Trainable Topology Supervision under Structurally Unreliable Pseudo Supervision

Jieyang Zhou[1, 2†] **Ziliang Wang**[1†] **Bin Hu**[3] **Ying Yang**[2]
Aoli He[2] **Xianhong Wen**[2] **Kehua Guo**[1, 2*]
[1]School of Artificial Intelligence, Central South University, Changsha 410083, China
[2]School of Computer Science and Engineering, Central South University, Changsha 410083, China
[3]School of Computer Science and Technology,
Changsha University of Science and Technology, Changsha 410076, China

Abstract

Topology supervision relies on meaningful structural targets, yet this assumption can fail under structurally unreliable pseudo supervision. In pseudo-label learning, the topology induced by pseudo masks may already contain distorted components, broken connections, or spurious holes, making direct supervision-side topology fitting unreliable and potentially propagating structural errors. We address this problem by reformulating topology supervision as trainable topology-side correction. Specifically, we propose a differentiable persistent-homology proxy loss that derives topology signals from a soft Euler-characteristic trajectory and threshold-wise structural variation, without computing or matching persistence barcodes. We further introduce a conservative topology-consistency injection mechanism that regulates when, how strongly, and in which direction these signals enter student optimization, while the teacher provides the noisy structural reference. We validate the proposed reformulation in unsupervised camouflaged object detection, a challenging setting where pseudo masks often contain severe structural distortions. Experiments under controlled same-carrier comparisons show consistent improvements in both task-level detection performance and topology-level structural reliability. Additional matched transfer experiments on histology segmentation provide evidence beyond UCOD. These results suggest that, under structurally unreliable pseudo supervision, topology supervision is more effective when derived as trainable proxy signals and introduced into optimization conservatively. Code is available at `https://github.com/zhoujieyang/TopoCons`.

1 Introduction

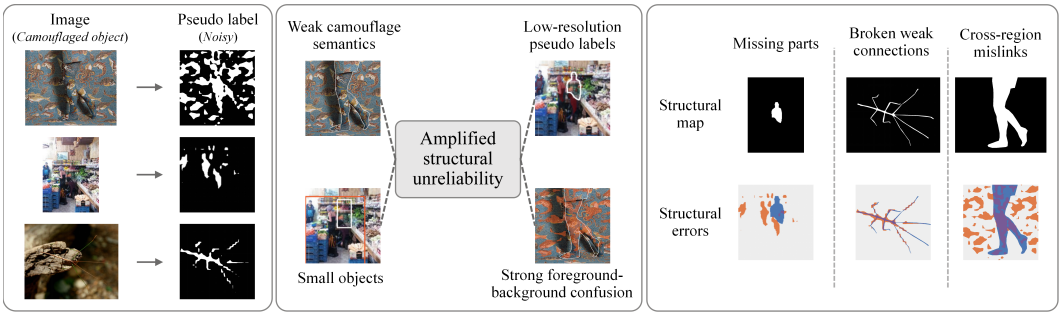
Topology supervision provides structural constraints beyond local pixel matching in dense prediction, especially when connected components and holes carry semantic meaning [1–3]. Existing topology-aware methods therefore exploit topology to improve structural correctness through formulations such as persistent-homology (PH) losses [4, 5], topology-preserving objectives [6], topology-aware modules [7, 8], topological transforms based on persistent homology and Euler characteristics [9, 10], and differentiable multiparameter persistence representations [11, 12]. Despite these differences, they largely share the same premise. The topology encoded by supervision is usable for direct alignment, matching, or structural regularization [13].

However, this premise becomes unreliable once dense prediction relies on pseudo supervision rather than human annotation, especially in self-training and teacher–student pipelines [14, 15].

*Corresponding author: `guokehua@csu.edu.cn`.
†These authors contributed equally.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

In such settings, supervision-side structure is no longer determined solely by reliable annotation, but is increasingly shaped by pseudo labels. When pseudo labels contain structural distortion, the supervision-side topology they induce can itself become unreliable. Pseudo-label errors in dense prediction do not remain purely local, but can be spatially organized and shape-dependent [16, 17], leading to structural distortions such as broken connections, missing components, and erroneous links, which may induce only limited pixel-level deviation while substantially altering the global structure [1, 2]. As a result, directly optimizing against supervision-side topology as a whole can fail to correct structural defects and may instead propagate pseudo-label-induced structural errors into training. The resulting problem is therefore no longer how to use supervision-side topology directly, but which topology signals remain trainable and how they should enter optimization.



(a) Fixed pseudo-mask generation introduces substantial supervision noise

(b) Structural unreliability is amplified in UCOD

(c) Typical structural distortions

Figure 1: Premise failure of direct topology supervision under structurally unreliable pseudo supervision in UCOD. (a) Fixed pseudo-mask generation introduces substantial supervision noise. (b) In UCOD, structural unreliability is further amplified by weak camouflage semantics, low-resolution pseudo masks, small objects, and strong foreground–background confusion. (c) The resulting pseudo-mask errors frequently appear as missing parts, broken weak connections, and cross-region mislinks.

Unsupervised camouflaged object detection (UCOD) is a concrete instance of this premise failure, as illustrated in Fig. 1. Fixed pseudo-mask generation in UCOD introduces substantial supervision noise [18]. In this setting, structural unreliability is further amplified by weak camouflage semantics, low-resolution pseudo masks, small objects, and strong foreground–background confusion [19, 20]. Under these conditions, pseudo-mask errors often manifest as missing parts, broken weak connections, and cross-region mislinks rather than isolated local deviations [21]. This setting therefore raises two coupled questions for trainable topology supervision. The first is which topology signals remain trainable when supervision-side structure itself may already be distorted [22–24]. The second is how these signals should enter optimization so that they can contribute without destabilizing the main supervision objective [21, 25].

To answer these two questions, we propose a differentiable PH proxy loss, denoted by $\mathcal{L}_{\mathrm{PHP}}$, together with a conservative topology-consistency injection mechanism. $\mathcal{L}_{\mathrm{PHP}}$ is motivated by PH-style event reasoning, but is defined through a differentiable soft Euler-characteristic trajectory and threshold-wise structural variation rather than explicit barcode matching. The injection mechanism constrains the entry, scale, and local direction of topology-side correction during optimization. The teacher provides a noisy structural reference, while correction acts on the student. We validate this reformulation in UCOD under the standard benchmark protocol and controlled same-carrier comparisons.

The main contributions of this paper are as follows.

1. We propose a differentiable PH proxy loss that derives trainable topology signals from a soft Euler-characteristic trajectory and threshold-wise structural variation, avoiding explicit barcode matching or direct fitting to distorted supervision-side topology.
2. We propose a conservative topology-consistency injection mechanism that regulates when and how topology signals enter optimization, enabling topology-side correction while reducing the risk of destabilizing the main supervision objective.
3. We provide systematic validation in UCOD under the standard UCOD benchmark protocol and controlled same-carrier comparisons, showing improvements in both task-level performance and topology-level structural reliability.

2 Related Work

2.1 Structurally Unreliable Pseudo Supervision in Dense Prediction

Pseudo supervision is widely used in dense prediction when reliable pixel-level annotation is unavailable. Representative strategies include cross pseudo supervision [26], cross-consistency training [27], unreliable pseudo-label modeling [14], weak-to-strong consistency [28], ranking-based regularization [29], and teacher-guided learning [15]. Existing analyses further reveal class- and pixel-level bias in pseudo labels [30, 31], as well as limitations of confidence-based pseudo-label selection [25, 32]. For dense prediction, however, pseudo-label reliability is not purely local. Errors can be spatially organized and shape-dependent, as reflected in studies on unreliable pseudo-label modeling [14], error localization [16], directional consistency [33], correlation- and shape-based propagation [17], and misleading pseudo labels [34]. This issue is amplified in UCOD, where fixed pseudo-mask generation [18, 19], weak camouflage semantics [18, 19], and high foreground–background similarity [20] increase the risk of missing parts, broken weak connections, and cross-region mislinks. These studies suggest that the key difficulty is not only pseudo-label usability, but also the reliability of supervision-side structure itself.

2.2 Topology Supervision under Structurally Unreliable Pseudo Supervision

Topology supervision is studied in dense prediction because pixel-level accuracy alone does not guarantee structural correctness [1, 2]. Existing formulations include PH losses [4, 5], persistent-homology and Euler-characteristic transforms [9, 10, 35], differentiable multiparameter persistence representations [11, 12], discrete-Morse-based objectives [36], induced barcode matching [37], topology-preserving losses such as clDice [6], topology-aware modules [7], and extensions to geometric constraints, strict topology preservation, and 3D segmentation [8, 38, 39]. Recent analysis also shows that topology-aware segmentation is sensitive to evaluation choices and implicit structural assumptions [13]. Despite their differences, many methods assume that supervision-side topology provides a sufficiently reliable structural reference for alignment, matching, or regularization. Recent work begins to relax this assumption under incomplete or noisy supervision through structure-level uncertainty [22], noise-aware topological consistency [23], adaptive topo-consistency [24], and spatial-aware persistent feature matching [40]. However, these methods generally estimate or exploit recoverable supervision-side structural information before imposing topology constraints. We study a stricter regime, where pseudo supervision may be structurally unreliable from the outset. The resulting question is which topology signals remain trainable, and how they should enter optimization, when supervision-side structure may already be distorted.

3 Method

Under structurally unreliable pseudo supervision, direct fitting to supervision-side topology becomes unreliable. We therefore reformulate topology supervision as trainable topology-side correction under constrained optimization. The reformulation consists of a differentiable PH proxy loss $\mathcal{L}_{\mathrm{PHP}}$, which derives trainable topology signals from threshold-wise structural variation, and a conservative topology-consistency injection mechanism, which controls how these signals enter the optimization dynamics. Together, they replace unreliable target-side topology fitting with conservative structural correction. Fig. 2 summarizes the framework.

3.1 Differentiable Persistent-Homology Proxy Loss

Many PH-based topology losses rely on explicit topological correspondence between the prediction and the supervision [4, 5]. This formulation is appropriate when the supervision-side topology is reliable, but becomes problematic when pseudo supervision itself contains structural distortion. In this case, direct target-side topology matching may reinforce erroneous components, broken connections, or spurious links induced by pseudo supervision. We therefore replace exact topological matching with a differentiable proxy defined on threshold-wise structural variation. The proxy is motivated by PH-style birth and death events, but it does not compute or match persistence barcodes. Instead, it extracts trainable topology-side signals from a soft Euler-characteristic trajectory and adjacent-threshold structural changes.

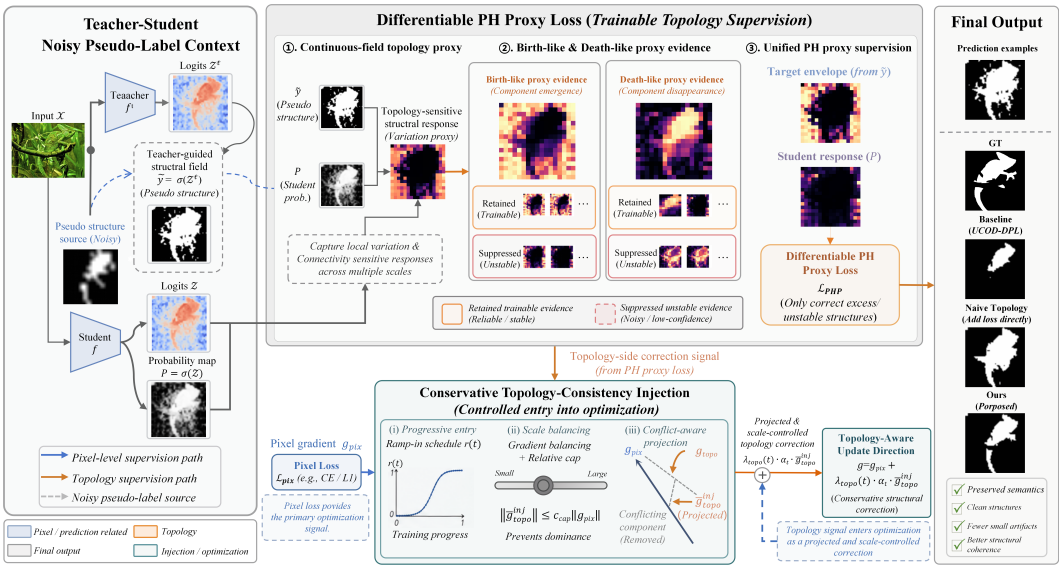


Figure 2: Overview of the proposed reformulation of topology supervision. Trainable topology signals are derived by the differentiable PH proxy loss and conservatively injected into optimization under structurally unreliable pseudo supervision.

Let x denote the input image. The student network produces a logit map $z = f_\theta(x)$ and the corresponding probability map $p = \sigma(z) \in [0,1]^{H\times W}$. In the noisy teacher–student setting, let z^t denote the teacher logit map for the same input, and define the teacher-guided probability field as $\tilde{y} = \sigma(z^t) \in [0,1]^{H\times W}$. In our formulation, $\tilde{y}$ serves as a noisy structural reference for topology-side correction rather than as an exact topological target to be matched directly. For notational continuity, quantities indexed by $\tilde{y}$ are still referred to as target-side terms, although $\tilde{y}$ is treated here as a noisy structural reference rather than as an exact topological target. Teacher-side quantities are held fixed during student differentiation.

To obtain a differentiable surrogate for threshold-wise topological events, we track how the soft Euler characteristic evolves as the threshold sweeps across a continuous scalar field $u \in [0,1]^{H\times W}$ [10]. The soft threshold response is defined in Eq. 1.

$$S(u, \tau_m) = \sigma\left(\frac{u - \tau_m}{\delta}\right), \tag{1}$$

where $\delta > 0$ is a smoothing coefficient. The corresponding soft Euler trajectory is defined in Eq. 2.

$$\chi_u(\tau_m) = V(S(u,\tau_m)) - E(S(u,\tau_m)) + F(S(u,\tau_m)), \tag{2}$$

where F, E, and V denote the soft face, edge, and vertex counts under the differentiable 4-neighborhood relaxation. Basic differentiability properties are given in Appendix B.2. Under the regularity assumptions in Appendix B.1, the active piecewise-smooth branches are locally fixed away from switching sets, making the instantiated proxy locally C^1 at regular points and differentiable almost everywhere.

Based on the discrete trajectory $\chi_u(\tau_m)$, adjacent-threshold variation induces birth-like and death-like proxy activities, as defined in Eq. 3.

$$\Delta\chi_u^{(m)} = \chi_u(\tau_{m+1}) - \chi_u(\tau_m), \qquad b_u^{(m)} = [\Delta\chi_u^{(m)}]_+, \qquad d_u^{(m)} = [-\Delta\chi_u^{(m)}]_+, \tag{3}$$

where $m = 1, \ldots, M-1$. These quantities are not exact persistence-pair multiplicities. Instead, they serve as differentiable event-level proxies for threshold-wise topological changes. For 2D binary sets, $\chi = \beta_0 - \beta_1$, so distinct Betti configurations may share the same Euler value; $\mathcal{L}_{\mathrm{PHP}}$ therefore remains a PH-style proxy rather than a replacement for persistence pairs or barcode-level descriptors. Each retained threshold interval contributes one event-level proxy term, weighted by event saliency

and modulated by target-side reliability. The proposed PH proxy loss is defined in Eq. 4.

$$\mathcal{L}_{\mathrm{PHP}} = \sum_{m \in \mathcal{K}} w_m \, \Phi_m(p, \tilde{y}; r_{\tilde{y}}) \,, \tag{4}$$

where $\mathcal{K}$ denotes the saliency-ranked top-k subset of threshold intervals, w_m denotes the normalized event weight, $r_{\tilde{y}} \in [0,1]$ denotes a sample-level target-side reliability score, and Φ_m denotes the event proxy operator on the m-th threshold interval. The operator Φ_m is defined in Eq. 5.

$$\Phi_m = \rho\Big(\Big[b_p^{(m)} - b_{\tilde{y}}^{(m)}\Big]_+\Big) + \rho\Big(\Big[d_p^{(m)} - d_{\tilde{y}}^{(m)}\Big]_+\Big) + \gamma(r_{\tilde{y}})\Big(\rho\Big(b_p^{(m)}\Big) + \rho\Big(d_p^{(m)}\Big)\Big) + \Psi_m(p, \tilde{y}), \tag{5}$$

where $\rho(\cdot)$ denotes a smooth robust penalty, $\gamma(\cdot)$ denotes a reliability-aware fallback coefficient that strengthens as target-side reliability decreases, and Ψ_m denotes the spatial anchoring term. The first two terms penalize prediction-side excess relative to target-side birth-like and death-like activities, the third penalizes prediction-side topological activity more strongly when the target-side structure is unreliable, and the last encourages interval-local spatial consistency. Thus, when $b_p^{(m)} \le b_{\tilde{y}}^{(m)}$ and $d_p^{(m)} \le d_{\tilde{y}}^{(m)}$, the target-referenced exceed terms vanish rather than enforcing equality with the noisy reference. Formal properties and implementation details are given in Appendix B.3, Appendix B.4, and Appendix A.

Since not all topological activities should contribute equally under structurally unreliable supervision [23], we further introduce event-selective weighting. Let s_m denote the saliency score of the m-th threshold interval, defined by the magnitude of its threshold-wise topological variation. The corresponding weighting rule is defined in Eq. 6.

$$w_m = \frac{\exp(s_m/T)\,\mathbf{1}[m \in \mathcal{K}]}{\sum_j \exp(s_j/T)\,\mathbf{1}[j \in \mathcal{K}]}, \tag{6}$$

where T is a temperature parameter. A concrete implementation of s_m is given in Appendix A.4.

