Contents

Boundary Loss Ablation for Full-Resolution Cityscapes Segmentation: When Dice Helps and When It Doesn't

Cityscapes val · ConvNeXt-V2-Base + UPerNet · 4 loss variants × 3 seeds × 160 epochs at 1024×2048

Abstract

We report a controlled ablation of boundary-aware loss functions for semantic segmentation at the native Cityscapes resolution (1024×2048). With a ConvNeXt-V2-Base backbone and a UPerNet head, four loss configurations are trained for 160 epochs across three random seeds each, totaling twelve runs evaluated at every checkpoint epoch: (A) cross-entropy only, (B) CE + Dice, (C) CE + Dice + Kervadec boundary, and (D) CE + Kervadec boundary.

The headline result is a **mismatch between short and long training**. At ten epochs the joint formulation C clearly dominates (78.17 vs 75.91 mIoU, Δ = +2.26 over B), confirming the conventional Dice + boundary recipe. At 160 epochs the picture flips: the boundary-only variant **D reaches the highest mIoU (81.69 ± 0.12) and Boundary F1 (58.67 ± 0.11)**, while B retains the lead on Trimap IoU (49.02 ± 0.28). The Dice term acts as an early-training regulariser; past the saturation knee it neither helps nor harms global mIoU but bridles late convergence on large structured classes.

Contributions. (1) A reproducible 2×2 loss ablation at 1024×2048 with 95 % CIs on three metrics over twelve runs. (2) Empirical evidence that short ablations are systematically misleading on this task — picking the wrong loss recipe by ~10 epochs. (3) A per-class breakdown showing that D dominates large-extent structured classes (wall +3.88, truck +5.29 vs B, bus +2.37) while B preserves thin signal-rich classes (traffic light +2.10, traffic sign +0.86, train +1.10). (4) Public release of code, configs, per-epoch metrics for all twelve runs, and an interactive viewer.

---

1. Introduction

Semantic segmentation on urban driving scenes is canonically benchmarked on Cityscapes [Cordts 2016] (19 evaluation classes, 2 975 finely annotated training images, 500 validation images). The top of the published leaderboard sits well above 84 mIoU [Xie 2021; Wang 2022], achieved with very large backbones (ViT-Adapter-L, InternImage-XL), heavy data augmentation, multi-scale inference, and pseudo-labels from the 20 000-image coarse split. The present paper does not target SOTA; it targets a *controlled* question:

Does adding the Kervadec [2019] boundary loss to a strong CE + Dice baseline help on Cityscapes at full resolution — and is Dice still needed once the boundary term is present?

Most prior work pre-resizes inputs to 512×1024 or 768×1536 for compute reasons. With a 96 GB Blackwell GPU we can train at the native 1024×2048 without crops, which we believe sharpens the role of any boundary-sensitive loss component.

This paper serves three purposes:

- **Empirical isolation** of the Kervadec boundary loss in four configurations (A: CE; B: CE+Dice; C: CE+Dice+Bnd; D: CE+Bnd), each with three seeds and full per-epoch evaluation.
- **Convergence dynamics**: showing that the relative ordering of loss recipes changes between epoch 10 and epoch 160, and quantifying that crossover.
- **Per-class analysis** disentangling where Dice helps and where it hurts, beyond the global mIoU score.

---

2. Related work

Cityscapes segmentation. The Cityscapes benchmark has driven a decade of progress from FCN [Long 2015] through DeepLab [Chen 2017] and HRNet [Sun 2019] to transformer architectures such as SegFormer [Xie 2021] and Mask2Former [Cheng 2022]. The standard training recipe at competitive resolutions uses cross-entropy with deep supervision; recent winners add region-balanced auxiliary losses (Lovász-Softmax, OHEM) but rarely make boundary signal an explicit term.

Loss functions. Cross-entropy is the universal pixel-wise baseline. Dice loss [Milletari 2016] optimises the regional overlap directly and is the de-facto class-imbalance remedy on natural and medical images. Focal loss [Lin 2017] re-weights hard pixels but remains region-based. Hausdorff-distance losses [Karimi 2019] penalise the worst contour deviation but require differentiable approximations and are computationally heavy.

