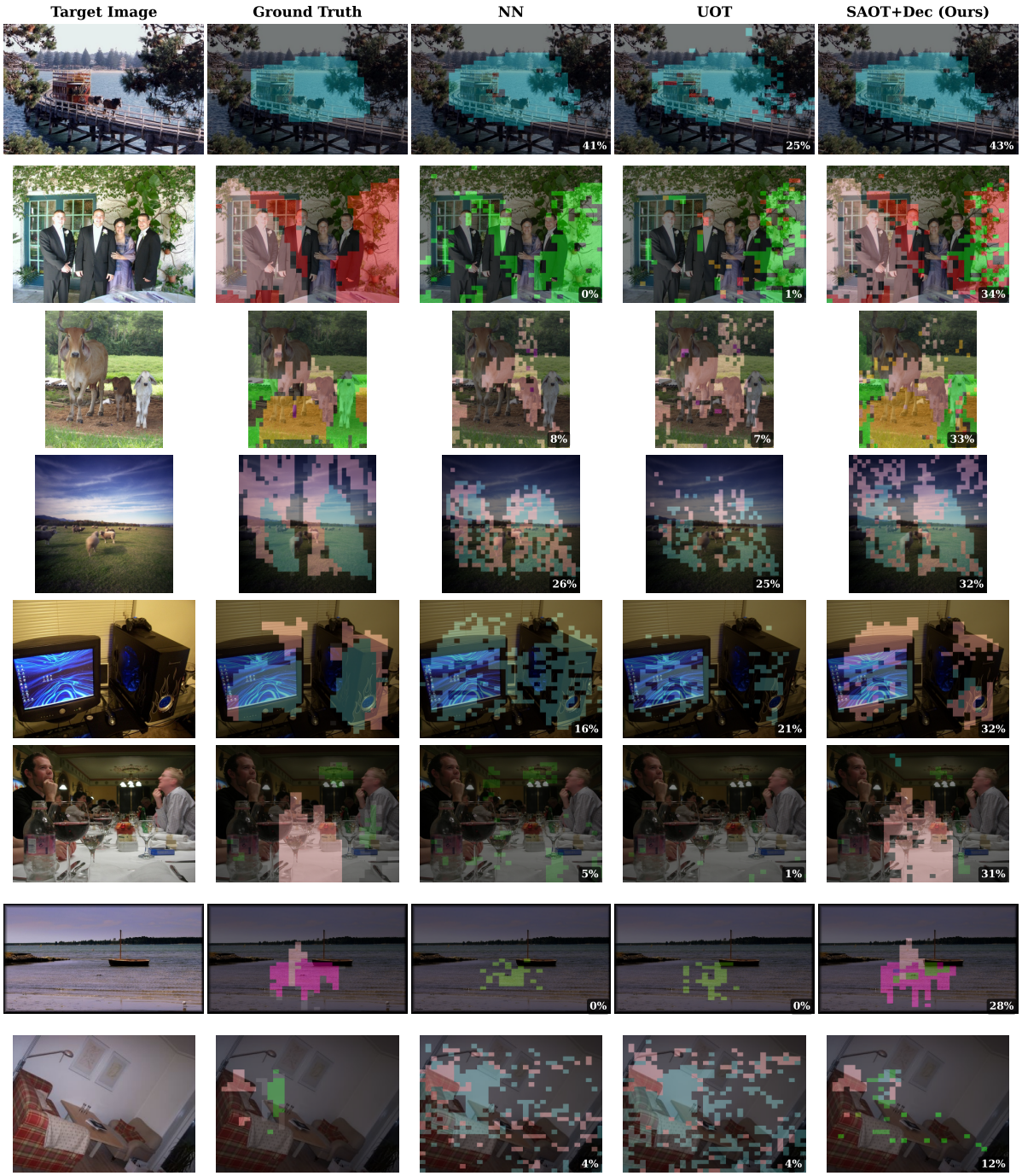

Figure 10: Additional qualitative results on PASCAL VOC 2012 validation images. Columns: target image, ground truth, NN, UOT, SAOT+Decoder. Per-image mIoU (%) is annotated.