To encourage spatial specificity without imposing explicit pairing of persistent features, we anchor the coarse spatial distribution of interval-local support through directional moments. For each threshold interval $[\tau_m, \tau_{m+1}]$, the band response is $B_m(u) = S(u, \tau_m) - S(u, \tau_{m+1})$. The normalized first-order directional moment is defined in Eq. 7.

$$\mu_{m,d}(u) = \frac{\sum_\xi B_m(u;\xi)\, h_d(\xi)}{\sum_\xi B_m(u;\xi) + \varepsilon}, \tag{7}$$

where d is the direction index, ξ is the pixel location, and $\varepsilon > 0$ is a numerical stabilizer. The resulting spatial anchoring term is defined in Eq. 8.

$$\Psi_m(p, \tilde{y}) = \frac{1}{D} \sum_{d=1}^{D} |\mu_{m,d}(p) - \mu_{m,d}(\tilde{y})|^q \,, \tag{8}$$

where $q \ge 1$ is the moment-discrepancy exponent and D is the number of directional fields. Eq. 8 encourages spatial-distribution consistency of interval-local band support between the prediction and the target side. A concrete implementation of the corresponding attenuation is given in Appendix A.5. Overall, Eq. 4 formulates topology supervision as a reliability-aware event-level proxy, rather than as direct fitting to potentially distorted target topology.

3.2 Conservative Topology-Consistency Injection

Because the pixel and topology objectives may induce gradients that differ in both scale and local update direction, naive additive introduction of $\mathcal{L}_{\mathrm{PHP}}$ is poorly matched to noisy teacher–student optimization. We therefore introduce a conservative topology-consistency injection mechanism that regulates topology-side correction through controlled entry, scale moderation, and conflict-aware update direction.

Let $\mathcal{L}_{\mathrm{pix}}$ denote the pixel objective and $\mathcal{L}_{\mathrm{PHP}}$ denote the proposed PH proxy loss. The nominal topology-augmented training objective is defined in Eq. 9.

$$\mathcal{L} = \mathcal{L}_{\mathrm{pix}} + \lambda_{\mathrm{topo}}(t)\mathcal{L}_{\mathrm{PHP}}, \tag{9}$$

where $t \in [0,1]$ denotes the normalized training progress, $\lambda_{\text{topo}}(t) = \lambda_{\max} s(t)$ denotes the topology weight, $\lambda_{\max} > 0$ denotes its maximum value, and $s(t) \in [0,1]$ denotes a monotonically increasing warm-up schedule. Eq. 9 specifies the nominal objective, while the actual topology contribution is further constrained at the gradient level by warm-up scheduling, gradient balancing, conflict-aware projection, and relative-cap control. A concrete implementation of $s(t)$ is given in Appendix A.6.

To control the effective magnitude of topology supervision, let $g_{\text{pix}} = \nabla_z \mathcal{L}_{\text{pix}}$ and $g_{\text{topo}} = \nabla_z \mathcal{L}_{\text{PHP}}$ denote the gradients of the pixel objective and the PH proxy loss with respect to the student logits z. Since the two gradients typically operate on different scales, the gradient-balancing coefficient is defined in Eq. 10.

$$\alpha_t = \left(\frac{\|g_{\text{pix}}\|}{\|g_{\text{topo}}\| + \varepsilon} \right)^{\frac{1}{2}}, \tag{10}$$

where $\varepsilon > 0$ is a numerical stability constant. Eq. 10 rescales the topology branch relative to the pixel branch, so that topology supervision is moderated by the current optimization state rather than injected at a fixed nominal strength. The square-root scaling avoids exact norm matching and yields a milder normalization of topology updates relative to the current pixel-gradient scale.

Magnitude control alone is insufficient, because the pixel and topology objectives need not be locally aligned. When $\langle g_{\text{topo}}, g_{\text{pix}} \rangle < 0$, we treat the two gradients as conflicting and project g_{topo} onto the orthogonal complement of g_{pix}. The orthogonalized topology gradient is defined in Eq. 11.

$$g_{\text{topo}}^{\perp} = g_{\text{topo}} - \frac{\langle g_{\text{topo}}, g_{\text{pix}} \rangle}{\|g_{\text{pix}}\|^2} g_{\text{pix}}, \tag{11}$$

Eq. 11 is written for a single sample and the nondegenerate case $\|g_{\text{pix}}\| > 0$; the safeguarded sample-wise implementation is given in Appendix A.6. The injected topological gradient is then defined in Eq. 12.

$$g_{\text{topo}}^{\text{inj}} = \begin{cases} g_{\text{topo}}^{\perp}, & \langle g_{\text{topo}}, g_{\text{pix}} \rangle < 0, \\ g_{\text{topo}}, & \text{otherwise.} \end{cases} \tag{12}$$

Eq. 12 preserves the original topology gradient in the non-conflicting case and uses the orthogonalized correction only when the topology and pixel gradients are locally misaligned, yielding $\langle g_{\text{topo}}^{\text{inj}}, g_{\text{pix}} \rangle \geq 0$. The corresponding geometric property is given in Appendix B.5.

A relative cap is applied to the injected topological gradient in Eq. 13.

$$\bar{g}_{\text{topo}}^{\text{inj}} = \min\left(1, \frac{c_{\text{cap}} \|g_{\text{pix}}\|}{\|g_{\text{topo}}^{\text{inj}}\| + \varepsilon}\right) g_{\text{topo}}^{\text{inj}}, \tag{13}$$

where $c_{\text{cap}} > 0$ denotes a relative cap ratio. Eq. 13 bounds the pre-scaled injected topology vector as $\|\bar{g}_{\text{topo}}^{\text{inj}}\| \leq c_{\text{cap}} \|g_{\text{pix}}\|$. The effective backward direction on the student logits is defined in Eq. 14.

$$g = g_{\text{pix}} + \lambda_{\text{topo}}(t) \alpha_t \bar{g}_{\text{topo}}^{\text{inj}}, \tag{14}$$

Eq. 14 shows that $\mathcal{L}_{\text{PHP}}$ is injected as a regulated topology-side correction rather than an unconstrained additive loss. Its activation, magnitude, and direction are controlled by the current optimization state, reducing the risk that topology gradients dominate the pixel objective or reinforce distorted pseudo-label structure. Eqs. 10–14 describe a generic topology gradient; in the reported implementation, these controls are applied to each retained event-level gradient before summation. Formal properties and implementation details are given in Appendix B.6, Appendix B.7, and Appendix A.6.

4 Experiments

4.1 Experimental Settings

Datasets. Following the standard UCOD protocol adopted by UCOD-DPL [18], training is performed on an unlabeled set constructed from CAMO [41] and COD10K [42], containing 1,000 and 3,040 training images, respectively. No ground-truth labels are used during training. Evaluation is conducted on four COD benchmarks, namely CAMO [41], CHAMELEON [43], COD10K [42], and NC4K [44], containing 250, 76, 2,026, and 4,121 testing images, respectively. To evaluate transfer beyond UCOD, we additionally use three histology segmentation datasets, CRAG [45], GlaS [46], and MoNuSeg [47]. The matched cross-task protocol and results are reported in Appendix D.3.

Implementation Details. All controlled comparisons are built on the UCOD-DPL carrier [18], whose dynamic pseudo-label learning paradigm matches the noisy pseudo-supervision setting studied in this paper while minimizing confounding changes in the underlying learning framework. All models are trained for 25 epochs with a batch size of 16, using an input resolution of 296×296 and an initial learning rate of 6×10^{-4}. DINOv1 and DINOv2 denote self-supervised Vision Transformer backbones pretrained with DINO and DINOv2, respectively. The model is implemented in PyTorch and trained on a workstation equipped with an NVIDIA GeForce RTX 5090D. For the topology branch, we use 16 thresholds, retain up to 12 highest-saliency intervals, and use four directional moments for interval-local spatial anchoring. Further run-specific configuration details are provided in Appendix C.

Evaluation Metrics. We evaluate performance from both pixel-level and topology-aware perspectives. At the pixel level, we report four standard COD metrics, namely mean absolute error (MAE), structure measure (S_α), enhanced-alignment measure (E_ϕ), and weighted F-measure (F_β^w). For topology-aware evaluation, we report the average counting errors of the zeroth and first Betti numbers, denoted by β_0 and β_1, respectively. Here, β_0 measures the discrepancy in connected components and β_1 measures the discrepancy in holes. We retain F_β^w as the reference COD metric when discussing the relation between structural quality and task-level performance. We additionally report Betti Matching Error (BME) in the controlled topology comparison as a complementary feature-wise topology metric; its protocol is provided in Appendix C.3. Formal definitions and implementation details of the topology-aware metrics are provided in Appendix C.2.

Compared Methods. The comparison protocol includes pixel-level and topology-level evaluations. For pixel-level comparison, we include representative unsupervised baselines and UCOD methods, including FOUND [48], VoteCut [49], UCOS-DA [50], RISE [20], EASE [51], and UCOD-DPL [18], and report literature results together with reproduced UCOD-DPL baselines and our method within each backbone family. For topology-level comparison, we fix the reproduced UCOD-DPL carrier and vary only the topology-supervision formulation, including TopoLoss [2], TopoSemiSeg [23], MATCH [24], and SATLoss [40], under the same backbone, pseudo-label pipeline, optimizer, resolution, training length, and evaluation protocol.

Table 1: Pixel-level comparison on four COD benchmarks. Methods are grouped by backbone. Rows marked with * denote reproduced UCOD-DPL baselines under the unified protocol. Gray rows indicate the designated controlled reference pair. The best and second-best results in each column are **bolded** and underlined, respectively.

Method	Pub./Year	CAMO				CHAMELEON				COD10K				NC4K			
		MAE ↓	S_α ↑	E_ϕ ↑	F_β^w ↑	MAE ↓	S_α ↑	E_ϕ ↑	F_β^w ↑	MAE ↓	S_α ↑	E_ϕ ↑	F_β^w ↑	MAE ↓	S_α ↑	E_ϕ ↑	F_β^w ↑
DINOv1-based methods and variants																	
FOUND [48]	$CVPR_{23}$	0.277	0.526	0.578	0.397	0.218	0.552	0.602	0.381	0.248	0.488	0.487	0.264	0.218	0.569	0.605	0.410
UCOS-DA [50]	$ICCVW_{23}$	0.127	0.701	0.784	0.606	0.095	0.715	0.802	0.591	0.086	0.689	0.740	0.513	0.085	0.755	0.819	0.656
RISE [20]	$ICCV_{25}$	0.132	0.664	0.715	0.507	0.090	0.679	0.741	0.508	0.062	0.724	0.795	0.526	0.081	0.759	0.818	0.633
EASE [51]	$CVPR_{25}$	0.166	0.653	0.737	0.563	0.105	0.676	0.765	0.550	0.109	0.673	0.732	0.514	0.108	0.728	0.790	0.633
UCOD-DPL* [18]	$CVPR_{25}$	0.1023	0.7347	0.8216	0.6508	0.0706	0.7456	0.8586	0.6340	0.0664	0.7192	0.7929	0.5590	0.0685	0.7774	0.8571	0.6925
Ours	–	**0.0972**	**0.7382**	**0.8313**	**0.6673**	**0.0668**	**0.7465**	**0.8637**	**0.6470**	**0.0537**	**0.7476**	**0.8434**	**0.6106**	**0.0639**	**0.7874**	**0.8750**	**0.7163**
DINOv2-based methods and variants																	
FOUND [48]	$CVPR_{23}$	0.172	0.558	0.622	0.373	0.104	0.659	0.745	0.490	0.100	0.605	0.675	0.386	0.131	0.621	0.693	0.455
UCOS-DA [50]	$ICCVW_{23}$	0.148	0.702	0.751	0.604	0.091	0.750	0.808	0.639	0.120	0.655	0.687	0.467	0.103	0.731	0.785	0.617
VoteCut [49]	$CVPR_{24}$	0.170	0.637	0.702	0.522	0.145	0.674	0.735	0.546	0.092	0.690	0.782	0.534	0.097	0.739	0.818	0.649
RISE [20]	$ICCV_{25}$	0.109	0.734	0.787	0.610	0.050	**0.822**	0.884	0.720	0.049	0.763	0.840	0.600	0.061	0.805	0.868	0.705
EASE [51]	$CVPR_{25}$	0.098	0.749	0.831	0.684	0.044	0.819	0.899	0.741	**0.040**	**0.773**	**0.866**	**0.656**	0.056	0.800	0.884	0.735
UCOD-DPL* [18]	$CVPR_{25}$	0.0863	0.7653	0.8506	0.6925	0.0462	0.8061	0.8988	0.7222	0.0524	0.7404	0.8220	0.6058	0.0590	0.7873	0.8706	0.7129
Ours	–	**0.0797**	**0.7821**	**0.8633**	**0.7178**	**0.0415**	0.8216	**0.9165**	**0.7438**	0.0459	0.7609	0.8468	0.6362	**0.0510**	**0.8077**	**0.8900**	**0.7408**

4.2 Comparison Experiments

Table 1 presents the pixel-level comparison under two complementary comparison regimes. The first consists of literature-reported unsupervised COD and UCOD methods, which provide a broad performance reference but are not strictly controlled under a unified training recipe. The second is the designated controlled reference pair under the unified pseudo-label learning protocol, namely the reproduced UCOD-DPL baseline and our method within each backbone family. Under this controlled comparison, our method consistently improves over the reproduced UCOD-DPL baseline under both backbones, with the larger average gains observed on the designated DINOv2 reference pair. Averaged over the four benchmarks, under the unified DINOv2 protocol our method reduces MAE from 0.0610 to 0.0545 and improves F_β^w from 0.6834 to 0.7097. Similar gains are also observed under DINOv1, where MAE is reduced by 0.0066 and F_β^w is improved by 0.0262 over the reproduced

baseline. Relative to literature-reported methods, the proposed method remains competitive overall, while the most relevant evidence comes from the controlled UCOD-DPL reference pairs, where the carrier, pseudo-label learning protocol, and evaluation setting are kept fixed. These findings establish task-level effectiveness under the noisy pseudo-label learning setting most relevant to our problem, but do not by themselves establish whether structural reliability is improved. A ranked diagnostic analysis of difficult COD10K samples is provided in Appendix D.6.

Table 2: Topology-level comparison on four COD benchmarks under the same carrier, training protocol, and pseudo-label pipeline. Methods differ only in the adopted topology supervision. BME provides a complementary feature-wise topology measure under a fixed method-independent protocol. The best and second-best results in each column are **bolded** and underlined, respectively.

Topology loss	Pub./Year	CAMO				CHAMELEON				COD10K				NC4K			
		$F_\beta^w \uparrow$	$\beta_0 \downarrow$	$\beta_1 \downarrow$	BME $\downarrow$	$F_\beta^w \uparrow$	$\beta_0 \downarrow$	$\beta_1 \downarrow$	BME $\downarrow$	$F_\beta^w \uparrow$	$\beta_0 \downarrow$	$\beta_1 \downarrow$	BME $\downarrow$	$F_\beta^w \uparrow$	$\beta_0 \downarrow$	$\beta_1 \downarrow$	BME $\downarrow$
No topology	–	0.6925	9.3947	4.3947	–	0.7222	10.564	5.2640	–	0.6058	9.8134	6.2216	–	0.7129	5.5773	3.2609	–
TopoLoss	NeurIPS$_{19}$	0.3860	15.0240	9.1960	24.800	0.3544	24.6842	6.3816	36.158	0.2416	21.4906	10.8885	39.537	0.2564	15.7518	6.9993	27.563
TopoSemiSeg	ECCV$_{24}$	0.4439	0.8600	0.6840	2.184	0.6358	9.7895	**0.9868**	10.645	0.5415	8.9205	5.7152	13.495	0.5880	1.7239	1.2157	5.412
SATLoss	ICCVW$_{25}$	0.3595	0.9960	**0.4440**	2.136	0.5622	9.7763	1.7895	**10.408**	0.4810	8.9072	5.6584	13.376	0.5142	1.6867	**1.0713**	5.181
MATCH	NeurIPS$_{25}$	0.4425	**0.6560**	0.4800	**1.668**	0.5861	9.8421	1.8026	10.447	0.5068	8.9783	5.6372	**13.300**	0.5470	**1.6843**	2.0374	**5.042**
Ours	–	**0.7178**	5.3600	3.2678	9.236	**0.7438**	**9.2368**	1.4211	13.171	**0.6362**	**8.7754**	**5.2142**	17.610	**0.7408**	3.5911	2.2863	9.237

To answer this question directly, Table 2 compares different topology-supervision formulations under the same UCOD-DPL carrier and training protocol. Because the compared topology losses were developed under different domains and supervision assumptions, this experiment should be interpreted as a controlled transfer comparison under the unified UCOD-DPL setting rather than as a claim of universal superiority in their original settings. Among the compared topology losses, our method is the only one that consistently improves F_β^w, β_0, and β_1 over the reproduced no-topology baseline on all four benchmarks. On average, it raises F_β^w from 0.6834 to 0.7097 while reducing β_0 from 8.8374 to 6.7408 and β_1 from 4.7853 to 3.0474, corresponding to relative changes of $+3.85\%$, -23.72%, and -36.32%, respectively. Other topology-supervision formulations can reduce some raw Betti-counting errors, but these reductions do not consistently translate into stronger COD quality and often come with degraded F_β^w in this noisy pseudo-supervision setting. BME further yields a different ranking among the topology-supervision methods, showing that feature-wise topological correspondence and foreground prediction quality are complementary rather than interchangeable. This comparison suggests that, in the present UCOD setting, topology supervision is more effective when trainable topology signals are explicitly derived and conservatively introduced into optimization. Controlled structural-corruption experiments are reported in Appendix D.5, while additional boundary cases are examined in Appendix D.2.