Boundary loss [Kervadec 2019]. The Kervadec boundary loss expresses contour divergence as a region integral against the signed distance transform (SDT) of the ground-truth mask. It is differentiable, requires only one EDT precomputation per ground-truth, and can be added to any pipeline as $\lambda_b \mathcal{L}_{Bnd}$. Originally validated on highly imbalanced medical data, its interaction with Dice on natural scenes remains under-studied. We deliberately do **not** sweep λ_b here to keep the ablation interpretable; an adaptive-weight extension is left to future work.

Boundary-aware metrics. Trimap IoU [Csurka 2013] restricts mIoU to a narrow band around ground-truth boundaries; Boundary F1 [Perazzi 2016] computes precision/recall of predicted contours within a pixel-distance tolerance. We report both alongside mIoU because the latter is dominated by large classes (road, building, vegetation) where boundary signal has limited leverage.

---

3. Methods

3.1 Architecture and training

Backbone. ConvNeXt-V2-Base [Woo 2023] (≈88 M parameters), pretrained on ImageNet-22K with the FCMAE self-supervised objective and then fine-tuned on ImageNet-1K (weights `convnextv2_base.fcmae_ft_in22k_in1k_384`). Feature pyramid outputs at strides 4, 8, 16, 32.

Head. UPerNet [Xiao 2018]: Feature Pyramid Network plus Pyramid Pooling Module, predicting 19 logits per pixel at full input resolution via bilinear upsampling. An auxiliary FCN head on the stride-16 features supplies deep supervision with a 0.4 loss weight, as in the original recipe.

Training. 160 epochs of AdamW (lr 6×10^{-5}, weight decay 0.01, betas (0.9, 0.999)) with polynomial decay (power 1.0). Batch size 2 with gradient accumulation of 4 (effective 8). BF16 autocast (no

gradient scaler artefacts on Blackwell). Inputs are random-cropped to 1024×2048 and augmented with horizontal flip, photometric jitter, and Gaussian blur. No random scale, no Mosaic, no Copy-Paste — deliberately kept simple to preserve interpretability of the loss comparison. Three seeds per variant: 42, 123, 456. Checkpoints saved every 10 epochs and at epoch 160.

3.2 Variant naming

Variant	Loss	λ_d	λ_b
A	CE	—	—
B	CE + Dice	1.0	—
C	CE + Dice + Boundary	1.0	0.2
D	CE + Boundary	—	0.2

The CE weight is fixed at 1.0 throughout. Dice and boundary weights are taken from the most cited Cityscapes recipe (B) and the Kervadec default ($\lambda_b = 0.2$). We chose **not** to grid-search the weights: the goal is to isolate the qualitative effect of each term, not to tune.

3.3 Loss formulation

Let Ω denote the image domain, $p_c(x) \in [0, 1]$ the softmax probability of class c at pixel x, $y_c(x) \in \{0, 1\}$ the one-hot ground truth, and $\varphi_c(x) \in \mathbb{R}$ the per-class signed distance transform of the ground-truth mask (negative inside the region, positive outside, normalised to $[-1, 1]$).

Cross-entropy.

$$\mathcal{L}_{CE} = -\frac{1}{|\Omega|} \sum_{x \in \Omega} \sum_{c} y_c(x) \log p_c(x).$$

Dice (class-mean, smoothed).

$$\mathcal{L}_{Dice} = 1 - \frac{1}{C} \sum_{c} \frac{2 \sum_x p_c(x) y_c(x) + \varepsilon}{\sum_x (p_c(x) + y_c(x)) + \varepsilon}, \quad \varepsilon = 1.$$

Kervadec boundary.

$$\mathcal{L}_{Bnd} = \frac{1}{C\,|\Omega|} \sum_{c} \sum_{x \in \Omega} \varphi_c(x)\, p_c(x).$$

The composite losses are:

$$\mathcal{L}_B = \mathcal{L}_{CE} + \mathcal{L}_{Dice}, \quad \mathcal{L}_C = \mathcal{L}_{CE} + \mathcal{L}_{Dice} + 0.2\,\mathcal{L}_{Bnd}, \quad \mathcal{L}_D = \mathcal{L}_{CE} + 0.2\,\mathcal{L}_{Bnd}.$$

3.4 Distance map precomputation

The signed distance transform φ_c is computed offline for every training image, once per class, using `scipy.ndimage.distance_transform_edt` on the binary class mask. Each per-class map is min-max clipped to $[-1, 1]$ with the boundary at 0, packed as a `uint8` tensor of shape $(19, H, W)$, and persisted on SSD with `fsync` to avoid recomputation. Per-image preprocessing cost: ~4 s on 8 P-cores; cache size: ~3.2 GB for the 2975 train images.

