DEVOSEG: BUDGET-AWARE SELF-EVOLVING SEMANTIC SEGMENTATION UNDER VISUAL DEGRADATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Semantic segmentation under visual degradation is challenging because class cues and object boundaries deteriorate differently across images. Effective inference therefore calls for decisions that adapt to the current image and improve through experience from earlier cases. We introduce DevoSeg, a self-evolving framework that learns how to deploy a frozen segmentor from completed segmentations. It (1) estimates the complementary, Direct-relative opportunity of policy sets and assigns an image-dependent execution budget; (2) compares masks from selected policies with an independently produced Direct mask using post-execution image–mask evidence; and (3) organizes observed effects and costs as contextual knowledge, with a disjoint-group validation rule for proposed decision-state updates. Separating policy acquisition from mask selection lets past outcomes guide which alternatives to explore while the final decision uses the predictions actually obtained. On 200 ACDC images, DevoSeg's mask selector improves pooled Cityscapes-19 mIoU by 2.084 percentage points over Direct. Image-adaptive acquisition executes 30.1% fewer candidate policies and reduces their collection time by 23.3% relative to fixed eight-policy acquisition.

1 INTRODUCTION

Semantic segmentation under visual degradation remains difficult because appearance changes suppress class evidence and obscure boundaries in image-dependent ways (Sakaridis et al., 2021). A strong predictor can still miss a small structure or confuse neighboring classes when the visual evidence changes. Such failures can be local and class-dependent: an image may preserve its broad layout while losing the boundary of a thin or partially obscured object. Consequently, a degradation label alone cannot determine which part of the prediction needs correction. Direct inference produces one mask, while restoration, prompt variation, and refinement can expose alternative predictions. The central challenge is to decide which alternatives are worth obtaining for a particular image and whether any should replace Direct. These decisions should also benefit from earlier cases: an intervention that repeatedly helps in one context may be unproductive in another. Figure 1 situates this problem against direct segmentation and fixed tool composition.

The two decisions must be made with different evidence. Before executing a policy, the system has the image, its semantic queries, the Direct result, and experience from completed images, but cannot inspect the candidate mask it has yet to produce. It must allocate a limited budget to policies whose outcomes may be useful together. Once those policies run, their masks reveal image-specific effects that could not be known at acquisition time. For example, an operation may recover a missing class yet alter another region; two individually promising operations may correct the same error and add little complementary value. A Router should therefore retain diverse opportunities rather than simply take the highest independent route scores, while a Comparator should judge the realized masks against Direct. Executing every policy to compare its mask would incur restoration, segmentor, and refinement work even on images that need few alternatives. Acquisition and selection are coupled: an unexecuted policy cannot be chosen, and an executed policy yields no benefit if its mask is not recognized as useful.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

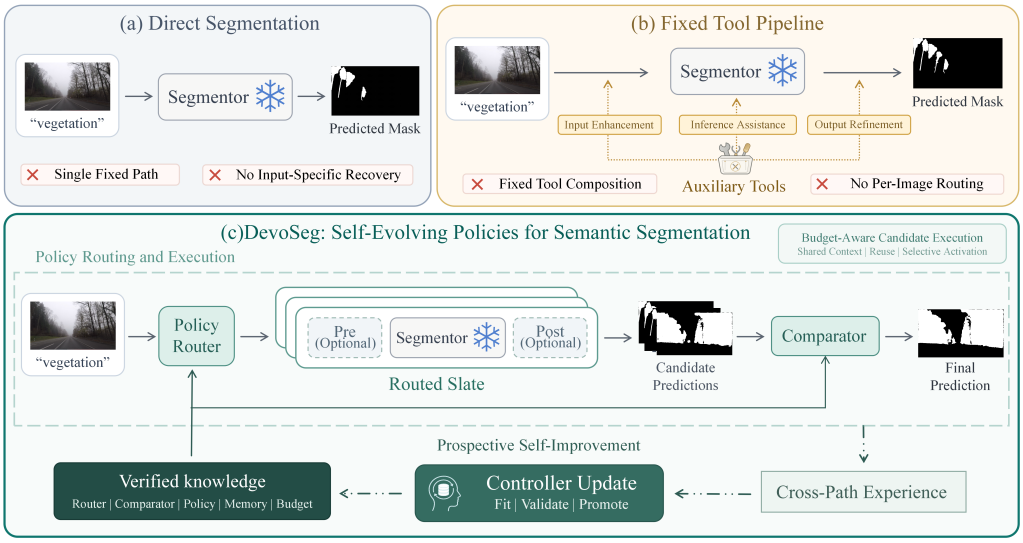


Figure 1: Direct segmentation and a fixed tool pipeline follow predetermined paths. DevoSeg routes policies per image, compares their masks, and uses completed cases to update its decision state.

We address this problem with DevoSeg, a self-evolving decision framework around a frozen SAM 3 segmentor (Carion et al., 2026). Completed segmentations provide contextual evidence about policy effects and execution costs; that evidence informs subsequent budget allocation and policy routing without changing the segmentor weights. Rather than storing successes alone, the decision state relates what was observable before execution to the mask changes and costs observed afterward. Later images can use this relationship to estimate both the potential improvement of a policy set and the work needed to obtain it. For each new image, the Router first chooses a compatible set of policies under an image-dependent budget. The Comparator then uses the resulting image–mask evidence to select among their predictions and independently produced Direct. Finally, the framework organizes observed outcomes as policy knowledge and defines disjoint-group validation for proposed decision-state updates. This design lets the same predictor be used more selectively across images while grounding each final mask choice in evidence from the current one. Figure 1 gives the overall design.