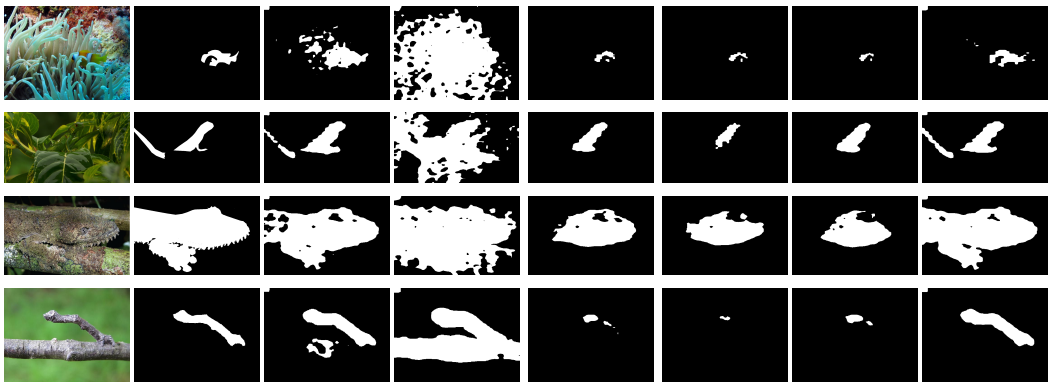


Figure 3: Qualitative comparison under the same UCOD-DPL carrier, showing representative cases of fragmented structure, weak-connection breakage, spurious hole-like artifacts, and over-correction risks in topology-supervised variants.

Figure 3 illustrates four representative structural regimes under the same UCOD-DPL carrier, including fragmented connected components, weak-connection breakage, spurious hole-like artifacts, and over-correction risks in topology-supervised variants. The no-topology baseline often captures coarse object semantics but leaves fragmented foreground structure, weak discontinuities, or spurious

hole-like artifacts unresolved. Compared topology losses can reduce some structural errors, but may also suppress fragile object regions in this noisy pseudo-supervision setting. By contrast, the proposed method more consistently improves structural coherence while preserving the main object region. This qualitative tendency is consistent with Table 2, where our method is the only compared topology loss that improves the reference pixel metric together with both Betti-counting errors. Additional qualitative comparisons are provided in Appendix D.1.

4.3 Ablation Experiments

Unless otherwise stated, all ablation experiments are conducted on NC4K under the unified DINOv2 protocol. NC4K is the largest evaluation set among the four COD benchmarks, which reduces small-sample fluctuation and provides a relatively more stable basis for controlled ablation.

4.3.1 Ablation on Conservative Topology-Consistency Injection

Table 3 examines whether PH-based topology supervision is better introduced through conservative gradient-constrained injection than through direct additive optimization. The carrier and $\mathcal{L}_{\mathrm{PHP}}$ are fixed: Direct PH addition uses the raw topology gradient without balancing, conflict-aware projection, or relative-cap control; Sum-level projection applies these controls after aggregating the retained-event gradients; and Per-term projection applies them separately before summation (Appendix A.6). Direct PH addition severely degrades task-level performance, suggesting that naive additive integration is poorly matched to optimization under structurally unreliable pseudo supervision. By contrast, both conservative-injection variants achieve strong performance, and the per-term realization further improves over the sum-level realization on both pixel-level and topology-level metrics.

Table 3: Ablation on conservative topology-consistency injection.

Injection variant	MAE ↓	S_α ↑	E_ϕ ↑	F_β^w ↑	β_0 ↓	β_1 ↓
No topology	0.0590	0.7873	0.8706	0.7129	5.5773	3.2609
Direct PH addition	0.1500	0.6375	0.6770	0.4956	5.7974	3.0269
Sum-level projection	0.0521	0.8043	0.8762	0.7387	3.7584	2.4765
Per-term projection (default)	0.0510	0.8077	0.8900	0.7408	3.5911	2.2863

4.3.2 Ablation on PH Proxy Formulation

Table 4 compares the PH proxy modes defined in Appendix C.4 under structurally unreliable pseudo supervision. Here, *Match* (distinct from the external MATCH baseline) symmetrically matches prediction- and reference-side proxy activities, *Simplicity* penalizes prediction-side activity alone, and *Cap* uses one-sided exceed control with prediction-side fallback. The default Cap formulation achieves the strongest task-level result, outperforming both *Match* and *Simplicity* in MAE and F_β^w. Although *Match* yields smaller raw Betti-counting errors, it remains inferior in COD quality. This result suggests that directly reducing Betti-count errors is not sufficient when pseudo supervision is structurally unreliable, because topology-side correction should also avoid suppressing fragile object regions. The Cap formulation therefore favors conservative complexity control over direct target-side matching.

Table 4: Ablation on PH proxy formulation.

PH proxy mode	MAE ↓	S_α ↑	E_ϕ ↑	F_β^w ↑	β_0 ↓	β_1 ↓
Match	0.0560	0.7931	0.8749	0.7200	2.4945	2.1728
Simplicity	0.0595	0.7843	0.8660	0.7077	3.4654	2.2218
Cap (default)	0.0510	0.8077	0.8900	0.7408	3.5911	2.2863

4.3.3 Ablation on Topology-Injection Strength

Table 5 evaluates sensitivity to the topology strength. Small nonzero weights already reduce Betti-counting errors relative to $\lambda_{\mathrm{topo}} = 0$, but yield limited task-level gains. Task-level performance improves as the topology weight increases to the 0.5–2.0 range, whereas stronger weights of 4.0 and 8.0 degrade all four COD metrics despite relatively small Betti-counting errors. We use $\lambda_{\mathrm{topo}} = 1.0$ as the default because the results at 1.0 and 2.0 are nearly identical, and 1.0 is

Table 5: Ablation on topology-injection strength.

λ_{topo}	MAE ↓	S_α ↑	E_ϕ ↑	F_β^w ↑	β_0 ↓	β_1 ↓
0.0	0.0590	0.7873	0.8706	0.7129	5.5773	3.2609
1e-2	0.0589	0.7874	0.8708	0.7131	3.8901	2.3152
5e-2	0.0586	0.7877	0.8710	0.7133	3.7935	2.2960
1e-1	0.0581	0.7887	0.8720	0.7147	3.6884	2.2866
5e-1	0.0519	0.8048	0.8876	0.7371	3.6096	2.2429
1.0 (default)	0.0510	0.8077	0.8900	0.7408	3.5911	2.2863
2.0	0.0510	0.8082	0.8899	0.7409	3.5969	2.3356
4.0	0.0588	0.7876	0.8681	0.7103	3.4681	2.1444
8.0	0.0701	0.7633	0.8361	0.6737	3.7212	2.1456

the smaller setting in this stable range. Sensitivity to the remaining topology hyperparameters is reported in Appendix D.4.

4.3.4 Ablation on Topology-Injection Timing

Table 6 examines when topology supervision should be introduced during training. Delaying topology injection too far into training is consistently weaker, as start epochs 15, 20, and especially 25 reduce both pixel-level and topology-level performance relative to earlier starts. By contrast, start epochs 0, 5, and 10 yield nearly identical task-level COD results. This pattern is consistent with the conservative injection design, under which early topology updates are strongly filtered before they can substantially influence optimization. In particular, the identical results for start epochs 0 and 5 suggest that early nominal activation does not necessarily translate into effective topology-side correction when unreliable topology gradients are suppressed. We adopt start epoch 10 as a conservative default because it delays topology entry beyond the earliest nominally enabled regime while preserving essentially the same task-level behavior. This choice should not be interpreted as a unique empirical optimum on NC4K, since earlier starts yield smaller raw Betti-counting errors in this ablation, but rather as a conservative scheduling preference under the present unified protocol.

Table 6: Ablation on topology-injection timing.

Start epoch	MAE ↓	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$\beta_0 \downarrow$	$\beta_1 \downarrow$
0	0.0510	0.8076	0.8898	0.7406	2.5894	2.2810
5	0.0510	0.8076	0.8898	0.7406	2.5894	2.2810
10 (default)	0.0510	0.8077	0.8900	0.7408	3.5911	2.2863
15	0.0518	0.8051	0.8876	0.7371	3.6339	2.3531
20	0.0536	0.8003	0.8833	0.7308	3.6462	2.3475
25	0.0590	0.7873	0.8706	0.7129	5.5773	3.2609

5 Conclusion

We studied topology supervision under structurally unreliable pseudo supervision, where supervision-side topology may already be distorted and should not be assumed to provide a reliable direct optimization target. We addressed this problem by reformulating topology supervision as trainable topology-side correction through a differentiable PH proxy loss and a conservative topology-consistency injection mechanism. In the teacher–student setting considered here, the teacher supplies the noisy structural reference, while topology-side correction acts on the student. Experiments in UCOD show that this reformulation improves both task-level COD performance and topology-level structural reliability under controlled same-carrier comparisons. These results support the central claim of this work: when supervision-side structure is unreliable, topology supervision should be derived from trainable proxy signals and introduced into optimization conservatively, rather than imposed as direct fitting to potentially distorted pseudo-label topology.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants U24A20333 and 62472443 and 62376287, the Hunan Provincial Natural Science Foundation under Grant 2024JJ3032, the Noncommunicable Chronic Diseases National Science and Technology Major Project under Grants 2023ZD0508200 and 2023ZD0508201, the International Science and Technology Innovation Joint Base of Machine Vision and Medical Image Processing in Hunan Province under Grant 2021CB1013, the Natural Science Foundation of Hunan Province under Grant 2025JJ60421, the Open Research Fund of Hunan Provincial Key Laboratory of Intelligent Processing of Big Data on Transportation under Grant B202405, and The Scientific Research Program of FuRong Laboratory under Grant 2025PT5009.

References

[1] Agata Mosinska, Pablo Marquez-Neila, Mateusz Kozinski, and Pascal Fua. Beyond the pixel-wise loss for topology-aware delineation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3136–3145, 2018.

[2] Xiaoling Hu, Fuxin Li, Dimitris Samaras, and Chao Chen. Topology-preserving deep image segmentation. In *Advances in Neural Information Processing Systems*, volume 32, pages 5657–5668, 2019.

[3] Meng Liu, Tamal K Dey, and David F Gleich. Topological structure of complex predictions. *Nature Machine Intelligence*, 5(12):1382–1389, 2023.

[4] James R. Clough, Nicholas Byrne, Ilkay Oksuz, Veronika A. Zimmer, Julia A. Schnabel, and Andrew P. King. A topological loss function for deep-learning based image segmentation using persistent homology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12): 8766–8778, 2022.

[5] Nick Byrne, James R. Clough, Israel Valverde, Giovanni Montana, and Andrew P. King. A persistent homology-based topological loss for cnn-based multiclass segmentation of cmr. *IEEE Transactions on Medical Imaging*, 42(1):3–14, 2023.

[6] Suprosanna Shit, Johannes C. Paetzold, Anjany Sekuboyina, Ivan Ezhov, Alexander Unger, Andrey Zhylka, Josien P. W. Pluim, Ulrich Bauer, and Bjoern H. Menze. cldice - a novel topology-preserving loss function for tubular structure segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16560–16569, 2021.

[7] Hongliang He, Jun Wang, Pengxu Wei, Fan Xu, Xiangyang Ji, Chang Liu, and Jie Chen. Toposeg: Topology-aware nuclear instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 21307—21316, 2023.

[8] Laurin Lux, Alexander H Berger, Alexander Weers, Nico Stucki, Daniel Rueckert, Ulrich Bauer, and Johannes C Paetzold. Topograph: An efficient graph-based framework for strictly topology preserving image segmentation. In *International Conference on Learning Representations*, 2025.

[9] Katharine Turner, Sayan Mukherjee, and Doug M. Boyer. Persistent homology transform for modeling shapes and surfaces. *Information and Inference: A Journal of the IMA*, 3(4):310–344, 2014.

[10] Ernst Röell and Bastian Rieck. Differentiable euler characteristic transforms for shape classification. In *International Conference on Learning Representations*, 2024.

[11] Luis Scoccola, Siddharth Setlur, David Loiseaux, Mathieu Carrière, and Steve Oudot. Differentiability and optimization of multiparameter persistent homology. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pages 43986–44011, 2024.

[12] Caner Korkmaz, Brighton Nuwagira, Baris Coskunuzer, and Tolga Birdal. Cumperlay: Learning cubical multiparameter persistence vectorizations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 27084–27094, 2025.

[13] Alexander H. Berger, Laurin Lux, Alexander Weers, Martin J. Menten, Daniel Rueckert, and Johannes C. Paetzold. Pitfalls of topology-aware image segmentation. In *Information Processing in Medical Imaging (IPMI)*, volume 15829, pages 297–312, 2026.

[14] Yuchao Wang, Haochen Wang, Yujun Shen, Jingjing Fei, Wei Li, Guoqiang Jin, Liwei Wu, Rui Zhao, and Xinyi Le. Semi-supervised semantic segmentation using unreliable pseudo-labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4248–4257, 2022.

[15] Ying Jin, Jiaqi Wang, and Dahua Lin. Semi-supervised semantic segmentation via gentle teaching assistant. In *Advances in Neural Information Processing Systems*, volume 35, pages 2803–2816, 2022.

[16] Donghyeon Kwon and Suha Kwak. Semi-supervised semantic segmentation with error localization network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9957—9967, 2022.

[17] Boyuan Sun, Yuqi Yang, Le Zhang, Ming-Ming Cheng, and Qibin Hou. Corrmatch: Label propagation via correlation matching for semi-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3097–3107, 2024.

[18] Weiqi Yan, Lvhai Chen, Huaijia Kou, Shengchuan Zhang, Yan Zhang, and Liujuan Cao. Ucod-dpl: Unsupervised camouflaged object detection via dynamic pseudo-label learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 30365–30375, 2025.

[19] Jin Zhang, Ruiheng Zhang, Yanjiao Shi, Zhe Cao, Nian Liu, and Fahad Shahbaz Khan. Learning camouflaged object detection from noisy pseudo label. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 158–174, 2024.

[20] Ji Du, Xin Wang, Fangwei Hao, Mingyang Yu, Chunyuan Chen, Jiesheng Wu, Bin Wang, Jing Xu, and Ping Li. Beyond single images: Retrieval self-augmented unsupervised camouflaged object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22131–22142, 2025.

[21] Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. In *International Conference on Learning Representations*, 2021.

[22] Saumya Gupta, Yikai Zhang, Xiaoling Hu, Prateek Prasanna, and Chao Chen. Topology-aware uncertainty for image segmentation. In *Advances in Neural Information Processing Systems*, volume 36, pages 8186–8207, 2023.

[23] Meilong Xu, Xiaoling Hu, Saumya Gupta, Shahira Abousamra, and Chao Chen. Semi-supervised segmentation of histopathology images with noise-aware topological consistency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 15136, pages 271–289, 2024.

[24] Meilong Xu, Xiaoling Hu, Shahira Abousamra, Chen Li, and Chao Chen. Match: Multi-faceted adaptive topo-consistency for semi-supervised histopathology segmentation. In *Advances in Neural Information Processing Systems*, 2025.

[25] Pan Liu and Jinshi Liu. When confidence fails: Revisiting pseudo-label selection in semi-supervised semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 21874–21884, 2025.

[26] Xiaokang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang. Semi-supervised semantic segmentation with cross pseudo supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2613–2622, 2021.

[27] Yassine Ouali, Céline Hudelot, and Myriam Tami. Semi-supervised semantic segmentation with cross-consistency training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12674—-12684, 2020.

[28] Lihe Yang, Lei Qi, Litong Feng, Wayne Zhang, and Yinghuan Shi. Revisiting weak-to-strong consistency in semi-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7236–7246, 2023.

[29] Huayu Mai, Rui Sun, Tianzhu Zhang, and Feng Wu. Rankmatch: Exploring the better consistency regularization for semi-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3391–3401, 2024.

[30] Ruifei He, Jihan Yang, and Xiaojuan Qi. Re-distributing biased pseudo labels for semi-supervised semantic segmentation: A baseline investigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6930–6940, 2021.

[31] Hanzhe Hu, Fangyun Wei, Han Hu, Qiwei Ye, Jinshi Cui, and Liwei Wang. Semi-supervised semantic segmentation via adaptive equalization learning. In *Advances in Neural Information Processing Systems*, volume 34, pages 22106–22118, 2021.

[32] Jie Ma, Chuan Wang, Yang Liu, Liang Lin, and Guanbin Li. Enhanced soft label for semi-supervised semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1185–1195, 2023.

[33] Xin Lai, Zhuotao Tian, Li Jiang, Shu Liu, Hengshuang Zhao, Liwei Wang, and Jiaya Jia. Semi-supervised semantic segmentation with directional context-aware consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1205–1214, 2021.

[34] Pengchong Qiao, Zhidan Wei, Yu Wang, Zhennan Wang, Guoli Song, Fan Xu, Xiangyang Ji, Chang Liu, and Jie Chen. Fuzzy positive learning for semi-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15465–15474, 2023.

[35] Udit Saxena. Scalable gpu-accelerated euler characteristic curves: Optimization and differentiable learning for pytorch, 2025.

[36] Xiaoling Hu, Yusu Wang, Li Fuxin, Dimitris Samaras, and Chao Chen. Topology-aware segmentation using discrete morse theory. In *International Conference on Learning Representations*, 2021.

[37] Nico Stucki, Johannes C. Paetzold, Suprosanna Shit, Bjoern Menze, and Ulrich Bauer. Topologically faithful image segmentation via induced matching of persistence barcodes. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202, pages 32698–32727, 2023.

[38] Yaolei Qi, Yuting He, Xiaoming Qi, Yuan Zhang, and Guanyu Yang. Dynamic snake convolution based on topological geometric constraints for tubular structure segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6070—-6079, 2023.

[39] Andac Demir, Elie Massaad, and Bulent Kiziltan. Topology-aware focal loss for 3d image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 580–589, 2023.

[40] Bo Wen, Haochen Zhang, Dirk-Uwe G. Bartsch, William R. Freeman, Truong Q. Nguyen, and Cheolhong An. Topology-preserving image segmentation with spatial-aware persistent feature matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 5821—-5830, 2025.

[41] Trung-Nghia Le, Tam V. Nguyen, Zhongliang Nie, Minh-Triet Tran, and Akihiro Sugimoto. Anabranch network for camouflaged object segmentation. *Computer Vision and Image Understanding*, 184:45–56, 2019.