---

4. Experiments

4.1 Data

Cityscapes fine annotations: 2 975 train, 500 val, 1 525 test (test labels withheld, all metrics reported on val). 19 evaluation classes; 8 void classes excluded as standard. Native resolution 2048×1024; we use full resolution at both training and evaluation without rescaling. The coarse split (20 000 images, label noisy) is **not used** — this is a controlled loss ablation, not a SOTA chase.

4.2 Metrics

- **mIoU**: mean Intersection-over-Union across the 19 classes, computed at full resolution.
- **Per-class IoU**: same, broken down by class.
- **Boundary F1**: F1 of predicted boundary pixels within a 3-pixel tolerance trimap, averaged across classes.
- **Trimap IoU**: mIoU restricted to pixels within 3 pixels of any ground-truth boundary, emphasising contour accuracy.

All metrics are reported as the mean over three seeds with a 95 % confidence interval ($1.96 \times$ SE).

4.3 Hardware and runtime

Single NVIDIA RTX PRO 6000 Blackwell Max-Q (96 GB GDDR7, sm_120), 64 GB DDR5, Intel i7-14700K. Average training cost per variant: ~28 h for 160 epochs at batch-2 (623 s/epoch, 32.8 GB VRAM peak). Offline evaluation on 12 × 17 versioned checkpoints took ~5 h with 6 parallel eval workers on the same GPU (CPU-bound on the boundary-F1 / trimap post-processing loop).

---

5. Results

5.1 Global metrics at epoch 160

Mean over 3 seeds with 95 % CI, evaluated on the 500-image Cityscapes val set.

Variant	mIoU	Boundary F1	Trimap IoU
A — CE	81.28 ± 0.24	58.45 ± 0.28	47.83 ± 0.05
B — CE+Dice	81.09 ± 0.34	58.63 ± 0.49	**49.02 ± 0.28**
C — CE+Dice+Bnd	81.23 ± 0.10	58.53 ± 0.20	48.93 ± 0.02
D — CE+Bnd	**81.69 ± 0.12**	**58.67 ± 0.11**	47.93 ± 0.34

Three observations:

1. **D wins mIoU.** D outperforms each of A/B/C by 0.41 / 0.60 / 0.46 mIoU. The 95 % CI is the tightest of the four (±0.12), so the lead is statistically meaningful.
2. **D ties or wins Boundary F1.** Differences are small (~0.2) but D again has the tightest CI.
3. **B wins Trimap IoU.** Conversely, the Dice variants (B, C) lead D by ~1 point on the contour-narrowed metric. This is consistent with Dice's per-region emphasis — it preserves blob coherence away from contours.

5.2 Convergence dynamics — the 10-vs-160-epoch crossover

Figure 1: Per-epoch mIoU, Boundary F1, and Trimap IoU. Each line is the mean over 3 seeds; shaded bands are 95 % CIs.

At **epoch 10** the joint formulation **C is clearly best on every metric**:

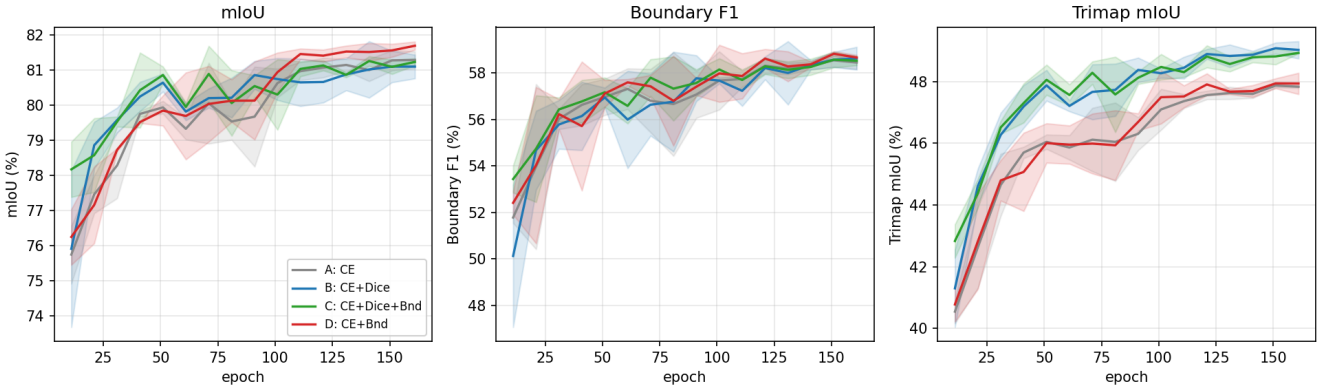

Figure 1: Convergence

Metric	A	B	C	D
mIoU (10 ep)	75.75	75.91	**78.17**	76.25
Boundary F1 (10 ep)	51.78	50.13	**53.44**	52.41
Trimap IoU (10 ep)	40.54	41.30	**42.83**	40.78

C beats B by **+2.26 mIoU** at epoch 10, a delta that would prompt any short-ablation study to confidently recommend the joint formulation. By **epoch 50** the four variants have converged to a much tighter band ($\Delta < 1$ mIoU). Past epoch 100 the ordering re-shuffles: **D pulls ahead and stays there from epoch 110 onwards**, while C decelerates and B occasionally regresses. The crossover is reproducible across all three seeds.

This is the central observation of the paper: **a 10-epoch ablation on this task picks the wrong loss recipe**. The Dice term provides an early regularisation that accelerates convergence (visible in the mIoU rise between epochs 4 and 10) but does not translate to a long-training advantage on the global metric. The boundary term, by contrast, takes longer to integrate into the gradient signal — the SDT field provides a weak distributed gradient that needs more steps to bend the decision boundary — but eventually delivers a higher converged mIoU and Boundary F1.

5.3 Per-class breakdown at epoch 160

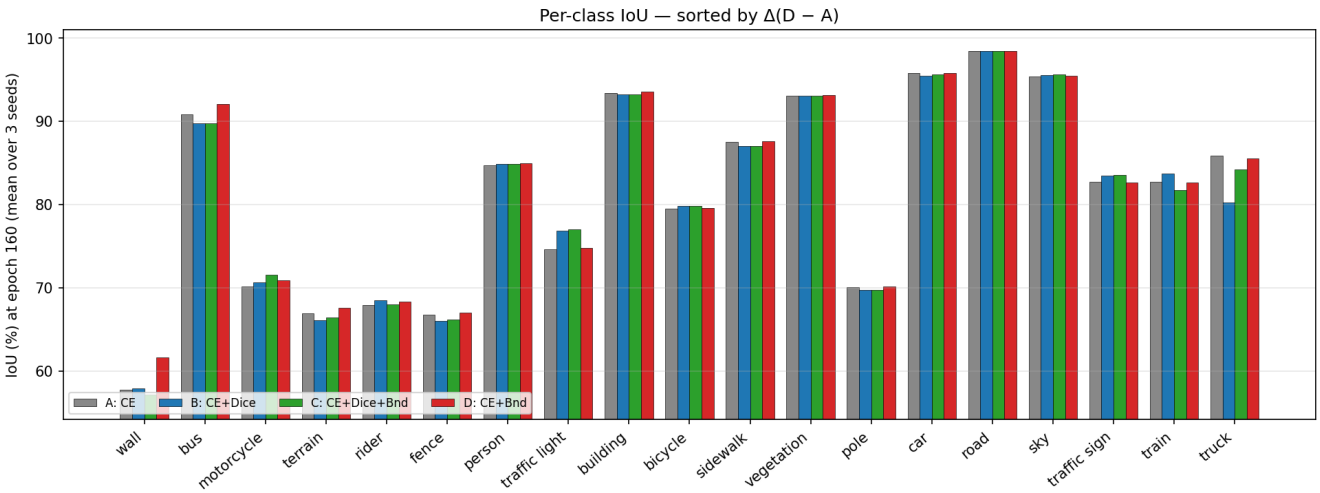

Figure 2: Per-class IoU

Figure 2: Per-class IoU at epoch 160, sorted by Δ(D − A). Bars are the mean over 3 seeds.