Our technical contributions are threefold:

(1) We formulate acquisition as a set-valued, Direct-relative opportunity problem: the Router estimates the improvement accessible through a compatible set under an image-dependent budget, rather than treating each route as an isolated choice.

(2) We separate outcome assessment from acquisition, using post-execution image–mask and region evidence to score executed masks against independent Direct; retaining Direct and selecting a candidate follow the same comparison rule.

(3) We define a cross-image update mechanism that consolidates contextual policy effects, with a disjoint-group validation criterion for promoting a proposed Router, Comparator, and budget state.

2 Related Work

2.1 Segmentation under Adverse Conditions

ACDC (Sakaridis et al., 2021) pairs adverse-condition scenes with normal-condition views and annotations, exposing the difficulty of semantic segmentation in fog, night, rain, and snow. SegFormer (Xie et al., 2021) develops multiscale transformer representations, while MaskFormer (Cheng et al., 2021), Mask2Former (Cheng et al., 2022), and OneFormer (Jain et al., 2023)

organize prediction around learned masks. These methods primarily improve how a segmentation model represents and decodes an image. Promptable SAM (Kirillov et al., 2023) and SAM 3 (Carion et al., 2026) instead provide a shared segmentor that can be queried through different prompts and inference procedures. DSS (Yang et al., 2026) selects among SAM masks for camouflaged objects. DevoSeg also decides which policies should produce masks in the first place, then compares the executed outputs with an independent Direct prediction.

2.2 Budgeted Inference and Routing

Dynamic routing in semantic segmentation allocates computation among paths inside a network subject to a resource budget (Li et al., 2020). Cost-aware model routing chooses between language models according to expected quality and cost (Ong et al., 2025), while prompt compression reduces tokens within an individual model interaction (Jiang et al., 2023). DevoSeg allocates work at the policy level around a frozen segmentor: restoration, prompting, and refinement routes are activated only when selected for the image. Unlike choosing one model or shortening one prompt, the Router may acquire several masks whose corrections are complementary. Because policies can share operations and yield correlated outputs, their joint opportunity cannot be inferred by adding independent route scores. The budget controls the number of candidate policies, and the measured efficiency results concern their physical execution and collection time.

2.3 Self-Evolving Decision Policies

Test-time adaptation methods such as Tent, CoTTA, EATA, and SAR adjust predictor parameters as the input distribution changes (Wang et al., 2021; 2022; Niu et al., 2022; 2023). A different line of work uses completed trials as experience: Reflexion retains feedback in episodic memory, Voyager builds reusable skills, and GEPA evolves prompts through reflection (Shinn et al., 2023; Wang et al., 2024; Agrawal et al., 2026). DevoSeg brings this cross-case perspective to segmentation decisions without updating the frozen segmentor. Its experience records link pre-execution context to observed mask effects and execution costs, informing which policies are attempted later. The final mask is still chosen from evidence on the current image, and proposed changes to routing, comparison, and budget knowledge are checked on disjoint development groups before acceptance.

3 Method

3.1 Task and Decision State

Let x_i be an image, $\mathcal{Q}_i$ its semantic queries, and F_θ a frozen segmentor. A policy π combines compatible optional operations around F_θ, such as restoration, prompting, and mask refinement. Direct policy π_0 produces M_{i0} independently. The Router chooses a set E_i from policy library $\mathcal{L}_t$; the Comparator then selects among the resulting masks and Direct. The decision state $B_t = (R_{\phi_t}, C_{\psi_t}, \tau_{\omega_t}, \mathcal{L}_t, \mathcal{K}_t)$ contains the Router, Comparator, budget policy, library, and accepted knowledge. For a labeled development image, $\Delta_{i\pi} = q(M_{i\pi}, y_i) - q(M_{i0}, y_i)$ is the gain of policy π over Direct under segmentation quality measure q. Labels train and evaluate decisions but are unavailable when selecting a mask.

3.2 Pre-execution Evidence and Routing

The Router sees o_i: image features, class-query encodings, Direct-mask evidence, uncertainty and geometry, and any acquired semantic observation. Retrieval from accepted knowledge forms $z_i = [o_i, \mathrm{Retrieve}(o_i; \mathcal{K}_t)]$. Semantic observations can describe visible classes or likely confusions. They contain neither candidate-mask features, which do not yet exist, nor quality labels from the current image. Each policy has a descriptor encoding its operations and compatibility constraints.