[42] Deng-Ping Fan, Ge-Peng Ji, Guolei Sun, Ming-Ming Cheng, Jianbing Shen, and Ling Shao. Camouflaged object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2777–2787, 2020.

[43] Przemysław Skurowski, Hassan Abdulameer, Jakub Błaszczyk, Tomasz Depta, Adam Kornacki, and Przemysław Kozieł. Animal camouflage analysis: Chameleon database. *Unpublished manuscript*, 2(6):7, 2018.

[44] Yunqiu Lv, Jing Zhang, Yuchao Dai, Aixuan Li, Bowen Liu, Nick Barnes, and Deng-Ping Fan. Simultaneously localize, segment and rank the camouflaged objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11591–11601, 2021.

[45] Simon Graham, Hao Chen, Jevgenij Gamper, Qi Dou, Pheng-Ann Heng, David Snead, Yee Wah Tsang, and Nasir Rajpoot. Mild-net: Minimal information loss dilated network for gland instance segmentation in colon histology images. *Medical Image Analysis*, 52:199–211, 2019.

[46] Korsuk Sirinukunwattana, Josien P. W. Pluim, Hao Chen, Xiaojuan Qi, Pheng-Ann Heng, Yun Bo Guo, Li Yang Wang, Bogdan J. Matuszewski, Elia Bruni, Urko Sanchez, Anton Böhm, Olaf Ronneberger, Bassem Ben Cheikh, Daniel Racoceanu, Philipp Kainz, Michael Pfeiffer, Martin Urschler, David R. J. Snead, and Nasir M. Rajpoot. Gland segmentation in colon histology images: The glas challenge contest. *Medical Image Analysis*, 35:489–502, 2017.

[47] Neeraj Kumar, Ruchika Verma, Deepak Anand, Yanning Zhou, Omer Fahri Onder, Efstratios Tsougenis, Hao Chen, Pheng-Ann Heng, Jiahui Li, Zhiqiang Hu, et al. A multi-organ nucleus segmentation challenge. *IEEE Transactions on Medical Imaging*, 39(5):1380–1391, 2020.

[48] Oriane Siméoni, Chloé Sekkat, Gilles Puy, Antonín Vobecký, Éloi Zablocki, and Patrick Pérez. Unsupervised object localization: Observing the background to discover objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3176–3186, 2023.

[49] Shahaf Arica, Or Rubin, Sapir Gershov, and Shlomi Laufer. Cuvler: Enhanced unsupervised object discoveries through exhaustive self-supervised transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23105–23114, 2024.

[50] Yi Zhang and Chengyi Wu. Unsupervised camouflaged object segmentation as domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 4334–4344, 2023.

[51] Ji Du, Fangwei Hao, Mingyang Yu, Desheng Kong, Jiesheng Wu, Bin Wang, Jing Xu, and Ping Li. Shift the lens: Environment-aware unsupervised camouflaged object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19271–19282, 2025.

A Implementation Details of the Proposed Method

This section specifies the implementation-level instantiations of the operators introduced in the main text. Its purpose is to make explicit how the abstract formulation in Sec. 3 is instantiated in the reported implementation, while leaving run-specific hyperparameter choices to Appendix C. Throughout this section, we preserve the notation used in the main text and only state the concrete functional forms required to instantiate the PH proxy branch and the conservative topology-consistency injection mechanism.

A.1 Differentiable 4-Neighborhood Relaxation for the Soft Euler Trajectory

The main text defines the soft threshold response $S(u, \tau_m)$ in Eq. 1 and the corresponding soft Euler trajectory $\chi_u(\tau_m)$ in Eq. 2. In implementation, the soft Euler trajectory is realized through a differentiable 4-neighborhood relaxation on the image grid.

Let $A = S(u, \tau_m) \in [0,1]^{H \times W}$ denote the soft threshold response at threshold τ_m. The soft vertex, edge, and face counts are instantiated in Eq. 15.

$$\begin{aligned} V(A) &= \sum_{i,j} A_{i,j}, \\ E(A) &= \sum_{i,j} A_{i,j} A_{i+1,j} + \sum_{i,j} A_{i,j} A_{i,j+1}, \\ F(A) &= \sum_{i,j} A_{i,j} A_{i+1,j} A_{i,j+1} A_{i+1,j+1}, \end{aligned} \tag{15}$$

where the summation ranges are restricted to valid grid indices. The corresponding soft Euler trajectory is then computed in Eq. 16.

$$\chi_u(\tau_m) = V(A) - E(A) + F(A). \tag{16}$$

Given the discrete trajectory $\{\chi_u(\tau_m)\}_{m=1}^{M}$, the adjacent-threshold topological variation is computed in Eq. 17.

$$\Delta\chi_u^{(m)} = \chi_u(\tau_{m+1}) - \chi_u(\tau_m), \qquad m = 1, \ldots, M-1, \tag{17}$$

The corresponding birth-type and death-type activities used in Eq. 4 are instantiated in Eq. 18.

$$b_u^{(m)} = [\Delta\chi_u^{(m)}]_+, \qquad d_u^{(m)} = [-\Delta\chi_u^{(m)}]_+. \tag{18}$$

In implementation, the PH proxy branch is evaluated on a resized probability field $\bar{u} = \mathcal{R}(u)$, where $\mathcal{R}(\cdot)$ denotes the resizing operator applied before topology evaluation. This resizing affects only the computational realization of the topology branch and does not alter the analytical form of the operators in the main text.

A.2 Target-side Reliability Estimation

The main text introduces $r_{\tilde{y}} \in [0,1]$ as a sample-level target-side reliability score in Eq. 4 and Eq. 5. In implementation, this score is derived from target-side uncertainty rather than from a direct mean-confidence statistic.

For the downsampled target-side probability field $p_t = \mathcal{R}(\tilde{y})$, the target-side uncertainty is estimated in Eq. 19.

$$u_t = \frac{1}{HW} \sum_{i,j} p_t(i,j)\big(1 - p_t(i,j)\big), \tag{19}$$

The corresponding reliability score is instantiated in Eq. 20.

$$r_{\tilde{y}} = \mathrm{clip}\Big(1 - \frac{u_t}{0.25},\, 0,\, 1\Big). \tag{20}$$

Since 0.25 is the maximum variance of a Bernoulli random variable, Eq. 20 maps maximal uncertainty to zero reliability and near-deterministic predictions to high reliability.

An analogous prediction-side confidence is defined by applying the same variance-to-confidence transformation to the student probability field. This quantity is used only in implementation-level modulation rules within Appendix A.4 and Appendix A.5, and does not alter the main-text definition of $\mathcal{L}_{\mathrm{PHP}}$.

A.3 Robust Penalty and Reliability-aware Fallback Instantiation

The penalty $\rho(\cdot)$ appearing in Eq. 5 is instantiated in Eq. 21.

$$\rho(a) = \sqrt{a^2 + \varepsilon_\rho^2} - \varepsilon_\rho, \tag{21}$$

where $\varepsilon_\rho > 0$ is a numerical stabilizer. Eq. 21 defines a centered smooth robust penalty satisfying $\rho(a) \geq 0$, $\rho(0) = 0$, and monotonicity on $[0, \infty)$.

At the implementation level, the analytical operator in Eq. 5 is realized as the sum of an exceed-only branch, a reliability-aware fallback branch, and an implementation-level spatial branch. The exceed-only branch is given in Eq. 22.

$$\Phi_m^{\mathrm{exc}} = \rho\Big([b_p^{(m)} - b_{\tilde{y}}^{(m)}]_+\Big) + \rho\Big([d_p^{(m)} - d_{\tilde{y}}^{(m)}]_+\Big), \tag{22}$$

whereas the fallback branch is given in Eq. 23.

$$\Phi_m^{\mathrm{fb}} = \eta(r_{\tilde{y}})\Big(\rho(b_p^{(m)}) + \rho(d_p^{(m)})\Big), \tag{23}$$

where $\eta(r_{\tilde{y}}) \geq 0$ denotes the implementation-level realization of the analytical fallback coefficient $\gamma(r_{\tilde{y}})$ in Eq. 5. In the reported implementation, $\eta(\cdot)$ is chosen to be monotone with respect to target-side uncertainty, so that the fallback branch strengthens as $r_{\tilde{y}}$ decreases.

The implementation-level proxy operator is then written in Eq. 24.

$$\Phi_m^{\mathrm{impl}} = \Phi_m^{\mathrm{exc}} + \Phi_m^{\mathrm{fb}} + \Psi_m^{\mathrm{impl}}(p, \tilde{y}), \tag{24}$$

where the implementation-level spatial branch is an attenuation of the analytical spatial anchoring term in Eq. 8. This attenuation is defined explicitly in Eq. 29, which is given later in Appendix A.5. Therefore, Eq. 24 should be interpreted as the implementation-level counterpart of Eq. 5, rather than as a redefinition of the analytical operator itself.

Any implementation-level coefficient used by the fallback branch is fixed by the reported configuration or by the reliability quantities defined above. Such coefficients affect only the weighting of Eq. 23 and do not alter the decomposition in Eq. 24.

A.4 Event-weighting Instantiations

The analytical event-weighting rule is stated in Eq. 6. For the reported implementation, the saliency score is instantiated by the fixed auto-switching rule in Eq. 25.

$$c_m = \begin{cases} \left|\Delta\chi_{\tilde{y}}^{(m)}\right|, & r_{\tilde{y}} \geq \tau_{\mathrm{auto}}, \\ r_{\tilde{y}} \left|\Delta\chi_{\tilde{y}}^{(m)}\right| + (1 - r_{\tilde{y}}) \left|\Delta\chi_p^{(m)}\right|, & r_{\tilde{y}} < \tau_{\mathrm{auto}}, \end{cases} \tag{25}$$

where τ_{auto} denotes the reliability threshold separating the two regimes. The raw saliency scores are normalized according to Eq. 26.

$$\hat{c}_m = \frac{c_m}{\frac{1}{M-1}\sum_j c_j + \varepsilon}, \tag{26}$$

where $\varepsilon > 0$ is a numerical stabilizer. The final implementation-level weights are then computed by the masked softmax rule in Eq. 27.

$$w_m = \frac{\exp(\hat{c}_m / T_{\mathrm{eff}})\mathbf{1}[m \in \mathcal{K}]}{\sum_j \exp(\hat{c}_j / T_{\mathrm{eff}})\mathbf{1}[j \in \mathcal{K}]}, \tag{27}$$

where $T_{\mathrm{eff}} > 0$ denotes the effective temperature and $\mathcal{K}$ denotes the retained top-k event set. Eq. 27 preserves the same normalized weighting structure as Eq. 6; only the saliency instantiation and effective temperature are implementation-level choices. Unless explicitly stated otherwise, additional internal saliency variants are not part of the reported method.

In the reported implementation, T_{eff} is kept fixed across all reported runs. It affects only the sharpness of the masked softmax in Eq. 27 and does not introduce additional saliency branches, learned parameters, or dataset-specific event-selection rules.

A.5 Spatial Anchoring Instantiations

The analytical spatial anchoring term $\Psi_m(p, \tilde{y})$ is defined in Eq. 8. In implementation, this term is instantiated through fixed directional height fields on normalized image coordinates.

Let $(x, y) \in [-1, 1]^2$ denote normalized image coordinates. For D directions, the directional height fields are defined in Eq. 28.

$$h_d(x, y) = \cos(\theta_d)x + \sin(\theta_d)y, \qquad \theta_d = \frac{(d-1)\pi}{D}, \quad d = 1, \ldots, D, \tag{28}$$

Thus, $\{\theta_d\}_{d=1}^{D}$ is evenly spaced on the half-open interval $[0, \pi)$, avoiding duplicated antipodal height fields. Eq. 28 provides the directional basis used to evaluate the moment map in Eq. 7. Using Eq. 7 and Eq. 8, the implementation-level spatial branch is written as the attenuation in Eq. 29.

$$\Psi_m^{\mathrm{impl}}(p, \tilde{y}) = a_{\mathrm{sp}}\, \Psi_m(p, \tilde{y}), \tag{29}$$

where $a_{\mathrm{sp}} \in [0, 1]$ denotes the effective implementation-level attenuation coefficient.

Eq. 29 should be interpreted as an implementation-level weighting of the analytical term in Eq. 8, not as a modification of its analytical definition. In the reported implementation, a_{sp} is a deterministic attenuation coefficient kept fixed by the same implementation rule across all reported runs. It only controls the effective strength of the spatial branch and does not alter the analytical role of Ψ_m in Eq. 5.

A.6 Implementation-level Realization of Conservative Injection

The main text defines conservative topology-consistency injection through the topology-entry schedule in Eq. 9, the gradient-balancing coefficient in Eq. 10, the conflict-aware projection in Eq. 11 and Eq. 12, the relative cap in Eq. 13, and the effective backward direction in Eq. 14. This subsection gives the corresponding implementation-level forms.

Projection safeguard and runtime handling. The analytical projection in Eq. 11 is applied only to samples whose pixel-gradient norm is numerically stable. In implementation, let $\epsilon_{\mathrm{pix}} > 0$ denote the projection safeguard threshold and define the sample-wise indicator $\mathbf{1}_{\mathrm{stab},i} = \mathbf{1}[\|g_{\mathrm{pix},i}\| > \epsilon_{\mathrm{pix}}]$. For samples with $\mathbf{1}_{\mathrm{stab},i} = 0$, the projection step is skipped and the raw topology gradient is retained. For samples with $\mathbf{1}_{\mathrm{stab},i} = 1$, conflict-aware projection is applied only when $\langle g_{\mathrm{topo},i}, g_{\mathrm{pix},i} \rangle < 0$; otherwise the original topology gradient is kept. This projection step is used as a standard local conflict-removal operation within the broader conservative injection mechanism, rather than as a standalone optimization algorithm. The projection denominator is implemented using the safeguarded pixel-gradient norm $\|g_{\mathrm{pix},i}\|_{\mathrm{safe}} = \max(\|g_{\mathrm{pix},i}\|, \epsilon_{\mathrm{pix}})$. In the relative cap, the same safeguarded pixel-gradient norm is used only to define the allowed topology-gradient scale, while the cap denominator is stabilized by $\|g_{\mathrm{topo},i}^{\mathrm{inj}}\| + \varepsilon$. This runtime rule is the implementation-level counterpart of Assumption B.1 and prevents degenerate division in Eq. 11 without altering the non-conflicting branch.

Warm-up schedule. The outer topology-entry schedule $s(t)$ appearing in $\lambda_{\mathrm{topo}}(t) = \lambda_{\max} s(t)$ is instantiated in Eq. 30.

$$s(t) = \begin{cases} 0, & t \le t_0, \\ 1, & t \ge t_1, \\ \frac{1}{2}\left(1 - \cos\left(\pi \frac{t - t_0}{t_1 - t_0}\right)\right), & t_0 < t < t_1, \end{cases} \tag{30}$$

where t_0 and t_1 denote the start and end of topology warm-up, respectively. Eq. 30 is the implementation-level realization of the outer schedule governing the entry of topology supervision into optimization.

In the reported implementation, early-stage attenuation is fully determined by the warm-up schedule in Eq. 30 and the PH-proxy weighting rules specified in Appendix A.4 and Appendix A.5. These deterministic attenuation factors only affect the effective strength of the topology branch during early optimization and do not introduce additional learned modules or adaptive selection procedures.

Relative cap. The implementation-level realization of the relative cap in Eq. 13 is given in Eq. 31.

$$\bar{g}_{\mathrm{topo}}^{\mathrm{inj}} = \min\left(1,\ \frac{c_{\mathrm{cap}}\|g_{\mathrm{pix}}\|_{\mathrm{safe}}}{\|g_{\mathrm{topo}}^{\mathrm{inj}}\| + \varepsilon}\right) g_{\mathrm{topo}}^{\mathrm{inj}}, \tag{31}$$

where $c_{\mathrm{cap}} > 0$ is a relative cap ratio, $\varepsilon > 0$ is a numerical stabilizer, and $\|g_{\mathrm{pix}}\|_{\mathrm{safe}}$ denotes the safeguarded pixel-gradient norm defined above. Eq. 31 is the implementation-level realization of the capped gradient already introduced analytically in Eq. 13.

Backward vector used by the surrogate. The implementation-level surrogate operates on the topology-side backward vector selected after projection and capping, before applying the outer topology-entry schedule. This vector is denoted by g^{use} and is defined in Eq. 32.

$$g^{\mathrm{use}} = \alpha_t \bar{g}_{\mathrm{topo}}^{\mathrm{inj}}, \tag{32}$$

where α_t and $\bar{g}_{\mathrm{topo}}^{\mathrm{inj}}$ are the same quantities appearing in Eq. 14. The raw topology scalar produced by the topology branch before surrogate replacement is denoted by $\mathcal{L}_{\mathrm{topo}}^{\mathrm{raw}}$.

Surrogate-based backward realization. The implementation-level surrogate scalar is then defined in Eq. 33.

$$\widetilde{\mathcal{L}}_{\mathrm{topo}} = \langle \mathrm{stopgrad}(g^{\mathrm{use}}), z\rangle - \mathrm{stopgrad}(\langle \mathrm{stopgrad}(g^{\mathrm{use}}), z\rangle) + \mathrm{stopgrad}\big(\mathcal{L}_{\mathrm{topo}}^{\mathrm{raw}}\big), \tag{33}$$

where $\mathrm{stopgrad}(\cdot)$ preserves forward values while blocking gradients. Eq. 33 is a value-preserving surrogate construction: its forward value matches $\mathcal{L}_{\mathrm{topo}}^{\mathrm{raw}}$, while its backward gradient equals g^{use}. When combined with the outer topology-entry schedule in Eq. 9, this surrogate realizes the effective backward direction in Eq. 14. The corresponding backward exactness statement is proved in Appendix B.7.