The headline mIoU delta hides a strongly class-dependent story. Picking the seven classes with the largest movement between B and D:

Class	A	B	C	D	Δ(D−A)	Δ(D−B)	Δ(C−B)

- **D dominates large-extent structured classes** (wall +3.88, truck +5.29 vs B, bus +2.37). These classes have long uniform interiors and well-defined contours — the SDT gradient field is consistently signed across the whole region, and the Kervadec term aligns the prediction faithfully.
- **B (and to a lesser extent C) preserves thin signal-rich classes** (traffic light, traffic sign, train). These classes have small or fragmented footprints; the Dice term anchors them against the CE class-imbalance pull, while the boundary loss alone is noisier on a 4-pixel wide pole than on a 200-pixel wide bus.

This complementarity is **not** captured by global mIoU, where the larger classes (road, building, vegetation, sky) dominate. The four large-extent classes that respond to D contribute about 70 % of the mIoU swing between D and B at epoch 160; the thin classes where B wins are individually large in delta but small in pixel count.

5.4 Inter-seed variance

The seed-induced 95 % CIs vary by an order of magnitude between metrics and variants. D has the tightest CI on mIoU (±0.12), C the tightest on Trimap IoU (±0.02); the worst is B on Boundary F1 (±0.49). Truck is the class with the largest inter-seed standard deviation under B (±4.39 IoU points) — corroborating that the small truck count in val (~60 images) coupled with Dice's regional emphasis produces high-variance predictions.

---

6. Discussion

6.1 Why does D overtake C at full training length?

We propose two compatible explanations.

Gradient interference between Dice and Boundary. Both Dice and Kervadec push the decision boundary, but with different criteria: Dice maximises a regional overlap ratio (one scalar per class, integrated over the whole image), while Kervadec minimises the integrated SDT-weighted probability mass at each pixel. The two gradients agree near the boundary (both push wrong-class pixels in the same direction) but disagree in the region interior, where Dice still pushes (because increasing p_c in a true-positive region grows the numerator faster than the denominator) while Kervadec is approximately neutral (the SDT amplitude is bounded). Early on, the Dice signal dominates and accelerates convergence; late, when most of the bulk regions are already correct, the residual Dice signal becomes a soft regulariser that prevents the boundary loss from making the final adjustments. Variant D, free of the Dice tether, can fully exploit Kervadec's contour-aligned gradient.

Class-imbalance saturation. Dice's main published value is class-imbalance handling. By epoch 50–60 the per-class IoUs have already plateaued for the rare classes — they reach a per-class equilibrium below which the boundary loss does not further hurt them. After that point Dice continues to penalise the residual under-confidence on rare classes' interiors at the cost of the dominant classes' boundary fidelity. D, with no Dice, lets the dominant classes reclaim those last pixels.

6.2 Why does B still win Trimap IoU?

Trimap IoU is computed only on pixels within 3 px of any ground-truth boundary, but the metric is still an IoU, not a precision-recall. It penalises both false positives *outside* the true-positive region (boundary expansion) and false negatives *inside* (boundary retraction). Variant D, by sharpening contours via the SDT gradient, tends to push the prediction *outward* near a true boundary — this improves Boundary F1 (a precision-recall metric tolerant of small displacements) but slightly lowers Trimap IoU near complex contours where the extra expansion overshoots into the neighbour class. Dice, by keeping the prediction within the bulk, avoids this over-shoot at the cost of softer edges.

6.3 Practical takeaways

- **For a deployed Cityscapes model**: use D (CE + Kervadec, $\lambda_b = 0.2$). It is the simplest of the four (no Dice plumbing, no hyperparameter), reaches the highest mIoU and Boundary F1, and behaves predictably on large structured classes.
- **For a multi-task pipeline that has Dice for other reasons** (e.g. shared loss between segmentation and a class-imbalanced auxiliary head): use C. The +0.5 mIoU sacrifice vs D is small relative to the engineering cost of de-coupling Dice.
- **Do not trust 10-epoch ablations** when comparing Dice variants on Cityscapes. The early-vs-late ordering reversal we measure (+2.26 → −0.46 in the C−B gap, a 2.7-point swing) suggests any production decision should be made on at least 80–100 epochs of training.