Given budget K_i, the Router scores feasible policies and retains a set E_i with $|E_i| \le K_i$. The following set-level objective describes the improvement that such a set can make available:

$$\widehat{U}_i(E) = \widehat{\mathbb{E}}\left[\max\left(0, \max_{\pi \in E} \Delta_{i\pi}\right) \mid z_i\right], \qquad E_i \approx \arg\max_{E \in \mathcal{F}(K_i)} \widehat{U}_i(E). \tag{1}$$

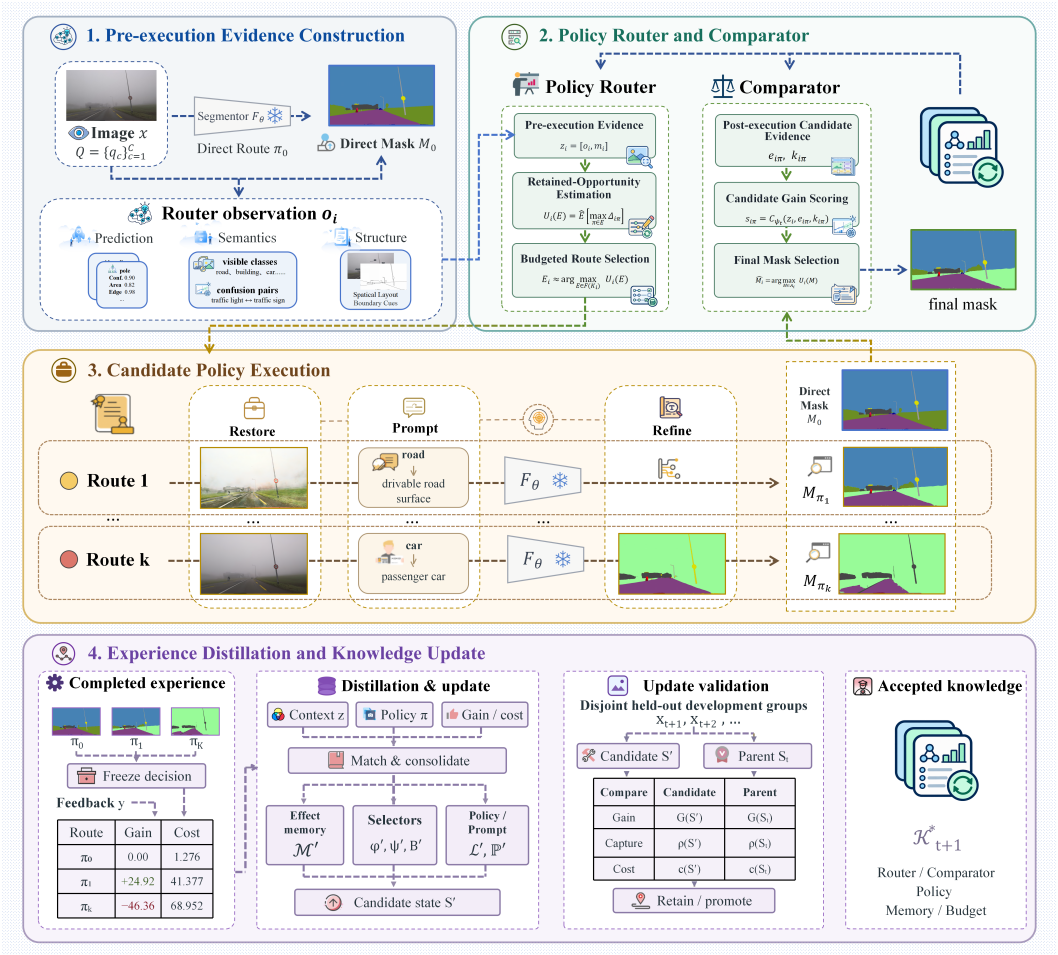


Figure 2: DevoSeg's decision process. The Router chooses policies before execution; the Comparator selects among their masks and the independent Direct mask afterward. Completed outcomes provide evidence for subsequent knowledge updates.

Here $\mathcal{F}(K_i)$ enforces budget and compatibility; the zero term represents independent Direct without consuming a candidate slot. Figure 2 abbreviates this reference option in its opportunity score. Development outcomes provide Direct-relative targets for route scores and the budget policy. The maximum inside the expectation expresses the opportunity retained by a set: two policies that tend to fix the same error offer less coverage than policies with complementary outcomes. It therefore guides acquisition without treating an individual route score as the final mask decision. Final quality also depends on the Comparator. Shared operations may make execution time nonadditive, so cost is measured for the executed set.

3.3 Post-execution Comparison

After execution, the Comparator receives candidate evidence $e_{i\pi}$: class scores, occupancy, boundary and region structure, and image–mask features. It predicts Direct-relative gain $s_{i\pi} = C_{\psi_t}(z_i, e_{i\pi}, \mathcal{K}_t)$, using relevant context from accepted knowledge when available. Selection is a single maximization over the executed policies and Direct:

$$\hat{\pi}_i = \arg \max_{\pi \in E_i \cup \{\pi_0\}} s_{i\pi}, \qquad s_{i\pi_0} = 0, \qquad \hat{M}_i = M_{i\hat{\pi}_i}. \tag{2}$$

Direct is assigned zero relative gain and lies outside the candidate budget. It wins when every candidate score is lower. Development targets compare each executed candidate with its image's independent Direct result; the Comparator never reads the current image's annotation at inference.

3.4 Experience Distillation and Knowledge Update

Completed episodes provide policy identities, selected masks, observed effects, and execution costs. Distillation aggregates this evidence across images into contextual policy support and effect estimates. Unlabeled observations may enrich descriptive memory, while gain targets require labeled development outcomes.

An experience links the observation available before execution to a policy's Direct-relative effect and its acquisition cost. Retrieving similar experiences can therefore change the value assigned to a policy set without assuming that one policy is uniformly useful across degradation conditions. The Comparator makes a second, image-specific judgment after seeing the executed masks. This separation lets past outcomes guide acquisition while the final choice remains grounded in the predictions obtained for the current image.