Per-term realization used in the reported method. The notation in Sec. 3 describes the constrained topology-side update at the level of a generic topology gradient. In the default reported implementation, this operation is applied to each retained event-level proxy term before summation. Specifically, for each $m \in \mathcal{K}$, we compute the event-level topology gradient $g_{\mathrm{topo}}^{(m)} = \nabla_z(w_m \Phi_m)$, apply the same balancing, projection, and cap operations relative to g_{pix}, and then sum the resulting constrained event-level directions. The sum-level projection variant in Table 3 instead applies the same operations after aggregating the topology gradient over m. Thus, the ablation isolates whether conservative injection is performed before or after event-level aggregation, while keeping the carrier and PH proxy formulation fixed.

Projection variants used in ablation. The default reported method uses the per-term realization described above. The sum-level projection and direct-addition variants in Table 3 are controlled diagnostic variants used to isolate the role of conservative topology-consistency injection.

B Formal Properties and Proofs

This section records several local formal properties of the proposed reformulation. These statements are used to support the regularity, one-sided conservativeness, uncertainty-aware modulation, projection geometry, magnitude control, and backward-exactness claims invoked in Sec. 3. They are local properties of the present formulation and implementation pathway and should not be interpreted as stronger global guarantees such as convergence, consistency, or optimality of the overall training dynamics. In particular, the differentiability result below is stated on regular regions separated by a finite family of switching sets, and its almost-everywhere form is obtained by excluding a measure-zero exceptional set.

B.1 Standing Assumptions

The statements in this section are understood under the following assumptions.

Assumption 1. The threshold family $\{\tau_m\}_{m=1}^{M}$ and the directional-field family $\{h_d\}_{d=1}^{D}$ are finite, with $M, D < \infty$.

Assumption 2. The sigmoid in Eq. 1 is smooth, and the smoothing coefficient satisfies $\delta > 0$. Consequently, for each fixed threshold τ_m, the map $u \mapsto S(u, \tau_m)$ is smooth.

Assumption 3. The differentiable 4-neighborhood realization in Eq. 15 is used to instantiate the soft Euler trajectory in Eq. 16. Accordingly, the soft counts and the resulting trajectory are finite compositions of additions and multiplications of smooth threshold responses.

Assumption 4. Unless explicitly stated otherwise, all gradients in this section are taken with respect to the student logits z. Teacher-side quantities, including $\tilde{y}$, $r_{\tilde{y}}$, and all target-derived event statistics, are treated as constants during the student update.

Assumption 5. The penalty $\rho(\cdot)$ is nonnegative, C^1, and nondecreasing on $[0, \infty)$, with $\rho(0) = 0$. The fallback coefficients $\gamma(\cdot)$ in Eq. 5 and $\eta(\cdot)$ in Eq. 23 are locally bounded and are either locally constant or locally C^1 on the regions where they are used. The same regularity assumption is imposed on any implementation-level modulation coefficient, including a_{sp} in Eq. 29.

Assumption 6. All stabilized denominators in Eq. 7, Eq. 26, and Eq. 31 are strictly positive because of the associated numerical stabilizers. In addition, the effective temperature in Eq. 27 satisfies $T_{\mathrm{eff}} > 0$.

Assumption 7. Let $\mathcal{S}$ denote the finite family of scalar functions whose zero sets determine branch changes in the student-side computation. The family $\mathcal{S}$ consists of the following switching scalars:

1. the arguments of $[\cdot]_+$ in Eq. 18 and Eq. 22;
2. the arguments of $|\cdot|$ in Eq. 8;
3. the student-side absolute-value arguments appearing in the saliency instantiation, including $\Delta\chi_p^{(m)}$ whenever Eq. 25 uses prediction-side saliency;
4. whenever the saliency rule depends on student-side quantities, the pairwise differences of the branch-resolved smooth saliency scores whose zero sets induce top-k ties in Eq. 27;
5. any student-dependent hard-switching scalar used by implementation-level attenuation coefficients, if such coefficients are not treated as stop-gradient constants.

Each nontrivial element of $\mathcal{S}$ is assumed to be C^1 and to have 0 as a regular value whenever its zero set is nonempty.

Remark 1. The regular-value requirement in Assumption 7 is a standard nondegeneracy condition introduced only to exclude nongeneric branch-contact pathologies. It should be interpreted as a genericity assumption rather than as an additional modeling assumption specific to the proposed method. In particular, Sard's theorem implies that, for smooth maps, almost every target value is a regular value. Thus, requiring 0 to be a regular value excludes only nongeneric degenerate switching events.

Assumption 8. A point z_0 is called a regular point if it does not belong to the zero set of any element of $\mathcal{S}$. Equivalently, at a regular point, all piecewise-smooth branches appearing in the student-side computation remain locally fixed.

Assumption 9. Whenever Eq. 11 is invoked, the pixel gradient satisfies $\|g_{\mathrm{pix}}\| > 0$.

Assumption 10. Inner products and norms are Euclidean unless otherwise stated. In Eq. 33, $\mathrm{stopgrad}(\cdot)$ preserves forward values and blocks gradients in the standard autodifferentiation sense.

Assumption 11. The thresholds satisfy $\tau_1 < \tau_2 < \cdots < \tau_M$, and the retained event set $\mathcal{K}$ is nonempty, i.e., $|\mathcal{K}| \geq 1$.

Under Assumptions 1–11, the subsequent propositions establish the local formal properties invoked by the main-text method formulation.

B.2 Differentiability of the PH Surrogate

Lemma 1. Let z_0 be a regular point in the sense of Assumption 8. Then there exists an open neighborhood U of z_0 such that, throughout U,

1. every positive-part branch appearing in Eq. 18 and Eq. 22 remains fixed;
2. every absolute-value branch appearing in Eq. 8 remains fixed;
3. whenever the saliency rule depends on student-side quantities, the induced top-k index set in Eq. 27 remains fixed.

Proof. By Assumption 8, the point z_0 does not belong to the zero set of any switching scalar in the finite family $\mathcal{S}$. Hence, for every $s \in \mathcal{S}$, one has $s(z_0) \neq 0$. Since each s is continuous, there exists an open neighborhood U_s of z_0 such that Eq. 34 holds.

$$\operatorname{sign}(s(z)) = \operatorname{sign}(s(z_0)), \qquad \forall z \in U_s. \tag{34}$$

Because $\mathcal{S}$ is finite by Assumption 7, the common neighborhood is defined in Eq. 35.

$$U = \bigcap_{s \in \mathcal{S}} U_s. \tag{35}$$

Eq. 34 implies sign preservation for each switching scalar on its corresponding neighborhood U_s, whereas Eq. 35 defines the common neighborhood on which all such sign constraints hold simultaneously. Therefore all associated piecewise-smooth branches remain fixed throughout U, which proves the claim.

Proposition 1. Let z_0 be a regular point in the sense of Assumption 8. Then the instantiated topology surrogate obtained by substituting the implementation-level branches from Appendix A into Eq. 4 is C^1 with respect to the student logits z in a neighborhood of z_0.

Proof. We verify the claim along the construction chain used in Sec. 3. The student probability field is introduced in Eq. 36.

$$p = \sigma(z). \tag{36}$$

Eq. 36 defines a C^∞ map from the student logits z to the probability field p.

For each fixed threshold τ_m, the soft threshold response is given in Eq. 37.

$$S(p, \tau_m) = \sigma\left(\frac{p - \tau_m}{\delta}\right). \tag{37}$$

By Assumption 2, Eq. 37 is C^∞ in p, and therefore C^∞ in z. Under Assumption 3, the soft counts in Eq. 15 are finite polynomial expressions in the entries of $S(p, \tau_m)$. Consequently, the soft Euler trajectory is defined in Eq. 38.

$$\chi_p(\tau_m) = V(S(p, \tau_m)) - E(S(p, \tau_m)) + F(S(p, \tau_m)). \tag{38}$$

Eq. 37 and Eq. 38 together show that the thresholded field and its induced soft Euler trajectory are smooth compositions of the student logits prior to the introduction of piecewise-smooth branch operators.

The adjacent-threshold variation is defined in Eq. 39.

$$\Delta\chi_p^{(m)} = \chi_p(\tau_{m+1}) - \chi_p(\tau_m). \tag{39}$$

Since Eq. 39 is the difference of two C^∞ quantities from Eq. 38, it is C^∞ in z. The birth-type and death-type activities are then introduced in Eq. 40.

$$b_p^{(m)} = [\Delta\chi_p^{(m)}]_+, \qquad d_p^{(m)} = [-\Delta\chi_p^{(m)}]_+. \tag{40}$$

Eq. 40 composes Eq. 39 with the positive-part operator. By Lemma 1, the corresponding positive-part branches remain fixed in a neighborhood of z_0. Hence, on that neighborhood, Eq. 40 reduces locally to either the identity map or the zero map, and both $b_p^{(m)}$ and $d_p^{(m)}$ are C^1.

The exceed-only and fallback branches of the proxy operator are given in Eq. 41 and Eq. 42, respectively.

$$\Phi_m^{\mathrm{exc}} = \rho\Big([b_p^{(m)} - b_{\tilde{y}}^{(m)}]_+\Big) + \rho\Big([d_p^{(m)} - d_{\tilde{y}}^{(m)}]_+\Big). \tag{41}$$

$$\Phi_m^{\mathrm{fb}} = \eta(r_{\tilde{y}})\Big(\rho(b_p^{(m)}) + \rho(d_p^{(m)})\Big). \tag{42}$$

Eq. 41 isolates the exceed-only contribution, whereas Eq. 42 captures the reliability-aware fallback contribution. By Assumptions 4 and 5, both equations define C^1 quantities in a neighborhood of z_0, because the teacher-side terms are frozen and the penalty and modulation functions are C^1 on the active region.

We next analyze the event-weighting branch. Under Assumption 4, the auto-switching rule in Eq. 25 depends only on teacher-side reliability and is therefore constant with respect to student differentiation. If the active saliency expression is target-only, the corresponding saliency values are constant with respect to z. Otherwise, Lemma 1 implies that the induced top-k set remains locally fixed and that all tie-inducing saliency-difference functions stay away from zero in a neighborhood of z_0. The normalized saliency is defined in Eq. 43.

$$\hat{c}_m = \frac{c_m}{\frac{1}{M-1}\sum_j c_j + \varepsilon}. \tag{43}$$

The corresponding masked softmax weighting rule is given in Eq. 44.

$$w_m = \frac{\exp(\hat{c}_m/T_{\mathrm{eff}})\mathbf{1}[m \in \mathcal{K}]}{\sum_j \exp(\hat{c}_j/T_{\mathrm{eff}})\mathbf{1}[j \in \mathcal{K}]}. \tag{44}$$

Eq. 43 is C^1 by Assumption 6. In addition, Assumption 11 guarantees that the retained set $\mathcal{K}$ is nonempty. Therefore the denominator in Eq. 44 is strictly positive, so Eq. 44 is well defined on the same regular neighborhood. Since the active branch and the top-k mask are locally fixed, Eq. 44 is C^1 there.

For the spatial branch, the band response is defined in Eq. 45.

$$B_m(u) = S(u, \tau_m) - S(u, \tau_{m+1}), \tag{45}$$

Under Assumption 11, the thresholds are strictly ordered, so each interval $[\tau_m, \tau_{m+1}]$ is well defined. By Eq. 45 and Assumption 2, the band response is C^1 in u, and therefore C^1 in z. Since the denominator in Eq. 7 is strictly positive by Assumption 6, the associated moment map is C^1. By Lemma 1, the absolute-value branches in Eq. 8 remain fixed in a neighborhood of z_0, and hence the analytical spatial term in Eq. 8 is C^1 there. The implementation-level spatial branch is given by Eq. 29, which preserves C^1-regularity under Assumption 5. Therefore the spatial contribution entering Eq. 24 is C^1 on the same neighborhood.

Finally, Eq. 4 expresses $\mathcal{L}_{\mathrm{PHP}}$ as a finite weighted sum of event-level proxy terms. The preceding steps show that, on a neighborhood of z_0, the weighting rule in Eq. 44, the exceed-only and fallback branches in Eq. 41 and Eq. 42, and the spatial contribution entering Eq. 24 are all C^1. Therefore Eq. 4 is a finite sum of C^1 terms on that neighborhood, and $\mathcal{L}_{\mathrm{PHP}}$ is C^1 with respect to z near z_0.

Corollary 1. Under Assumptions 1–11, $\mathcal{L}_{\mathrm{PHP}}$ is differentiable with respect to z almost everywhere.

Proof. Let the exceptional set be defined in Eq. 46.

$$\mathcal{E} = \bigcup_{s \in \mathcal{S}} \{z : s(z) = 0\}. \tag{46}$$

Eq. 46 collects all branch-switching locations that must be excluded from the local C^1-analysis. Since $\mathcal{S}$ is finite, it suffices to show that each set $\{z : s(z) = 0\}$ has Lebesgue measure zero.

Fix any nontrivial $s \in \mathcal{S}$. By Assumption 7, the map s is C^1, and 0 is a regular value whenever the zero set is nonempty. Therefore, by the regular value theorem, the zero set $\{z : s(z) = 0\}$ is either empty or a C^1 embedded submanifold of codimension 1. Any embedded C^1 submanifold of positive codimension in a Euclidean space is locally the graph of a C^1 map over a lower-dimensional coordinate chart, and therefore has Lebesgue measure zero.

Because Eq. 46 is a finite union of measure-zero sets, the exceptional set $\mathcal{E}$ also has measure zero. On the complement of $\mathcal{E}$, every point is regular in the sense of Assumption 8, and Proposition 1 applies. Therefore $\mathcal{L}_{\mathrm{PHP}}$ is differentiable with respect to z almost everywhere.

B.3 One-sided Exceed-only Property of the Proxy Operator

Proposition 2. For a fixed threshold interval m, let the exceed-only component of the proxy operator be defined by Eq. 47.

$$\Phi_m^{\mathrm{exc}} = \rho\Big([b_p^{(m)} - b_{\tilde{y}}^{(m)}]_+\Big) + \rho\Big([d_p^{(m)} - d_{\tilde{y}}^{(m)}]_+\Big). \tag{47}$$

Assume that the inequalities in Eq. 48 hold.

$$b_p^{(m)} \le b_{\tilde{y}}^{(m)}, \qquad d_p^{(m)} \le d_{\tilde{y}}^{(m)}. \tag{48}$$

Then $\Phi_m^{\mathrm{exc}} \ge 0$ and $\Phi_m^{\mathrm{exc}} = 0$. Consequently, apart from the fallback and spatial terms, Eq. 5 penalizes only prediction-side topological activity that exceeds the target-referenced level on the current threshold interval.

Proof. By definition of the positive-part operator, one has $[a]_+ \ge 0$ for all a, and $[a]_+ = 0$ whenever $a \le 0$. Therefore the two bracketed terms in Eq. 47 are nonnegative. Since $\rho(\cdot)$ is nonnegative on $[0, \infty)$ and satisfies $\rho(0) = 0$ by Assumption 5, each summand in Eq. 47 is nonnegative. Hence $\Phi_m^{\mathrm{exc}} \ge 0$.

If Eq. 48 holds, then both positive-part terms in Eq. 47 vanish. Substituting Eq. 48 into Eq. 47 gives $\Phi_m^{\mathrm{exc}} = \rho(0) + \rho(0) = 0$. Therefore the exceed-only component vanishes whenever the prediction-side birth-type and death-type activities do not exceed their target-referenced counterparts. This proves both claims.

Corollary 2. If the fallback and spatial terms are omitted from Eq. 5, then the resulting operator is conservative in the sense that it does not penalize prediction-side topological activity lying entirely below the target-referenced level on the current threshold interval.

Proof. When the fallback and spatial terms are removed from Eq. 5, the remaining operator is exactly the exceed-only component defined in Eq. 47. Under the inequalities in Eq. 48, Proposition 2 yields $\Phi_m^{\mathrm{exc}} = 0$. Therefore, in the absence of the fallback and spatial branches, no penalty is imposed on prediction-side topological activity that remains entirely below the target-referenced level on the current threshold interval.

B.4 Monotonicity of the Fallback Strength with Respect to Target-side Uncertainty

Proposition 3. Let the target-side uncertainty u_t and the corresponding reliability score $r_{\tilde{y}}$ be defined by Eq. 19 and Eq. 20, respectively. Suppose that the implementation-level fallback coefficient $\eta(r_{\tilde{y}})$ in Eq. 23 is nonincreasing with respect to $r_{\tilde{y}}$. Then $\eta(r_{\tilde{y}})$ is nondecreasing with respect to u_t. Furthermore, for fixed $b_p^{(m)}$ and $d_p^{(m)}$, the fallback component in Eq. 49 is nondecreasing with respect to u_t.

$$\Phi_m^{\mathrm{fb}} = \eta(r_{\tilde{y}})\Big(\rho(b_p^{(m)}) + \rho(d_p^{(m)})\Big). \tag{49}$$

Proof. The reliability score is given in Eq. 50.

$$r_{\tilde{y}} = \mathrm{clip}\Big(1 - \frac{u_t}{0.25},\, 0,\, 1\Big). \tag{50}$$

Eq. 50 expresses $r_{\tilde{y}}$ as the clipping of the affine map $1 - u_t/0.25$. Since the affine map $1 - u_t/0.25$ is nonincreasing in u_t, and clipping preserves order, Eq. 50 defines a nonincreasing map from u_t to $r_{\tilde{y}}$. Hence $r_{\tilde{y}}$ is nonincreasing with respect to u_t.

By assumption, $\eta(r_{\tilde{y}})$ is nonincreasing with respect to $r_{\tilde{y}}$. Therefore the composite map $u_t \mapsto \eta(r_{\tilde{y}}(u_t))$ is nondecreasing with respect to u_t. This proves the first claim.

For the second claim, the prediction-side factor in Eq. 49 is nonnegative. More precisely, Eq. 51 holds by Assumption 5.

$$\rho(b_p^{(m)}) + \rho(d_p^{(m)}) \ge 0. \tag{51}$$

When $b_p^{(m)}$ and $d_p^{(m)}$ are fixed, Eq. 51 shows that the factor multiplying $\eta(r_{\tilde{y}})$ in Eq. 49 is a nonnegative constant with respect to u_t. Multiplication by a nonnegative constant preserves monotonicity. Therefore Eq. 49 is nondecreasing with respect to u_t.