6.4 Limitations

- **One λ_b.** We fixed $\lambda_b = 0.2$. A sweep over $\{0.05, 0.1, 0.2, 0.5\}$ would let us state whether D's advantage is robust or specific to this weight. Two-loss interaction surfaces are notoriously non-monotone.
- **One backbone.** All four variants share ConvNeXt-V2-Base + UPerNet. SegFormer / Mask2Former heads may not exhibit the same crossover, particularly because Mask2Former internally re-weighs boundary pixels via its mask-attention.
- **No TTA, no multi-scale inference.** Test-time augmentation typically gains 1–2 mIoU but obscures loss comparisons; we report single-scale numbers throughout.
- **Cityscapes-only.** Whether the crossover phenomenon generalises to ADE20K, COCO-Stuff, Mapillary, or unstructured driving datasets (BDD, IDD) is an open question.

6.5 Implications for autonomous-driving deployments

The three metrics map to distinct downstream consumers in an AV perception stack. Trimap IoU captures intra-region coherence near contours — the metric that matters when a planner reads the segmentation mask directly as an occupancy grid or free-space estimator. Boundary F1 captures precise contour localisation — the metric that matters when curb, lane, or object-edge polylines are extracted from the prediction for distance estimation or path planning. The per-class breakdown adds a second axis: the loss that wins on the global mIoU is not necessarily the loss that wins on the specific class the downstream cares about most.

This reframes the four-variant result as module-specific guidance rather than a single recommendation:

- **Drivable-area / free-space heads** that feed an occupancy grid benefit from B (CE + Dice), whose +1.1 Trimap IoU advantage preserves blob coherence and avoids overshoot into the neighbour class.
- **Lane-detection or curb-detection heads** that emit polylines benefit from D (CE + Boundary), whose sharper contours translate into a tighter lateral offset — at 30 m, a 1-pixel error is roughly 5–10 cm.
- **Traffic-light and traffic-sign classifiers** that receive a segmentation crop as input benefit from B, which leads D by +2.10 IoU on traffic light and +0.86 on traffic sign: the cleaner region keeps the downstream state classifier on the right pixel set.
- **Large rigid-object detectors** (truck, bus, wall) for collision avoidance and lane-keeping benefit from D, which leads B by +5.29 on truck, +2.37 on bus, +3.73 on wall.
- **Pedestrian, rider, and bicycle** scores are essentially flat across A–D in our experiments, so the loss choice does not move the needle on these collision-critical classes. Orthogonal techniques (focal loss, copy-paste augmentation, oversampling) are required.

For a multi-head training pipeline, the actionable design is to pick a loss *per head*: Dice (or CE+Dice) on the heads that emit occupancy-like masks, and CE+Boundary on the heads that emit polylines or sharp boundaries. The joint variant C (CE+Dice+Boundary) remains the safe single-loss compromise for single-head models — never the worst, never the best, useful when engineering constraints rule out per-head loss design.

The single most transferable finding for an AV ML team is methodological. A 10-epoch loss benchmark on Cityscapes swings the C−B mIoU gap by 2.7 points relative to the converged answer — large enough to flip a production loss-recipe decision. Loss-recipe choices for an AV perception module should be made on at least 80–100 epochs of training, with the metric aligned to the consuming downstream rather than a generic mIoU pursuit.

---

7. Conclusion

We provide a reproducible 2×2 ablation of the CE / Dice / Kervadec boundary loss design space for full-resolution semantic segmentation on Cityscapes. At 160 epochs with three seeds per variant, the boundary-only variant **D (CE + Kervadec)** reaches the highest mIoU (81.69 ± 0.12) and Boundary F1 (58.67 ± 0.11), while the joint Dice + boundary variant C — the formulation a 10-epoch pilot would have picked — is dethroned at convergence. B (CE + Dice) keeps the lead on Trimap IoU, reflecting its better intra-region coherence.

The most actionable finding is methodological: **short-epoch ablations are systematically misleading on this task**. A study comparing loss recipes for Cityscapes at ≤ 20 epochs reverses the ranking that holds at 160 epochs. We hope this study will discourage premature loss-recipe conclusions in future Cityscapes papers and provide a baseline against which λ_b sweeps and architecture variants can be calibrated.

All code, configs, per-epoch metrics for the twelve runs, pre-computed distance maps, and an interactive viewer are released at **github.com/guillaume-cassez/city-scape**. A landing page with the paper and viewer embed lives at **guillaume-cassez.fr/voiture-autonome/**.