A candidate update is fitted using prior development episodes and evaluated on later groups disjoint from its construction data:

$$\widetilde{B}_t = \mathrm{Fit}(B_t, \mathcal{D}_t), \qquad B_{t+1} = \mathrm{Select}(B_t, \widetilde{B}_t; \mathcal{V}_t), \quad \mathcal{D}_t \cap \mathcal{V}_t = \varnothing. \tag{3}$$

The proposal is retained only if it meets the validation criterion on those later groups; the knowledge snapshot and decision weights are updated together.

4 Experiments

4.1 Data, Protocol, and Metrics

We evaluate on 200 ACDC images (Sakaridis et al., 2021), balanced across fog, night, rain, and snow (50 per condition). In this evaluation, $\mathcal{Q}_i$ contains the Cityscapes-19 class queries. All matched comparisons use the same class mapping, ignored-pixel rule, image resizing, and frozen SAM 3 backbone (Carion et al., 2026). Direct is executed separately for each image; image, route, and mask identities are verified for each comparison.

Our primary statistic averages present-class IoU over images. Paired gains compare each selected mask with that image's independent Direct mask and are expressed in percentage points (pp). We also report dataset-pooled Cityscapes-19 mIoU: confusion counts are accumulated across images before the 19 class IoUs are averaged. Confidence intervals use 10,000 paired image-bootstrap resamples for the frozen three-seed ensemble. Candidate coverage is the fraction of images for which the Comparator selects an executed policy instead of Direct.

The Comparator studies use 1,080 development image groups with independently executed Direct masks and 8,640 executed candidate masks. We fit on 920 groups and reserve 160 development groups; the 200 evaluation images are disjoint from both. The native feature extractor is trained without evaluation annotations. All candidate choices are recorded before the evaluation labels are opened for scoring. The evaluation pool contains 200 independently executed Direct masks and 1,600 physically executed fixed-$K = 8$ candidate masks.

4.2 Comparator Evidence on Executed Masks

To isolate post-execution evidence, four Comparators choose from the same eight masks per image and the same Direct option. Architecture, training groups, three per-seed initial states, optimizer, and update count are matched; only the evidence presented to the Comparator varies. Image–mask evidence describes the predicted regions and their correspondence with the input; region evidence uses changed-region features from DINOv2 spatial maps (Oquab et al., 2024). The combined arm uses both. Scores are averaged across the three seeds before applying Eq. 2.

Comparator input	Image IoU (%)	ΔDirect (pp)	Pooled mIoU (%)
Independent Direct	55.732	–	62.795
Context + route	55.581	−0.151	62.755
Region features	55.835	+0.103	64.450
Image–mask features	56.407	+0.675	64.835
Full Comparator (ours)	**56.652**	**+0.920**	**64.879**

Table 1: Comparator inputs on the same executed fixed-$K = 8$ masks (200 ACDC images). The full Comparator combines image–mask and DINOv2 region features. Image IoU is the mean of per-image present-class IoU; pooled mIoU averages class IoUs after confusion counts are pooled over images.

The full Comparator improves image IoU by 0.920pp over Direct (paired 95% interval $[0.391, 1.451]$pp). Dataset-pooled mIoU increases by 2.084pp (paired image-bootstrap interval $[-0.092, 4.298]$pp). The input ablation identifies the value of inspecting a mask after it has been produced: context and route identity alone do not capture the image-specific outcome, whereas image–mask features raise the mean by 0.675pp over Direct. Combining them with region features gives the highest mean in Table 1. The weather breakdown in Appendix A shows positive mean gains in each condition, with the largest gain on snow.

Figure 3 shows changes in vegetation, road edges, and thin structures. The yellow overlay marks changed predictions rather than ground-truth correctness; these outcome-ranked cases illustrate visible differences, while Table 1 reports the complete 200-image evaluation. A 50-case gallery accompanies the paper.

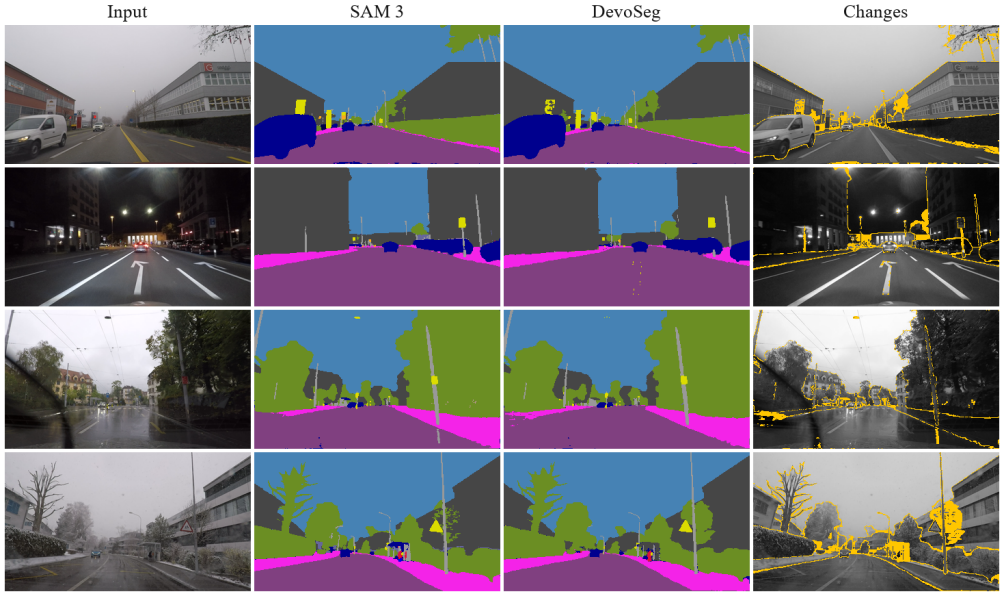


Figure 3: Qualitative comparison on ACDC. SAM 3 is Direct; yellow highlights pixels assigned different classes by DevoSeg.