Remark 2. Proposition 3 is intentionally stated for fixed prediction-side activities $b_p^{(m)}$ and $d_p^{(m)}$. This is the precise scope required to justify monotone strengthening of the fallback branch with increasing target-side uncertainty, without conflating that effect with the separate dependence of the prediction-side activities on the student output.

B.5 Geometry of Conflict-aware Projection

Proposition 4. Assume that $\|g_{\mathrm{pix}}\| > 0$, and let $g_{\mathrm{topo}}^{\perp}$ be defined by Eq. 11. Then the following statements hold:

1. $\langle g_{\mathrm{topo}}^{\perp}, g_{\mathrm{pix}} \rangle = 0$;
2. among all vectors h satisfying $\langle h, g_{\mathrm{pix}} \rangle = 0$, the vector $g_{\mathrm{topo}}^{\perp}$ uniquely minimizes $\|h - g_{\mathrm{topo}}\|$;
3. if the injected topological gradient is defined by Eq. 12, then its inner product with the pixel gradient satisfies Eq. 52,

$$\langle g_{\mathrm{topo}}^{\mathrm{inj}}, g_{\mathrm{pix}} \rangle = \max\big(0, \langle g_{\mathrm{topo}}, g_{\mathrm{pix}} \rangle\big). \tag{52}$$

   where the conflict condition is given by Eq. 53,

$$\langle g_{\mathrm{topo}}, g_{\mathrm{pix}} \rangle < 0. \tag{53}$$

Proof. By Eq. 11, the projected topology gradient is given by Eq. 54.

$$g_{\mathrm{topo}}^{\perp} = g_{\mathrm{topo}} - \frac{\langle g_{\mathrm{topo}}, g_{\mathrm{pix}} \rangle}{\|g_{\mathrm{pix}}\|^2} g_{\mathrm{pix}}. \tag{54}$$

Taking the inner product of Eq. 54 with g_{pix} yields Eq. 55.

$$\begin{aligned} \langle g_{\mathrm{topo}}^{\perp}, g_{\mathrm{pix}} \rangle &= \langle g_{\mathrm{topo}}, g_{\mathrm{pix}} \rangle - \frac{\langle g_{\mathrm{topo}}, g_{\mathrm{pix}} \rangle}{\|g_{\mathrm{pix}}\|^2} \langle g_{\mathrm{pix}}, g_{\mathrm{pix}} \rangle \\ &= 0. \end{aligned} \tag{55}$$

This proves item (1).

To prove item (2), let h satisfy $\langle h, g_{\mathrm{pix}} \rangle = 0$. Then $h - g_{\mathrm{topo}}^{\perp}$ is orthogonal to g_{pix}, whereas the removed correction is identified by Eq. 56.

$$g_{\mathrm{topo}} - g_{\mathrm{topo}}^{\perp} = \frac{\langle g_{\mathrm{topo}}, g_{\mathrm{pix}} \rangle}{\|g_{\mathrm{pix}}\|^2} g_{\mathrm{pix}}. \tag{56}$$

Eq. 56 shows that $g_{\mathrm{topo}} - g_{\mathrm{topo}}^{\perp}$ is parallel to g_{pix}. Hence $h - g_{\mathrm{topo}}^{\perp}$ is orthogonal to $g_{\mathrm{topo}} - g_{\mathrm{topo}}^{\perp}$, and the corresponding Pythagorean decomposition is given by Eq. 57.

$$\begin{aligned} \|h - g_{\mathrm{topo}}\|^2 &= \|(h - g_{\mathrm{topo}}^{\perp}) + (g_{\mathrm{topo}}^{\perp} - g_{\mathrm{topo}})\|^2 \\ &= \|h - g_{\mathrm{topo}}^{\perp}\|^2 + \|g_{\mathrm{topo}}^{\perp} - g_{\mathrm{topo}}\|^2 \\ &\geq \|g_{\mathrm{topo}}^{\perp} - g_{\mathrm{topo}}\|^2. \end{aligned} \tag{57}$$

Equality in Eq. 57 holds if and only if $h = g_{\mathrm{topo}}^{\perp}$. Therefore Eq. 56 and Eq. 57 together show that $g_{\mathrm{topo}}^{\perp}$ is the unique orthogonal projection of g_{topo} onto the subspace $\{h : \langle h, g_{\mathrm{pix}} \rangle = 0\}$, and hence the unique closest vector in that subspace. This proves item (2).

To prove item (3), Eq. 12 defines two cases. If Eq. 53 holds, then $g_{\mathrm{topo}}^{\mathrm{inj}} = g_{\mathrm{topo}}^{\perp}$, and Eq. 55 yields Eq. 58.

$$\langle g_{\mathrm{topo}}^{\mathrm{inj}}, g_{\mathrm{pix}} \rangle = 0. \tag{58}$$

If Eq. 53 does not hold, then $g_{\mathrm{topo}}^{\mathrm{inj}} = g_{\mathrm{topo}}$, and therefore Eq. 59 holds.

$$\langle g_{\mathrm{topo}}^{\mathrm{inj}}, g_{\mathrm{pix}} \rangle = \langle g_{\mathrm{topo}}, g_{\mathrm{pix}} \rangle \geq 0. \tag{59}$$

Combining Eq. 58 and Eq. 59 gives Eq. 52.

Corollary 3. If the capped gradient is defined by Eq. 31, then Eq. 60 holds.

$$\langle \bar{g}_{\mathrm{topo}}^{\mathrm{inj}}, g_{\mathrm{pix}} \rangle \geq 0. \tag{60}$$

Proof. Eq. 31 defines $\bar{g}_{\mathrm{topo}}^{\mathrm{inj}}$ as the scaled vector in Eq. 61.

$$\bar{g}_{\mathrm{topo}}^{\mathrm{inj}} = \lambda\, g_{\mathrm{topo}}^{\mathrm{inj}}, \qquad \lambda = \min\left(1,\ \frac{c_{\mathrm{cap}}\|g_{\mathrm{pix}}\|}{\|g_{\mathrm{topo}}^{\mathrm{inj}}\| + \varepsilon}\right). \tag{61}$$

By definition, $\lambda \geq 0$. Taking the inner product of Eq. 61 with g_{pix} yields Eq. 62.

$$\left\langle \bar{g}_{\mathrm{topo}}^{\mathrm{inj}}, g_{\mathrm{pix}} \right\rangle = \lambda \left\langle g_{\mathrm{topo}}^{\mathrm{inj}}, g_{\mathrm{pix}} \right\rangle. \tag{62}$$

By Eq. 52, the inner product $\langle g_{\mathrm{topo}}^{\mathrm{inj}}, g_{\mathrm{pix}} \rangle$ is nonnegative. Therefore Eq. 62, together with $\lambda \geq 0$, implies Eq. 60. Thus the capping step preserves the nonnegative alignment property established in Proposition 4.

B.6 Boundedness of the Relative-cap Operator

Proposition 5. Let the capped topology gradient $\bar{g}_{\mathrm{topo}}^{\mathrm{inj}}$ be defined by Eq. 31. Then the norm bound in Eq. 63 holds.

$$\|\bar{g}_{\mathrm{topo}}^{\mathrm{inj}}\| \leq c_{\mathrm{cap}}\|g_{\mathrm{pix}}\|. \tag{63}$$

Moreover, if the inactive-cap condition in Eq. 64 holds,

$$\|g_{\mathrm{topo}}^{\mathrm{inj}}\| + \varepsilon \leq c_{\mathrm{cap}}\|g_{\mathrm{pix}}\|, \tag{64}$$

then the cap is inactive and Eq. 65 follows.

$$\bar{g}_{\mathrm{topo}}^{\mathrm{inj}} = g_{\mathrm{topo}}^{\mathrm{inj}}. \tag{65}$$

Proof. By Eq. 31, the capped topology gradient admits the representation in Eq. 66.

$$\bar{g}_{\mathrm{topo}}^{\mathrm{inj}} = \lambda g_{\mathrm{topo}}^{\mathrm{inj}}, \qquad \lambda = \min\left(1,\ \frac{c_{\mathrm{cap}}\|g_{\mathrm{pix}}\|}{\|g_{\mathrm{topo}}^{\mathrm{inj}}\| + \varepsilon}\right). \tag{66}$$

By definition, one has $\lambda \geq 0$. Consequently, Eq. 66 implies the norm identity in Eq. 67.

$$\|\bar{g}_{\mathrm{topo}}^{\mathrm{inj}}\| = \lambda \|g_{\mathrm{topo}}^{\mathrm{inj}}\|. \tag{67}$$

We first prove the inactive-cap statement. If Eq. 64 holds, then $\frac{c_{\mathrm{cap}}\|g_{\mathrm{pix}}\|}{\|g_{\mathrm{topo}}^{\mathrm{inj}}\|+\varepsilon} \geq 1$, so the minimum in Eq. 66 is attained by its first argument. Hence $\lambda = 1$, and substituting this identity into Eq. 66 yields Eq. 65. This proves the second statement.

It remains to prove Eq. 63. If Eq. 64 holds, then Eq. 65 shows that $\bar{g}_{\mathrm{topo}}^{\mathrm{inj}} = g_{\mathrm{topo}}^{\mathrm{inj}}$, and Eq. 64 immediately gives Eq. 63. If Eq. 64 does not hold, then the second argument of the minimum in Eq. 66 is strictly smaller than 1, and therefore Eq. 68 holds.

$$\lambda = \frac{c_{\mathrm{cap}}\|g_{\mathrm{pix}}\|}{\|g_{\mathrm{topo}}^{\mathrm{inj}}\| + \varepsilon}. \tag{68}$$

Substituting Eq. 68 into Eq. 67 yields Eq. 69.

$$\begin{aligned} \|\bar{g}_{\mathrm{topo}}^{\mathrm{inj}}\| &= \frac{c_{\mathrm{cap}}\|g_{\mathrm{pix}}\|}{\|g_{\mathrm{topo}}^{\mathrm{inj}}\| + \varepsilon}\|g_{\mathrm{topo}}^{\mathrm{inj}}\| \\ &= c_{\mathrm{cap}}\|g_{\mathrm{pix}}\|\frac{\|g_{\mathrm{topo}}^{\mathrm{inj}}\|}{\|g_{\mathrm{topo}}^{\mathrm{inj}}\| + \varepsilon} \leq c_{\mathrm{cap}}\|g_{\mathrm{pix}}\|. \end{aligned} \tag{69}$$

Thus Eq. 63 holds in both cases.

Corollary 4. The topology contribution appearing in the final update of Eq. 14 satisfies Eq. 70.

$$\left\|\lambda_{\mathrm{topo}}(t)\alpha_t \bar{g}_{\mathrm{topo}}^{\mathrm{inj}}\right\| \le \lambda_{\mathrm{topo}}(t)\alpha_t c_{\mathrm{cap}}\|g_{\mathrm{pix}}\|, \tag{70}$$

Eq. 70 is a direct scaled consequence of the pre-scaled cap in Proposition 5. It should not be interpreted as an additional relative-cap guarantee at the final update level.

Proof. By Proposition 5, the capped topology vector satisfies Eq. 71.

$$\|\bar{g}_{\mathrm{topo}}^{\mathrm{inj}}\| \le c_{\mathrm{cap}}\|g_{\mathrm{pix}}\|, \tag{71}$$

Since $\lambda_{\mathrm{topo}}(t) \ge 0$ by construction and $\alpha_t \ge 0$ by Eq. 10, multiplying both sides of Eq. 71 by the nonnegative scalar $\lambda_{\mathrm{topo}}(t)\alpha_t$ preserves the inequality and yields Eq. 70. Thus Eq. 70 only states the scaled norm relation induced by Proposition 5 and does not introduce a new cap statement for the final update term.

B.7 Exactness of the Surrogate-based Backward Realization

Throughout this subsection, g^{use} denotes the pre-schedule topology-side backward vector defined in Eq. 32, and $\widetilde{\mathcal{L}}_{\mathrm{topo}}$ denotes the value-preserving surrogate scalar defined in Eq. 33.

Proposition 6. Let $\widetilde{\mathcal{L}}_{\mathrm{topo}}$ denote the surrogate scalar defined by Eq. 33. Then the following two statements hold:

1. the forward value of $\widetilde{\mathcal{L}}_{\mathrm{topo}}$ is exactly $\mathcal{L}_{\mathrm{topo}}^{\mathrm{raw}}$;
2. the gradient of $\widetilde{\mathcal{L}}_{\mathrm{topo}}$ with respect to z is exactly the pre-schedule constrained topology-side backward vector g^{use}.

Remark 3. The construction in Eq. 33 is a deterministic surrogate-gradient realization of the same algebraic form as the surrogate-objective constructions used in stochastic computation graphs and related surrogate-gradient methods. Its role is to preserve the forward scalar while replacing the backward signal by the desired gradient object.

Proof. We first prove the forward-value statement. By Assumption 10, $\mathrm{stopgrad}(\cdot)$ preserves forward values. Applying this property directly to Eq. 33 yields Eq. 72.

$$\begin{aligned}\widetilde{\mathcal{L}}_{\mathrm{topo}} &= \langle \mathrm{stopgrad}(g^{\mathrm{use}}), z\rangle - \mathrm{stopgrad}(\langle \mathrm{stopgrad}(g^{\mathrm{use}}), z\rangle) + \mathrm{stopgrad}\big(\mathcal{L}_{\mathrm{topo}}^{\mathrm{raw}}\big) \\ &= \langle g^{\mathrm{use}}, z\rangle - \langle g^{\mathrm{use}}, z\rangle + \mathcal{L}_{\mathrm{topo}}^{\mathrm{raw}} \\ &= \mathcal{L}_{\mathrm{topo}}^{\mathrm{raw}}.\end{aligned} \tag{72}$$

Eq. 72 proves the first statement.

We next prove the gradient statement. By Assumption 10, all quantities wrapped by $\mathrm{stopgrad}(\cdot)$ are treated as constants with respect to z. Consequently, among the terms in Eq. 33, the only term that contributes a nonzero gradient with respect to z is the bilinear term $\langle \mathrm{stopgrad}(g^{\mathrm{use}}), z\rangle$. Since $\mathrm{stopgrad}(g^{\mathrm{use}})$ is constant with respect to z, differentiating that term yields Eq. 73.

$$\nabla_z \langle \mathrm{stopgrad}(g^{\mathrm{use}}), z\rangle = g^{\mathrm{use}}. \tag{73}$$

Therefore the surrogate scalar satisfies Eq. 74.

$$\nabla_z \widetilde{\mathcal{L}}_{\mathrm{topo}} = g^{\mathrm{use}}. \tag{74}$$

Eq. 74 proves the second statement.

Corollary 5. Replacing $\mathcal{L}_{\mathrm{topo}}^{\mathrm{raw}}$ by $\widetilde{\mathcal{L}}_{\mathrm{topo}}$ preserves the forward scalar value while replacing the raw topology gradient by the pre-schedule constrained topology-side backward direction.

Proof. By Proposition 6, Eq. 72 shows that $\widetilde{\mathcal{L}}_{\mathrm{topo}}$ has exactly the same forward value as $\mathcal{L}_{\mathrm{topo}}^{\mathrm{raw}}$. Hence the forward scalar value is unchanged under surrogate substitution. On the other hand, Eq. 74 shows that the gradient of $\widetilde{\mathcal{L}}_{\mathrm{topo}}$ with respect to z is exactly g^{use}, rather than the raw topology gradient. Therefore surrogate substitution preserves the forward scalar value while replacing the backward direction used for optimization.

C Experimental Configuration Details

This section specifies the reported configuration used in the main results. It complements Sec. 4.1 by fixing the reported topology protocol and PH proxy instantiation, without repeating the generic training setup already given in the main text.

C.1 Reported Configuration

Unless an explicit deviation is stated for an ablation or supplementary UCOD comparison, all UCOD topology-related results follow the shared reported configuration below. The matched cross-task histology experiments use task-specific settings and are specified separately in Appendix D.3. In the reported main runs, the proposed PH proxy is the sole active topology supervision signal. The overall topology weight is set to 1.0. To remain consistent with the conservative injection principle in Sec. 3, topology supervision is activated from epoch 10 rather than at the beginning of training, and the topology branch is evaluated once every four training iterations.

For the reported main results, the PH proxy is instantiated with 16 thresholds uniformly spanning $[0.05,\ 0.95]$, a soft-threshold smoothing coefficient of 0.08, a topology-evaluation downsampling factor of 4, 4 directional moments, and at most 12 retained threshold intervals. All implementation-level constants in the topology branch are fixed across the reported runs and are not tuned separately for individual datasets or compared methods. Appendix A.4 and Appendix A.5 specify how these constants enter the event-weighting and spatial-anchoring branches.

This reported configuration is shared by both the main UCOD benchmark evaluation of the proposed method and the controlled UCOD topology-level comparisons. For the benchmark evaluation, it is combined with the general UCOD-DPL training setup summarized in Sec. 4.1. For the controlled comparisons, the carrier, backbone family, training recipe, and pseudo-label pipeline are fixed, and only the adopted topology supervision is varied. In particular, the reproduced UCOD-DPL baseline uses the same carrier, backbone family, input resolution, batch size, learning-rate schedule, and training epochs as our method, and differs only in the adopted topology supervision.

For the reported main results, the event-saliency policy is fixed to the auto-switching construction in Eq. 25. The target-only and mixed expressions in Eq. 25 are therefore not separate reported variants, but two regimes of the same reported policy.

If a particular experiment adopts a different topology weight, topology entry epoch, topology evaluation interval, threshold discretization, event-selection setting, or proxy instantiation, the deviation is stated explicitly together with the corresponding result.