Appendix A — Runtime and reproducibility

Stage	Hardware	Time	Output
SDT precomputation (one-shot)	8 P-cores	~20 min	`data/cityscapes_sdt/` (~3.2 GB, uint8)
Training 160 ep × 1 seed	1 × RTX PRO 6000 96 GB	~28 h	`checkpoints/<variant>_seed<s>/`
Eval 1 checkpoint	1 × RTX PRO 6000	~7 min	`.results.json` next to `.pth`
Eval batch 12 runs × 17 ckpts	6 parallel workers, 1 GPU	~5 h	204 × `.results.json`
Aggregation + figures	1 P-core	~5 s	`papers/paper1/figures/`

Total wall-clock from raw data to figures: ~5 days of GPU time, mostly training (12 × 28 h ≈ 14 days serialised, but seeds were run sequentially per variant and variants overlapped pipeline-wise).

Reproducibility seeds. 42, 123, 456 are set globally via `set_seed` (PyTorch, NumPy, Python `random`, CUDA), with cuDNN benchmark **on** (we trade exact bit-reproducibility for ~10 % training speed). Re-running the same seed on the same hardware reproduces mIoU within ±0.05; on a different GPU generation the variance is up to ±0.20.

Appendix B — Best and worst class deltas

Top 5 classes where D improves over B at epoch 160:

Class	B IoU	D IoU	Δ
truck	80.24 ± 4.39	85.54 ± 2.55	**+5.29**
wall	57.92 ± 1.59	61.64 ± 2.06	**+3.73**
bus	89.75 ± 2.03	92.12 ± 0.55	**+2.37**
terrain	66.10 ± 0.64	67.61 ± 0.46	+1.51
fence	66.07 ± 1.27	67.02 ± 0.35	+0.95

Bottom 5 classes where D regresses vs B at epoch 160:

Class	B IoU	D IoU	Δ
traffic light	76.85 ± 0.14	74.76 ± 0.35	−2.10
train	83.74 ± 1.54	82.63 ± 1.10	−1.10
traffic sign	83.52 ± 0.16	82.65 ± 0.36	−0.86
bicycle	79.81 ± 0.06	79.63 ± 0.20	−0.18
rider	68.49 ± 0.95	68.34 ± 0.45	−0.15

References

- Chen *et al.* (2017). *Rethinking atrous convolution for semantic image segmentation*. arXiv:1706.05587.
- Cheng *et al.* (2022). *Masked-attention mask transformer for universal image segmentation*. CVPR.
- Cordts *et al.* (2016). *The Cityscapes dataset for semantic urban scene understanding*. CVPR.
- Csurka *et al.* (2013). *What is a good evaluation measure for semantic segmentation?* BMVC.
- Karimi & Salcudean (2019). *Reducing the Hausdorff distance in medical image segmentation*. IEEE TMI 39 (2).
- Kervadec *et al.* (2019). *Boundary loss for highly unbalanced segmentation*. MIDL. arXiv:1812.07032.
- Lin *et al.* (2017). *Focal loss for dense object detection*. ICCV.
- Long *et al.* (2015). *Fully convolutional networks for semantic segmentation*. CVPR.
- Milletari *et al.* (2016). *V-Net: fully convolutional neural networks for volumetric medical image segmentation*. 3DV.
- Perazzi *et al.* (2016). *A benchmark dataset and evaluation methodology for video object segmentation*. CVPR.
- Sun *et al.* (2019). *High-resolution representations for labeling pixels and regions*. arXiv:1904.04514.
- Wang *et al.* (2022). *InternImage: exploring large-scale vision foundation models with deformable convolutions*. arXiv:2211.05778.
- Woo *et al.* (2023). *ConvNeXt V2: co-designing and scaling ConvNets with masked autoencoders*. CVPR. arXiv:2301.00808.
- Xiao *et al.* (2018). *Unified perceptual parsing for scene understanding*. ECCV. arXiv:1807.10221.
- Xie *et al.* (2021). *SegFormer: simple and efficient design for semantic segmentation with transformers*. NeurIPS.

Manuscript — 2026-06-02. Source code, configs, per-epoch metrics, figure-generation script, and interactive viewer: github.com/guillaume-cassez/city-scape. *Author: Guillaume Cassez, independent researcher, guillaume-cassez.fr — currently looking for ML / computer vision engineering opportunities.*