4.3 Matched Decision-Module Training

We next vary which decision heads are trained. Router-only, Comparator-only, budget-only, and joint arms start from the same per-seed state and traverse the same 920 groups in the same order. Each receives 870 updates with AdamW (2×10^{-4} learning rate, 10^{-3} weight decay) over 30 epochs. A fixed update mask gives each arm 599,432 effective trainable parameters. We evaluate three seeds. Enabled heads update, disabled heads remain fixed, and evaluation labels are absent from inference.

The budget head chooses among $K \in \{1, 3, 5, 8\}$, controlling which prefix of the executed K8 pool an arm may inspect. Direct remains the independent zero-gain option.

Updated heads	Image IoU (%)	ΔDirect (pp)
Router only	55.482	-0.250
Comparator only	56.254	$+0.522$
Budget policy only	54.978	-0.754
Router + Comparator + budget	**56.369**	$+\mathbf{0.637}$

Table 2: Decision-head training on the same executed K8 pool (200 images). Each arm has 599,432 effective trainable parameters and 870 updates per seed.

The joint arm improves over Direct by 0.637pp (paired 95% interval $[0.232, 1.072]$pp) and has the highest mean among the matched decision-head arms. Comparator-only training also improves over Direct, consistent with the post-execution evidence study above: better selection can help even when acquisition is held to the same executed pool. The matched starts, training trajectories, and parameter counts in Table 2 keep this comparison focused on which heads learn. Pairwise contrasts appear in Appendix A.3.

4.4 Actual Candidate-Execution Cost

The collection experiments execute the requested policies rather than reading them from a stored pool. Dynamic acquisition runs 5.59 policies per image on average, 30.1% fewer than fixed $K = 8$. Collection takes 110.171 versus 143.687 seconds per image: a paired saving of 33.516 seconds (23.3%; 95% interval $[28.172, 39.079]$ seconds). Restoration calls average 11.050 versus 16.015. Separately executed fixed-budget campaigns give the cost curve in Table 3 and Figure 4.

Executed set	Total routes	Routes/image	Time (s/image)	Saved vs $K = 8$ (s/image) [95% CI]
Fixed $K = 1$	200	1.00	29.937	113.750 [110.498, 116.929]
Fixed $K = 3$	600	3.00	71.513	72.174 [69.898, 74.391]
Fixed $K = 5$	1,000	5.00	105.393	38.294 [36.470, 40.098]
Dynamic K	1,118	5.59	110.171	33.516 [28.172, 39.079]
Fixed $K = 8$	1,600	8.00	143.687	–

Table 3: Measured candidate collection on 200 images. Time savings are paired against fixed $K = 8$; brackets give 95% bootstrap intervals.

Figure 4 resolves the aggregate saving by weather. Fixed-budget collection time rises with the number of executed policies in every condition, while the dynamic policy assigns different budget mixes: 28 of 50 snow images receive eight policies, compared with 14 of 50 night images. Measured savings against eight-policy execution remain positive in each weather group, ranging from 26.482 to 39.988 seconds per image. The accompanying budget counts make clear that the reduction comes from fewer policy executions, not a cheaper implementation of an individual route. Collection time excludes comparison and other serving stages; the physical-cost runs execute policies without Comparator selection. Exact weather-level measurements are given in Appendix B.

4.5 External Model Context

Table 4 compares predictions from external segmentation models on the same 200 images; the training-label counts distinguish their protocols.

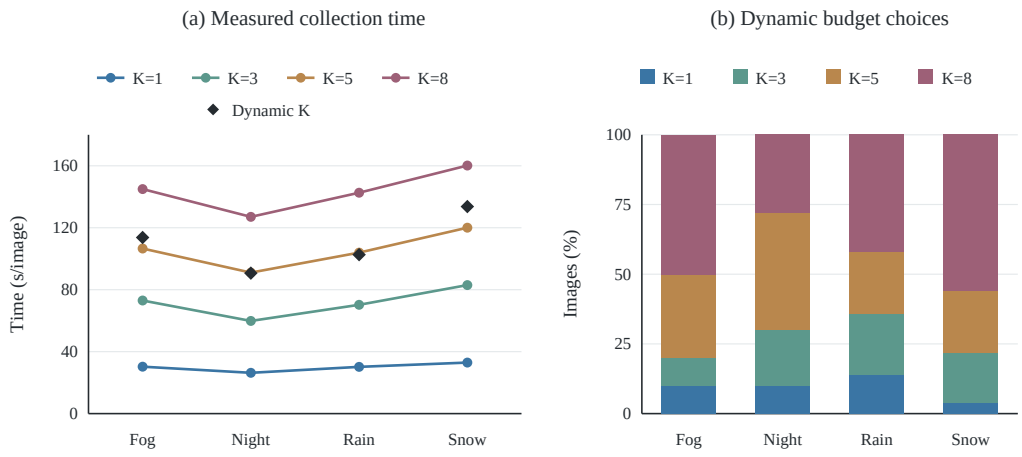


Figure 4: Measured candidate-collection cost across four ACDC conditions (50 images each). (a) Time for fixed $K \in \{1, 3, 5, 8\}$ and dynamic acquisition. (b) Dynamic budget allocation.