The reported UCOD runs under this configuration are conducted on a workstation equipped with a single NVIDIA GeForce RTX 5090D GPU with 32 GB memory. Under the reported UCOD-DPL carrier and the training setup stated in Sec. 4.1, one training run takes approximately 20 minutes. Inference on the four evaluation benchmarks together with metric computation takes approximately 18 minutes. Since the proposed method does not modify the carrier architecture itself, the inference-time model complexity remains the same as UCOD-DPL, and the additional cost is incurred only during the training-time topology supervision and optimization process.

C.2 Detailed Definitions of Evaluation Metrics

In the main text, we report four standard COD metrics, namely mean absolute error (MAE), structure measure (S_α), enhanced-alignment measure (E_ϕ), and weighted F-measure (F_β^w). To evaluate topology-level structural reliability, we report the counting errors β_0 and β_1, while Betti Matching Error (BME) provides the complementary feature-wise topology evaluation described separately in Appendix C.3. This subsection gives the formal definitions of the standard COD metrics and Betti-counting errors.

Mean Absolute Error (MAE). MAE is defined in Eq. 75 as the average pixel-wise absolute difference between the predicted saliency map and the ground-truth mask.

$$\mathrm{MAE} = \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} \left| P(i,j) - G(i,j) \right|, \tag{75}$$

where $P(i,j) \in [0,1]$ and $G(i,j) \in \{0,1\}$ denote the predicted value and the ground-truth label at location (i,j), respectively. Lower MAE in Eq. 75 indicates better pixel-level fidelity.

Structure Measure (S_α). The structure measure is defined in Eq. 76 as a convex combination of object-aware and region-aware structural consistency.

$$S_\alpha = \alpha S_{\text{object}} + (1-\alpha) S_{\text{region}}, \qquad \alpha \in [0,1]. \tag{76}$$

Eq. 76 gives the canonical form of S_α. A larger S_α indicates better structural preservation. In our experiments, the quantities S_{object} and S_{region} in Eq. 76 are computed following the standard COD evaluation protocol and its canonical implementation.

Enhanced-alignment Measure (E_ϕ). The enhanced-alignment measure is defined in Eq. 77.

$$E_\phi = \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} \phi(B_P(i,j), B_G(i,j)), \tag{77}$$

In Eq. 77, B_P and B_G denote the binarized prediction and ground-truth maps, respectively, and $\phi(\cdot,\cdot)$ denotes the enhanced alignment function. A larger E_ϕ in Eq. 77 indicates stronger perceptual and structural alignment. In our experiments, E_ϕ is computed following the standard COD evaluation protocol.

Weighted F-measure (F_β^w). The weighted F-measure is defined in Eq. 78.

$$F_\beta^w = \frac{(1+\beta^2)\,\text{Precision}_w\,\text{Recall}_w}{\beta^2\,\text{Precision}_w + \text{Recall}_w}, \tag{78}$$

In Eq. 78, Precision_w and Recall_w are the weighted precision and weighted recall, respectively. A larger F_β^w in Eq. 78 indicates a better spatially aware precision–recall trade-off.

Topology-aware counting errors (β_0 and β_1). To quantify topology-level structural discrepancy, we report the average absolute counting errors of the zeroth and first Betti numbers. Let $\hat{M}_n$ and M_n denote the predicted binary mask and the ground-truth binary mask for the n-th test image, respectively, and let N denote the number of evaluated images. For a binary mask M, let $b_0(M)$ denote the number of connected foreground components and let $b_1(M)$ denote the number of holes. The topology-aware counting errors are defined in Eq. 79 and Eq. 80, respectively.

$$\beta_0 = \frac{1}{N} \sum_{n=1}^{N} \left| b_0(\hat{M}_n) - b_0(M_n) \right|, \tag{79}$$

$$\beta_1 = \frac{1}{N} \sum_{n=1}^{N} \left| b_1(\hat{M}_n) - b_1(M_n) \right|. \tag{80}$$

Eq. 79 measures the average counting error in connected foreground components, whereas Eq. 80 measures the average counting error in holes. Under this fixed evaluation protocol, lower values in Eq. 79 and Eq. 80 indicate better topology-level consistency with the ground-truth structure.

Discrete topology evaluation protocol. For the n-th test image, let P_n denote the predicted saliency map after resizing to the ground-truth resolution when necessary. The normalized prediction map is defined in Eq. 81.

$$\bar{P}_n = \frac{P_n - \min(P_n)}{\max(P_n) - \min(P_n) + \varepsilon}, \tag{81}$$

where $\varepsilon > 0$ is a numerical stabilizer. The corresponding binary prediction mask is then defined in Eq. 82.

$$\hat{M}_n(i,j) = \mathbf{1}\left[\bar{P}_n(i,j) \geq 0.5\right], \tag{82}$$

Eq. 81 and Eq. 82 specify the saliency-normalized binarization pipeline used before Betti-number evaluation. This post-hoc pipeline is applied identically to all compared methods and is used only for topology-aware evaluation, not for training or for the pixel-level COD metrics.

Under this discrete evaluation protocol, both connected components and holes are counted under 8-neighborhood connectivity. More specifically, the quantities entering Eq. 79 and Eq. 80 are computed on the binary mask in Eq. 82. In this setting, $b_0(\hat{M}_n)$ is computed as the number of connected foreground components, whereas $b_1(\hat{M}_n)$ is computed by inverting a padded foreground mask and counting connected background components that do not touch the padded image border.

It is important to emphasize that the topology-aware quantities in Eq. 79 and Eq. 80 are discrete post-hoc evaluation metrics defined under the same saliency-normalized mask-extraction protocol for all methods. They should therefore be interpreted as relative structural-discrepancy indicators under this fixed evaluation protocol, rather than as calibration-invariant or threshold-free topological errors. These quantities are also distinct from the differentiable 4-neighborhood relaxation used in the training objective, namely the soft Euler construction in Eq. 2 and its implementation-level realization in Eq. 15.

Together, MAE in Eq. 75, S_α in Eq. 76, E_ϕ in Eq. 77, and F_β^w in Eq. 78 evaluate pixel-level accuracy, structural similarity, perceptual alignment, and spatially aware precision–recall balance, whereas β_0 and β_1 in Eq. 79 and Eq. 80 quantify topology-level discrepancy through Betti-number counting errors. Complementary feature-wise topological correspondence is evaluated using the BME protocol in Appendix C.3.

C.3 Betti Matching Error Protocol

Betti-number counting errors quantify discrepancies in the numbers of connected components and holes, but equal counts do not necessarily imply feature-wise topological correspondence. We therefore additionally report Betti Matching Error (BME) in the controlled topology comparison in Table 2. BME is computed using one fixed, method-independent binary-mask Betti-matching protocol for all compared topology-supervision methods.

For each evaluated image, the prediction and ground-truth mask are jointly resized by nearest-neighbor interpolation when resizing is required, with the longer side capped at 296 pixels. The same preprocessing and Betti-matching procedure are applied to every compared method. Reported BME values are averaged over all evaluated images within each dataset, and lower values indicate closer feature-wise topological correspondence under this fixed protocol.

BME should be interpreted as complementary to the Betti-number counting errors in Appendix C.2. The latter quantify discrepancies in component and hole counts, whereas BME evaluates feature-wise topological correspondence. Consequently, lower BME does not necessarily imply stronger foreground prediction quality, and BME is considered together with the task-level metrics rather than as a replacement for them.

C.4 Ablation Proxy Modes for Table 4

This subsection defines the three PH proxy modes used in Table 4. The internal *Match* proxy mode used in this ablation is distinct from the external MATCH topology-supervision baseline in Table 2. The purpose of this ablation is to isolate the role of the event proxy operator Φ_m under the same carrier and optimization protocol. All three modes use the same event weighting, retained threshold intervals, topology-injection mechanism, backbone, pseudo-label pipeline, and training setting. They differ only in how the birth-like and death-like proxy activities are converted into the interval-level proxy term.

Match. The *Match* mode replaces the one-sided exceed-only terms in Eq. 5 with symmetric discrepancies between prediction-side and target-side proxy activities:

$$\Phi_m^{\mathrm{match}} = \rho\Big(\Big|b_p^{(m)} - b_{\tilde{y}}^{(m)}\Big|\Big) + \rho\Big(\Big|d_p^{(m)} - d_{\tilde{y}}^{(m)}\Big|\Big) + \Psi_m(p, \tilde{y}). \tag{83}$$

This mode encourages direct matching to the target-side proxy activities. It therefore serves as a diagnostic variant for testing whether symmetric target-side proxy matching is suitable when pseudo supervision may itself be structurally distorted.

Simplicity. The *Simplicity* mode removes the target-referenced birth/death discrepancy terms and directly penalizes prediction-side proxy activities:

$$\Phi_m^{\mathrm{simp}} = \rho\Big(b_p^{(m)}\Big) + \rho\Big(d_p^{(m)}\Big) + \Psi_m(p, \tilde{y}). \tag{84}$$

This mode encourages lower prediction-side topological activity without explicitly using target-side birth-like or death-like counts as matching targets. It therefore tests whether suppressing topological complexity alone is sufficient under structurally unreliable pseudo supervision.

Cap. The default *Cap* mode uses the event proxy operator in Eq. 5. Unlike *Match*, it does not enforce symmetric target-side proxy matching. Unlike *Simplicity*, it does not ignore the target-side proxy activities. Instead, it uses one-sided exceed-only penalties together with reliability-aware fallback and spatial anchoring:

$$\Phi_m^{\mathrm{cap}} = \rho\bigg(\Big[b_p^{(m)} - b_{\tilde{y}}^{(m)}\Big]_+\bigg) + \rho\bigg(\Big[d_p^{(m)} - d_{\tilde{y}}^{(m)}\Big]_+\bigg) + \gamma(r_{\tilde{y}})\Big(\rho\Big(b_p^{(m)}\Big) + \rho\Big(d_p^{(m)}\Big)\Big) + \Psi_m(p, \tilde{y}). \tag{85}$$

Here, "Cap" refers to conservative excess control relative to the noisy target-side proxy envelope, rather than to exact Betti-number matching. This formulation is intended to reduce prediction-side excessive topological activity while avoiding direct fitting to potentially distorted supervision-side topology.

D Supplementary Experiments and Results

D.1 Supplementary Qualitative Comparisons

We provide additional qualitative comparisons to complement Fig. 3 in the main text. These examples in Fig. 4 are collected under the same UCOD-DPL carrier and the same set of compared topology supervision strategies, but extend beyond the representative cases selected for the main paper. Across these additional samples, the same qualitative tendency remains consistent. The no-topology baseline often leaves fragmented foreground structure, weak structural discontinuities, or spurious hole-like artifacts unresolved, whereas directly imposed topology losses may over-correct the prediction and suppress fragile object regions. By contrast, the proposed method more consistently improves structural coherence while preserving the main object semantics, which further supports the quantitative observations in Table 2.

D.2 Supplementary Failure-Case Analysis

We provide supplementary failure-case analysis to complement the qualitative comparisons in the main text and Appendix D.1. These examples in Fig. 5 are collected under the same UCOD-DPL carrier and the same set of compared topology supervision strategies, but focus on representative regimes in which all compared methods remain challenged under structurally unreliable pseudo supervision. Across these additional samples, the same qualitative tendency remains consistent. When the recoverable structural evidence is too weak, too thin, or too fragmented, the no-topology baseline often misses the object or leaves incomplete and broken foreground structure unresolved, whereas directly imposed topology losses may over-correct local structure and further suppress already fragile object regions. By contrast, the proposed method is often more conservative and semantically more stable, but complete recovery remains difficult in these regimes. These examples therefore mark the boundary of the present reformulation rather than contradict the main conclusion of the paper.

The failure regimes in Fig. 5 also indicate the current scope limitations of the proposed reformulation. First, although the matched histology experiments in Appendix D.3 provide evidence beyond UCOD, the present validation remains limited to UCOD and three histology segmentation benchmarks; transfer to broader dense-prediction domains and pseudo-supervision regimes remains to be established. Second, when the recoverable structural evidence is too weak, too thin, or too fragmented, complete structural recovery remains difficult for all compared methods, including ours. Third, the formal properties provided in Appendix B are local properties of the present formulation and implementation pathway, and should not be interpreted as global guarantees on the overall training dynamics.

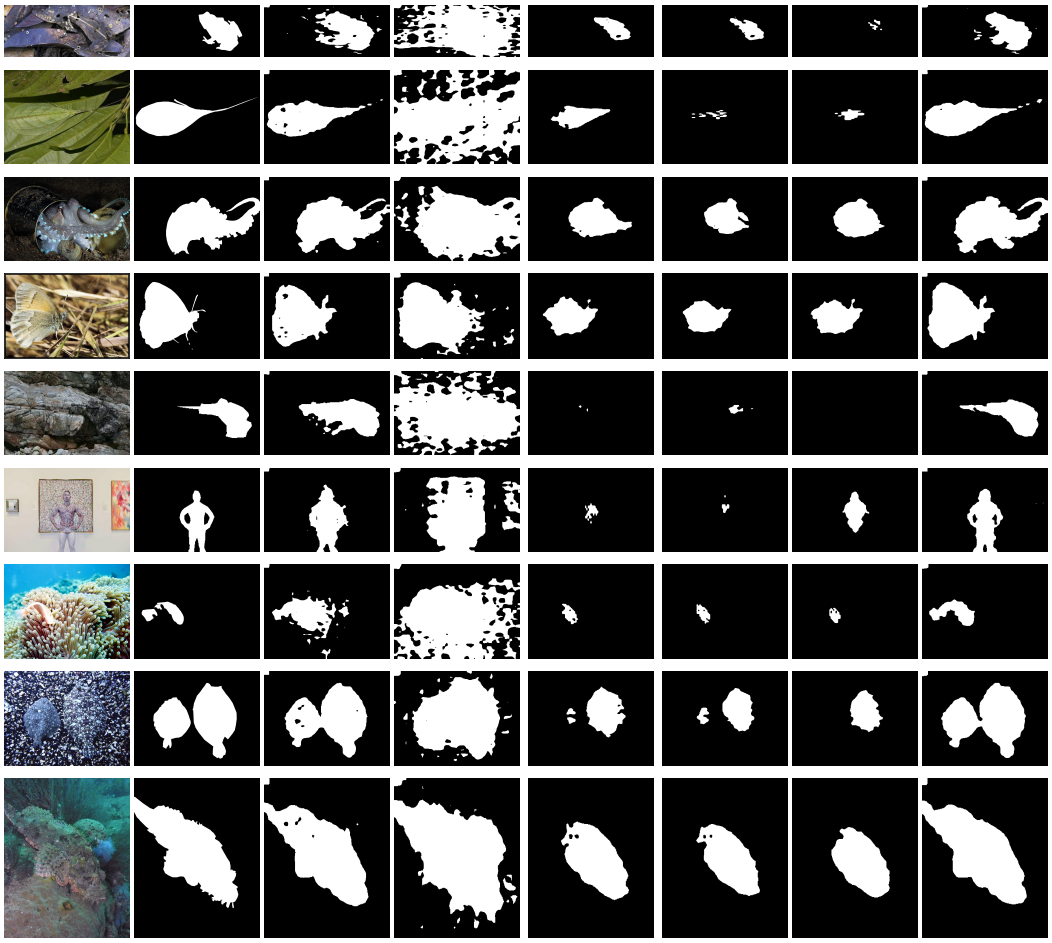


Figure 4: Additional qualitative comparisons under the same UCOD-DPL carrier, complementing Fig. 3.

D.3 Matched Cross-Task Transfer Beyond UCOD

To evaluate transfer beyond UCOD, we conduct matched experiments on three histology segmentation datasets, CRAG [45], GlaS [46], and MoNuSeg [47]. All compared methods share the same carrier, backbone, pseudo-label pipeline, task objective, optimizer, training budget, checkpoint rule, and evaluator; only the topology-supervision formulation and its fixed method-specific coefficient vary. We report weighted F-measure, Dice, and Betti Matching Error (BME), with BME compared only within each dataset.

Table 7: Matched cross-task transfer on histology segmentation. All methods share the same carrier and training pipeline; only the topology-supervision formulation varies. Best task-level results in each dataset are in bold.