Model	ACDC training labels	Image IoU (%)
Independent SAM 3 Direct	0	55.732
DevoSeg (full Comparator, $K = 8$)	920	56.652
SegFormer (Xie et al., 2021)	0 (Cityscapes checkpoint)	43.973
MaskFormer (Cheng et al., 2021)	0 (Cityscapes checkpoint)	40.037
Mask2Former (Cheng et al., 2022)	0 (Cityscapes checkpoint)	42.005
OneFormer (Jain et al., 2023)	0 (Cityscapes checkpoint)	52.834
SegFormer	1,400	55.922
MaskFormer	1,400	53.036
Mask2Former	1,400	48.419

Table 4: Contextual comparison on the same 200 ACDC images. Scores are mean per-image present-class IoU; training-label counts identify the protocols. DevoSeg uses the fixed-pool Comparator of Table 1.

A *Cityscapes checkpoint* is applied directly to ACDC without ACDC training; rows with 1,400 ACDC labels fine-tune the corresponding model on labeled ACDC images. DevoSeg's decision components use 920 labeled ACDC development groups. Tables 1 and 2 compare DevoSeg components under matched protocols.

Among the external models shown, ACDC-trained SegFormer has the highest image IoU at 55.922%, while DevoSeg reaches 56.652% with a frozen segmentation backbone. These rows provide context across training regimes; the within-method ablations above isolate the effect of the decision components.

Sequential test-time adaptation and four-shot SegGPT (Wang et al., 2023) follow different protocols; their scores and 168-image results appear in Appendix A.4. The external-model gallery uses 50 outcome-ranked illustrative images; quantitative scores use all 200.

Figures 5 and 6 align each input with all method predictions at the same scale. The four-weather panel makes differences at road boundaries, roadside structures, and larger regions visible within the same scene. The second panel adds an ACDC-trained SegFormer beside the unadapted checkpoints, so the effect of using in-domain labels can be read alongside the masks produced around a frozen segmentor.

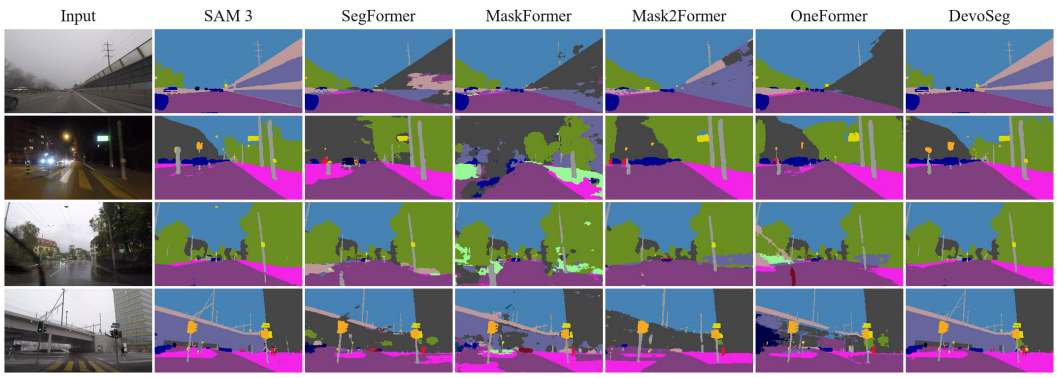


Figure 5: Same-image segmentation comparison on ACDC. SAM 3 is Direct; SegFormer, MaskFormer, Mask2Former, and OneFormer use Cityscapes checkpoints.

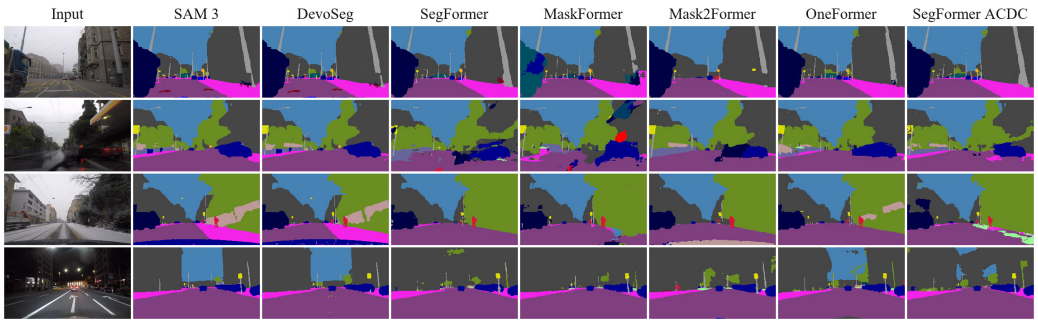


Figure 6: Same-image ACDC comparison, including a SegFormer checkpoint trained on 1,400 ACDC images.

5 Conclusion

DevoSeg separates two decisions around a frozen segmentor: which policies to execute and which resulting mask to retain. Cross-image experience informs both stages, and proposed contextual knowledge updates are governed by held-out validation. The ACDC results show improved mask selection and reduced candidate-collection work, supporting policy learning as a complement to adapting segmentor weights.

Separating acquisition from selection lets DevoSeg allocate computation to promising alternatives while judging each realized prediction against Direct. The same decision structure can accommodate different policy libraries and semantic queries without changing the underlying segmentor.

AI Use Statement

Generative AI tools assisted with research and manuscript preparation. Reported results were checked against experimental records, and the authors take responsibility for the submission.

REFERENCES

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alex Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. GEPA: Reflective prompt evolution can outperform reinforcement learning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/0e9e708b6f48e14fd0ac29e167413f76-Abstract-Conference.html.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, et al. SAM 3: Segment anything with concepts. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/e0982cbc81401df3430ee1ff780dc7a2-Abstract-Conference.html.

Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/950a4152c2b4aa3ad78bdd6b366cc179-Abstract.html.

Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1290–1299, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Cheng_Masked-Attention_Mask_Transformer_for_Universal_Image_Segmentation_CVPR_2022_paper.html.

Jitesh Jain, Jiachen Li, Mang Tik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. OneFormer: One transformer to rule universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2989–2998, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Jain_OneFormer_One_Transformer_To_Rule_Universal_Image_Segmentation_CVPR_2023_paper.html.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13358–13376, 2023. doi: 10.18653/v1/2023.emnlp-main.825. URL https://aclanthology.org/2023.emnlp-main.825/.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4015–4026, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Kirillov_Segment_Anything_ICCV_2023_paper.html.

Yanwei Li, Lin Song, Yukang Chen, Zeming Li, Xiangyu Zhang, Xingang Wang, and Jian Sun. Learning dynamic routing for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8553–8562, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Li_Learning_Dynamic_Routing_for_Semantic_Segmentation_CVPR_2020_paper.html.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 16888–16905, 2022. URL https://proceedings.mlr.press/v162/niu22a.html.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=g2YraF75Tj.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M. Kadous, and Ion Stoica. RouteLLM: Learning to route LLMs from preference data. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/5503a7c69d48a2f86fc00b3dc09de686-Abstract-Conference.html.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=a68SUt6zFt.

Christos Sakaridis, Dengxin Dai, and Luc Van Gool. ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10765–10775, 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Sakaridis_ACDC_The_Adverse_Conditions_Dataset_With_Correspondences_for_Semantic_Driving_ICCV_2021_paper.html.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/pdf/4de0af9691a5dcc52de7de756676fded33d037ef.pdf.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=ehfRiF0R3a.

Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7201–7211, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Wang_Continual_Test-Time_Domain_Adaptation_CVPR_2022_paper.html.

Xinlong Wang, Xiaosong Zhang, Yue Cao, Wen Wang, Chunhua Shen, and Tiejun Huang. SegGPT: Towards segmenting everything in context. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1130–1140, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Wang_SegGPT_Towards_Segmenting_Everything_in_Context_ICCV_2023_paper.html.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. SegFormer: Simple and efficient design for semantic segmentation with transformers. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/64f1f27bf1b4ec22924fd0acb550c235-Abstract.html.

Yilong Yang, Jianxin Tian, Shengchuan Zhang, and Liujuan Cao. Discover, segment, and select: A progressive mechanism for zero-shot camouflaged object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 34745–34754, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Yang_Discover_Segment_and_Select_A_Progressive_Mechanism_for_Zero-shot_Camouflaged_CVPR_2026_paper.html.

A Additional Evaluation Detail

A.1 Weather Conditions

Table 5 reports the full-200 weather breakdown for the full Comparator and the matched-budget joint arm. The four weather groups have 50 images each. The full Comparator chooses an executed candidate rather than independent Direct for 60.5% of images; its per-image gain is negative for 23.5% and below -5pp for 4.5%.

Weather	Direct mean IoU (%)	Comparator gain (pp)	Joint gain (pp)
Fog	59.612	$+0.124$	-0.093
Night	49.704	$+0.099$	$+0.175$
Rain	58.842	$+0.467$	$+0.594$
Snow	54.770	$+2.988$	$+1.874$

Table 5: Mean per-image present-class IoU and direct-relative gains on the complete 200-image evaluation. Each weather group contains 50 images. The full-Comparator and joint rows come from different matched-control studies on the same executed fixed-K8 candidate pool.

A.2 Additional Descriptive Subset

The fixed 168-image subset comprises 42 images per weather condition and was selected after outcomes of an earlier campaign had been inspected. It was not used to fit or select the clean models. On this subset, the full Comparator gains $+1.115$pp in mean per-image IoU and $+2.632$pp in dataset-pooled Cityscapes-19 mIoU against independent Direct. The matched-budget joint arm gains $+0.774$pp in mean per-image IoU, while the Comparator-only arm gains $+0.670$pp. The primary comparisons use all 200 images.

A.3 Matched Decision-Head Contrast

The joint arm improves mean per-image IoU by $+0.115$pp relative to Comparator-only on the same fixed candidate pool (paired 95% image-bootstrap interval $[-0.316, 0.542]$pp).

A.4 External Adaptation and Task-Extension Protocols

The SegFormer variants apply sequential test-time adaptation to a common source model. SegGPT (Wang et al., 2023) uses four-shot task examples instead of a native 19-class semantic head. Table 7 reports their observed scores under these protocols.