Method	CRAG			GlaS			MoNuSeg		
	F_β^w	Dice	BME	F_β^w	Dice	BME	F_β^w	Dice	BME
No topology	0.6684	0.7653	21.980	0.7769	0.8146	14.950	0.2379	0.2993	393.143
TopoLoss	0.4344	0.5685	23.400	0.5406	0.6679	18.038	0.2414	0.2970	426.571
TopoSemiSeg	0.6495	0.7130	18.650	0.6844	0.7161	14.975	0.1458	0.1743	409.071
SATLoss	0.4912	0.5925	18.450	0.5756	0.6532	13.038	0.2390	0.3054	411.000
MATCH	0.5225	0.5706	18.400	0.5277	0.5533	13.625	0.1603	0.1938	409.714
Ours	**0.6881**	**0.7752**	21.675	**0.7867**	**0.8345**	14.475	**0.2483**	**0.3099**	392.857

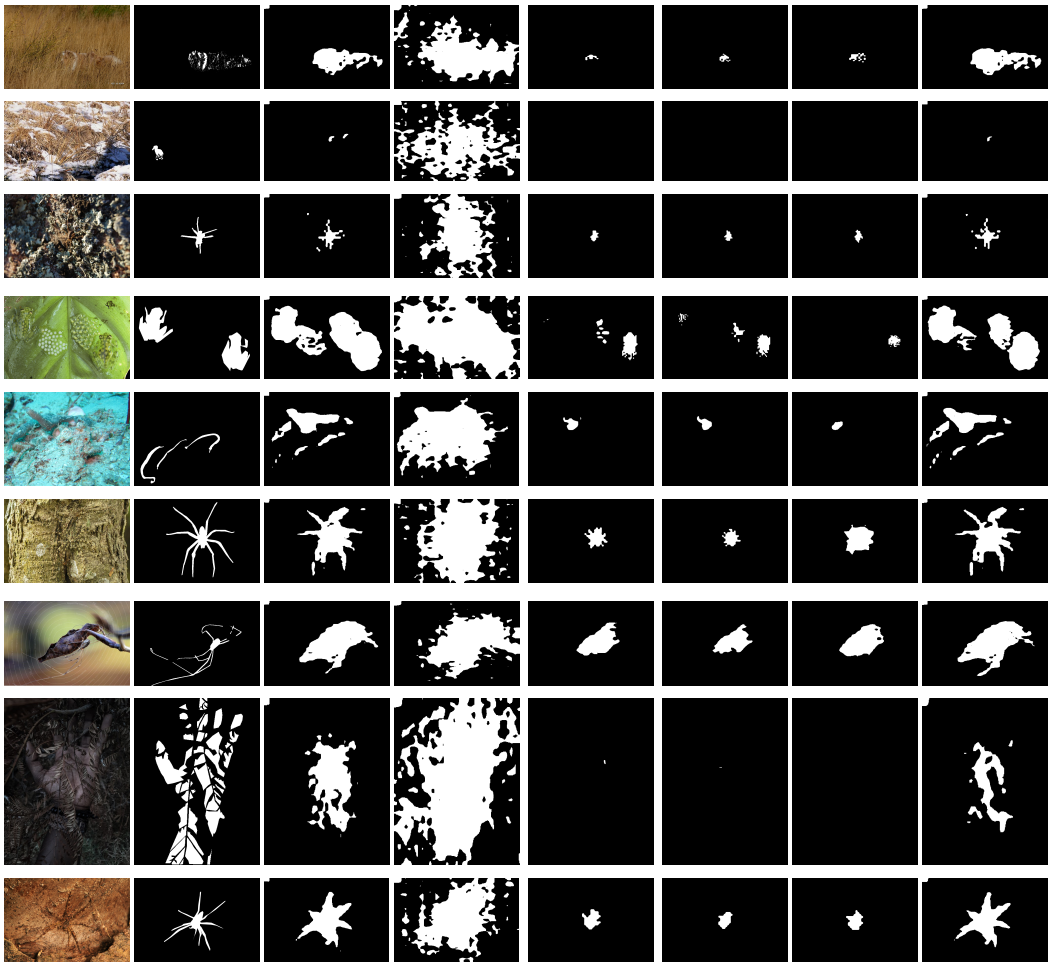


Figure 5: Supplementary failure-case analysis under the same UCOD-DPL carrier, illustrating representative challenging regimes under structurally unreliable pseudo supervision.

Relative to No topology, our method improves weighted F-measure by 0.0197/0.0098/0.0104 and Dice by 0.0099/0.0199/0.0106 on CRAG, GlaS, and MoNuSeg, respectively, while reducing BME by 0.305/0.475/0.286. Ours achieves the highest weighted F-measure and Dice on all three datasets. MATCH and SATLoss obtain lower BME on CRAG and GlaS, respectively, but with substantially lower segmentation quality. These results provide matched transfer evidence beyond UCOD while again showing that feature-wise topological correspondence and foreground prediction quality are complementary rather than interchangeable. They should not be interpreted as universal superiority across dense-prediction domains.

D.4 Sensitivity to the Main Hyperparameters

The topology-weight sweep, including the extended settings $\lambda_{\text{topo}} = 4$ and 8, is reported directly in Table 5 of the main text. Here, we evaluate the remaining topology-specific controls using one-factor-at-a-time studies on NC4K under the DINOv2 UCOD-DPL setting. In each sweep, only one parameter is changed while all others are fixed to their default values.

Table 8 evaluates the threshold-grid size M and the number of retained intervals k. Both lower and higher settings remain better than No topology in task-level performance, while the default $M = 16$ and $k = 12$ give the strongest MAE and F_β^w among the tested values. The smaller Betti-counting errors obtained by several non-default settings again show that lower topology-side count discrepancy does not necessarily imply stronger foreground prediction.

Table 8: Sensitivity to threshold discretization and retained-event number. All other hyperparameters are fixed to their default values.

Parameter	Setting	MAE ↓	F_β^w ↑	β_0 ↓	β_1 ↓
M	14	0.0551	0.7239	3.4948	2.1718
	16 (default)	0.0510	0.7408	3.5911	2.2863
	20	0.0562	0.7195	3.4877	2.1958
k	8	0.0554	0.7227	3.4933	2.1691
	12 (default)	0.0510	0.7408	3.5911	2.2863
	15	0.0557	0.7215	3.4965	2.1718

Table 9: Sensitivity to the soft-threshold smoothing coefficient and reliability switching threshold. All other hyperparameters are fixed to their default values.

Parameter	Setting	MAE ↓	F_β^w ↑	β_0 ↓	β_1 ↓
δ	0.04	0.0596	0.7091	3.5724	2.2900
	0.08 (default)	0.0510	0.7408	3.5911	2.2863
	0.16	0.0623	0.6998	3.5120	2.3793
τ_{auto}	0.45	0.0555	0.7222	3.4916	2.1682
	0.60 (default)	0.0510	0.7408	3.5911	2.2863
	0.75	0.0557	0.7216	3.4822	2.1662

Table 9 shows that δ is the clearest task-level sensitivity among the tested controls. Making the soft threshold either sharper ($\delta = 0.04$) or more diffuse ($\delta = 0.16$) reduces F_β^w to 0.7091 and 0.6998, respectively. By contrast, varying τ_{auto} to 0.45 or 0.75 retains task-level performance above the No-topology reference, indicating substantially weaker sensitivity to the reliability switching threshold.

Table 10: Sensitivity to spatial-branch attenuation and relative-cap control. All other hyperparameters are fixed to their default values.

Parameter	Setting	MAE ↓	F_β^w ↑	β_0 ↓	β_1 ↓
a_{sp}	0.175	0.0555	0.7222	3.5013	2.1703
	0.35 (default)	0.0510	0.7408	3.5911	2.2863
	0.70	0.0557	0.7216	3.4771	2.1771
c_{cap}	0.125	0.0556	0.7220	3.4754	2.1781
	0.25 (default)	0.0510	0.7408	3.5911	2.2863
	0.50	0.0556	0.7216	3.5127	2.1779

Table 10 evaluates the spatial-branch attenuation a_{sp} and the relative cap c_{cap}. Both parameters are stable over the tested ranges: all perturbed settings retain $F_\beta^w \geq 0.7216$ and MAE ≤ 0.0557, remaining better than No topology ($0.7129/0.0590$). The default settings nevertheless provide the strongest task-level performance. Overall, five of the six controls remain robust over the tested ranges, whereas the smoothing coefficient δ is the primary task-level sensitivity.

D.5 Controlled Structural Corruption

To directly test robustness under different types of structural unreliability, we further construct controlled corruptions on the cached training pseudo masks. We consider four deterministic corruption types: *breaks*, which cut a narrow foreground path; *holes*, which remove an interior foreground region; *merges*, which bridge separated components; and *blobs*, which add an isolated foreground component. The corruption is applied offline to the cached pseudo masks with severity 0.12 and seed 3221. The 4,040 training samples, RGB images, cached features, carrier, backbone, task loss, optimizer, schedule, and 25-epoch budget remain fixed, and no ground-truth mask is used during training.

A corruption is accepted only when it produces the intended Betti-direction change for that operator; otherwise the original mask is retained and no sample is removed. The resulting valid/eligible statistics are shown below.

Table 11: Eligibility of the four synthetic structural corruptions.

Type	Valid / Eligible (%)	Mean defining change
Breaks	3311 / 4031 (82.1)	$\Delta\beta_0 = +1.410$
Holes	1386 / 4031 (34.4)	$\Delta\beta_1 = +1.000$
Merges	2048 / 2048 (100.0)	$\Delta\beta_0 = -1.037$
Blobs	1858 / 2377 (78.2)	$\Delta\beta_0 = +1.000$

Within each corruption condition, No topology, MATCH, and Ours receive the same modified pseudo-mask cache and are evaluated on all 4,121 NC4K samples. Here, MATCH denotes the external topology-supervision baseline, rather than the internal *Match* proxy mode in Table 4. BME is computed against the NC4K ground truth using the same protocol as in App. C.3. The original condition means no added synthetic corruption, rather than clean supervision.

Table 12: Results under controlled structural corruption. Each entry reports F_β^w/MAE/BME.

Condition	No topology	MATCH	Ours
Breaks	0.7770 / 0.0526 / 8.509	0.5996 / 0.0876 / 5.156	0.7919 / 0.0461 / 8.333
Holes	0.7618 / 0.0513 / 7.822	0.2081 / 0.1338 / 5.134	0.7710 / 0.0472 / 7.734
Merges	0.7040 / 0.0604 / 9.190	0.5874 / 0.0900 / 5.051	0.7109 / 0.0576 / 8.879
Blobs	0.7346 / 0.0552 / 7.673	0.6458 / 0.0809 / 5.114	0.7398 / 0.0528 / 7.520
Mean	0.7444 / 0.0549 / 8.298	0.5102 / 0.0981 / 5.114	0.7534 / 0.0509 / 8.117

Ours improves F_β^w, MAE, and BME over No topology under all four corruption types. Averaged across the four conditions, F_β^w/MAE/BME change from 0.7444/0.0549/8.298 to 0.7534/0.0509/8.117. MATCH achieves lower BME but substantially worse localization quality, with mean $F_\beta^w = 0.5102$ and MAE $= 0.0981$. Thus, under controlled structural corruption, Ours is the only compared formulation that improves both task-level quality and BME over No topology in every condition.

D.6 Ranked COD10K Hard-Case Audit

For diagnosis rather than benchmark estimation, we rank all 2,026 COD10K test samples by the per-image MAE of our method and define the top 10% (203 samples) as the hard subset. Relative to the remaining samples, the no-topology carrier exhibits lower median binary precision (0.329 vs. 0.648), a larger median predicted-area ratio (0.211 vs. 0.095), and a higher median β_0 error (6 vs. 2), whereas the median GT-area ratio differs much less (0.070 vs. 0.059). These statistics indicate that foreground over-expansion and larger connected-component discrepancies, rather than object size alone, characterize the hard subset.

Across this subset, Ours improves over the fixed no-topology carrier on 66.0% of samples by MAE and 67.0% by per-image binary F1. Median MAE improves from 0.154 to 0.133, while median binary F1 improves from 0.449 to 0.485. Nevertheless, the ranked audit also exposes several remaining failure modes. Thin, weakly connected, and multi-part targets are frequently fragmented or over-expanded, while several topology-supervision baselines produce overly compact or nearly empty masks. In particular, topology-side correction cannot fully repair severe foreground-support errors once the underlying prediction substantially over-covers background.

Fig. 6 provides representative examples of these ranked failure modes, complementing the broader supplementary failure-case analysis in Appendix D.2. The ranked hard cases also delimit the current scope of the proposed reformulation. Although the matched histology experiments in Appendix D.3 provide evidence beyond UCOD, the current validation remains limited to UCOD and histology segmentation benchmarks, and transfer to broader dense-prediction domains and pseudo-supervision regimes remains to be established. Moreover, when recoverable structural evidence is extremely weak, thin, fragmented, or dominated by foreground-support errors, complete structural recovery remains

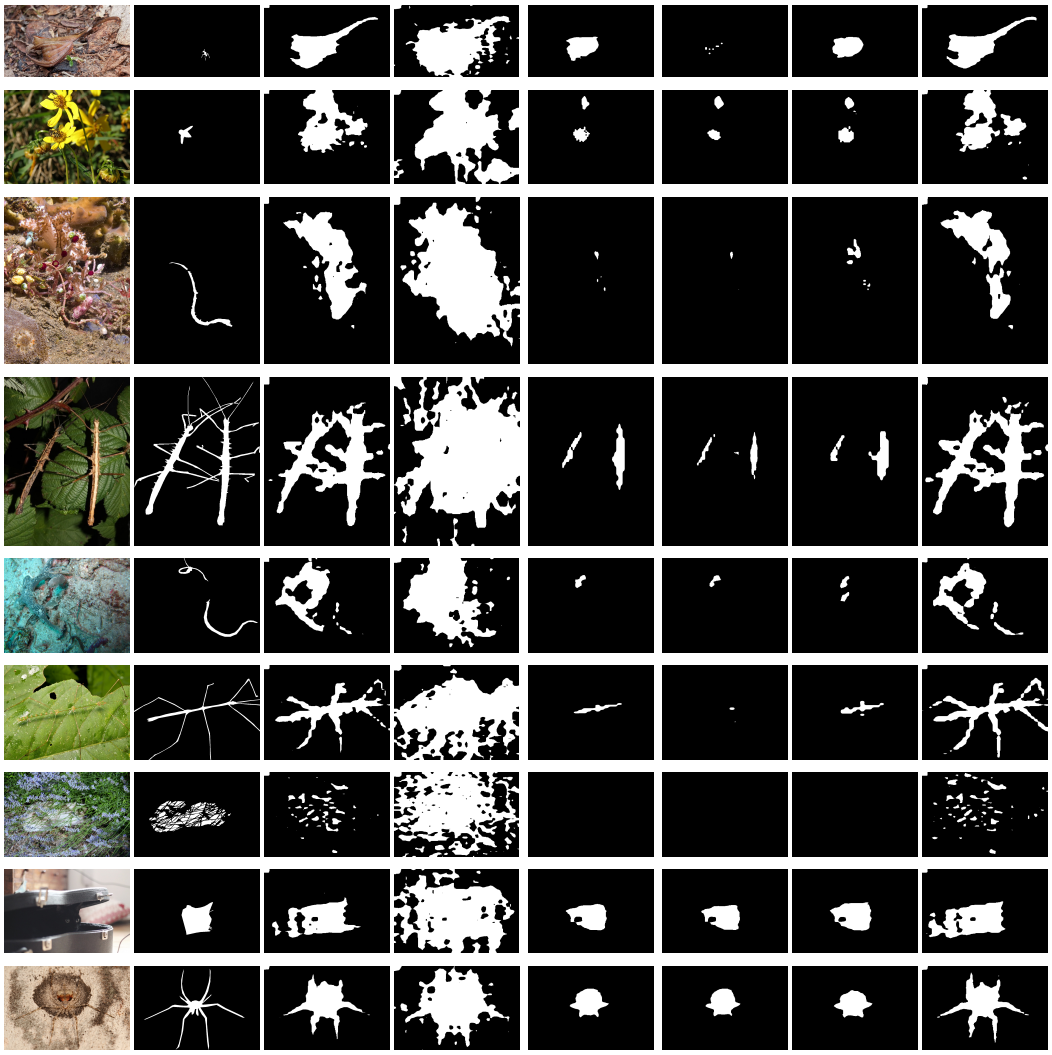


Figure 6: Representative ranked hard cases from the COD10K top-10% hard subset. Rows correspond, from top to bottom, to ranks #80, #84, #97, #112, #114, #129, #136, #145, and #150. The examples include severe foreground over-expansion, thin or weakly connected targets, branched and multi-part structures, and compact targets with limited foreground support.

difficult. Finally, the formal properties in Appendix B are local properties of the present formulation and implementation pathway rather than global guarantees on the overall training dynamics.

D.7 Broader Impacts

This work is methodological research on topology supervision under noisy pseudo supervision. Improving structural reliability in dense prediction may benefit annotation-limited settings where dense labels are expensive and structural consistency matters, such as scientific image analysis and conservation-oriented monitoring. However, more reliable camouflaged object detection and structure-aware visual analysis may also support sensitive uses, including surveillance, military-related scene analysis, or monitoring applications. The method is evaluated on existing benchmark datasets and is not deployment-ready; downstream use should require task-specific validation, human oversight, and applicable privacy or legal review. The released code is a minimal topology component rather than a deployment-ready recognition system.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the paper's scope, namely trainable topology supervision under structurally unreliable pseudo supervision, and summarize the two proposed components together with the UCOD validation setting; see Abstract, Sec. 1, and the contribution list.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses empirical and theoretical limitations in the appendix, including the difficult regimes highlighted by the supplementary failure-case analysis and the fact that the formal properties are local rather than global guarantees; see Appendix D.2 and Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper states the assumptions explicitly and provides formal proofs in the appendix, including the standing assumptions, local differentiability results, conservativeness properties, projection geometry, cap control, and surrogate backward exactness; see Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper discloses the method formulation, evaluation protocol, compared methods, datasets, and run-specific implementation details needed to reproduce the main reported results; see Secs. 3–4 and Appendices A and C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide code for the proposed topology component together with reproduction instructions and detailed method/configuration descriptions. The paper further specifies the datasets, carrier, comparison protocol, and run-specific settings needed to reproduce the reported experiments; see Secs. 3–4 and the appendix.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the datasets, data splits, comparison protocol, optimization settings, evaluation metrics, and implementation platform in the main text, with additional run-specific details in the appendix; see Sec. 4.1 and Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

Justification: The paper does not report repeated-run error bars or formal significance tests. We therefore do not claim statistical significance. The empirical evidence is based on controlled same-carrier comparisons across four benchmarks and targeted ablations under a fixed protocol.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper reports the hardware platform, GPU memory, and approximate training and evaluation time in the appendix, together with the main training setup in Sec. 4.1; see Sec. 4.1 and Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research is conducted on existing benchmark datasets for method development and evaluation, and we are not aware of any aspect of the work that conflicts with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper includes a brief broader-impacts discussion covering both potential benefits of more reliable dense prediction under limited annotation and potential risks in sensitive downstream uses such as surveillance-related visual analysis; see Appendix D.7.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release high-risk generative models, scraped datasets, or other assets that require special misuse safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper credits the original datasets, the UCOD-DPL carrier, and the existing comparison methods through citations in the main text and appendix. The assets used in this work are drawn from publicly available academic resources, and we use them in accordance with their publicly stated licenses or research usage conditions.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The paper releases code for the proposed topology component. The repository includes documentation and instructions for using the released implementation. The paper does not introduce a new dataset, benchmark, or trained model asset.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The work does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The work does not involve human subjects research and therefore does not require IRB approval.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: Large language models are not part of the core methodology, experiments, or theoretical development of this work.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.