External model	Mean IoU, 200 (%)	Mean IoU, selected 168 (%)
Independent SAM 3 Direct	55.732	55.591
DevoSeg full Comparator, fixed $K = 8$	56.652	56.706
SegFormer (Cityscapes checkpoint)	43.973	44.239
MaskFormer (Cityscapes checkpoint)	40.037	40.078
Mask2Former (Cityscapes checkpoint)	42.005	42.252
OneFormer (Cityscapes checkpoint)	52.834	53.427
SegFormer, ACDC-1400	55.922	56.333
MaskFormer, ACDC-1400	53.036	53.481
Mask2Former, ACDC-1400	48.419	48.724

Table 6: Scores on the full set and descriptive selected subset. Each value is mean per-image present-class IoU; DevoSeg uses the matched fixed-$K = 8$ Comparator, while ACDC-1400 rows use 1,400 labeled target-domain training images.

Adaptation method	Mean IoU, 200 (%)	Mean IoU, selected 168 (%)
SegFormer + Tent-LN (Wang et al., 2021)	37.387	37.277
SegFormer + SAR-LN (Niu et al., 2023)	43.988	44.219
SegFormer + CoTTA-LN (Wang et al., 2022)	43.965	44.194
SegFormer + EATA-LN (Niu et al., 2022)	43.864	44.112
SegGPT four-shot (Wang et al., 2023)	20.795	20.625

Table 7: Existing predictions re-scored on the same ACDC image identities. Adaptation runs are sequential and SegGPT follows a separate four-shot protocol. Values are mean per-image present-class IoU, not dataset-pooled mIoU.

B Candidate Collection Cost

Physical acquisition	Fog	Night	Rain	Snow
Fixed $K = 1$	30.292	26.336	30.176	32.945
Fixed $K = 3$	73.016	59.827	70.239	82.970
Fixed $K = 5$	106.577	91.030	103.936	120.029
Dynamic K	113.701	90.689	102.620	133.673
Fixed $K = 8$	144.960	127.027	142.608	160.154

Table 8: Measured candidate collection time in seconds per image by weather (50 images per condition). Full-set averages and paired savings are in Table 3.

Weather	$K = 1$	$K = 3$	$K = 5$	$K = 8$	Mean K	Saved (s/image)
Fog	5	5	15	25	5.90	31.259
Night	5	10	21	14	5.04	36.338
Rain	7	11	11	21	5.26	39.988
Snow	2	9	11	28	6.16	26.482
All	19	35	58	88	5.59	33.516

Table 9: Dynamic budget allocation by weather. Each row contains 50 images; savings compare measured collection time with fixed $K = 8$ on the same images.

K1, K3, K5, K8, and dynamic acquisition were each physically executed. Relative to K8, same-route masks differ in pixels for 11/200 K1, 63/600 K3, and 132/1,000 K5 cases; Table 3 therefore reports measured collection cost for each execution. The campaigns made zero online language-model requests. Under a uniform t tokens per route after shared cost T_{shared}, the dynamic policy count implies a modeled route-dependent reduction of $2.41t$ tokens per image, or $2.41t/(T_{\text{shared}} + 8t)$ of modeled total tokens.

C Implementation and Evaluation Protocol

For each evaluated image, the policy decision is fixed before its label is read. Image, route, and decoded-mask identities are matched to the independently executed Direct and candidate outputs before metrics are aggregated. Candidate collection costs are measured separately from mask selection.

Training targets use independent Direct masks from 1,080 development groups and 8,640 executed candidate masks. All 200 evaluation images are excluded from fitting. The four input arms use 920 fitting groups and three seeds. The joint training study uses a common per-seed initial state, matching data order, 870 updates, and 599,432 effective trainable scalars in each arm. Evaluation labels are read only after the four arms record their choices. Independently executed Direct has score zero outside the eight candidate slots.

C.1 Decision Models and Paired Contrast

The fixed-pool Comparator maps a 269-dimensional context vector, a 15-dimensional policy descriptor, and a 4,822-dimensional visual vector to 384-dimensional embeddings. Its visual vector has 4,052 image–mask dimensions and 770 DINOv2 region dimensions. A cross-attention block reads 64 learned memory vectors before prediction heads score candidate gain and auxiliary targets. The four input arms mask the relevant visual blocks while retaining the same architecture and initialization for each seed. Their candidate gain predictions are averaged across seeds, then compared with the independent Direct score of zero. The paired full-minus-image–mask-only image-IoU contrast is $+0.245$pp (10,000-resample image-bootstrap interval $[-0.260, 0.752]$pp).

The matched Router projects context and policy descriptors to a 192-dimensional set encoder with two Transformer layers and six attention heads. Route scores are supervised by per-policy Direct-relative gains; for each budget, the budget head targets the best nonnegative gain among policies in the Router-ordered prefix. The budget head predicts utilities over $K \in \{1, 3, 5, 8\}$. The heads are trained on the same eight executed masks per development image and compared under the 599,432-effective-parameter update mask. At evaluation, Router scores order the existing eight policies, the budget head selects a prefix length, and the Comparator compares the resulting masks with independent Direct.

C.2 Qualitative Examples

The accompanying 50-case plate contains 13 fog, 13 night, 12 rain, and 12 snow images from the full 200-image evaluation. Cases are ranked by measured Direct-relative gain within each weather condition for illustration; this ranking is not used for model fitting or quantitative aggregation. Each row shows the input, independent Direct overlay, and selected DevoSeg overlay. Figure 3 shows four such examples; Table 1 uses all 200 images.