UniPro: Unified Multi-Mode Medical Image Segmentation from 2D Images to 3D Volumes via Propagation

Bangwei Guo[1] **Yunhe Gao**[2] **Meng Ye**[3] **Yang Zhou**[1]
Difei Gu[1] **Guoning Zhang**[1] **Leon Axel**[4] **Dimitris N. Metaxas**[1]

[1]Rutgers University [2]Stanford University
[3]The University of Texas at Arlington [4]New York University

Abstract

Medical image segmentation remains fragmented along two axes: segmentation paradigms and data dimensionality. Existing methods are typically developed separately for semantic, in-context, and interactive segmentation, and are further specialized to either native 2D images or 3D volumetric data. In clinical practice, however, segmentation workflows take many forms: a case may be initialized by semantic prediction, reference-guided segmentation, or user interaction. Regardless of how it begins, fine-grained refinement is naturally performed on 2D views; for volumetric scans, such 2D edits must propagate coherently to the rest of the volume. We present **UniPro**, a **uni**fied model that bridges segmentation paradigms and data dimensionality, using **pro**pagation to extend 2D segmentation to 3D volumes. Our key insight is that volumetric propagation and in-context segmentation share the same reference-conditioned prediction mechanism, differing only in whether the reference image–mask pairs come from other cases or from previously segmented neighboring slices. Building on this view, UniPro supports semantic, in-context, interactive, and propagation-based segmentation within a single slice-based framework, using class priors, reference exemplars, user clicks, and neighboring-slice predictions as mode-specific conditioning inputs. To improve propagation reliability, UniPro further incorporates bidirectional and 3D supervision to regularize slice-wise propagation beyond per-slice losses. Extensive experiments across diverse modalities and anatomies show that UniPro achieves strong performance across all segmentation settings, enabling annotation-efficient 3D segmentation from sparse 2D initialization and reducing slice-by-slice correction effort.

1 Introduction

Medical image segmentation is a cornerstone of quantitative analysis and clinical decision-making. By delineating anatomical structures and pathological regions, it supports a wide range of downstream applications, including structural assessment [6, 49], disease diagnosis [43, 23], and agent-driven clinical analysis workflows [34, 37]. Recent advances in deep learning have substantially improved segmentation performance across diverse organs, imaging modalities, and datasets [25, 39]. Despite this progress, medical image segmentation remains fragmented along two axes: segmentation paradigms and data dimensionality. Existing methods are typically developed separately for semantic, in-context, and interactive segmentation [8, 27, 28, 39], and are further specialized to either native 2D images or 3D volumetric data [19, 20, 52, 40, 54]. As a result, advances in individual settings have not yet led to a unified framework that naturally supports the full spectrum of medical segmentation workflows.

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

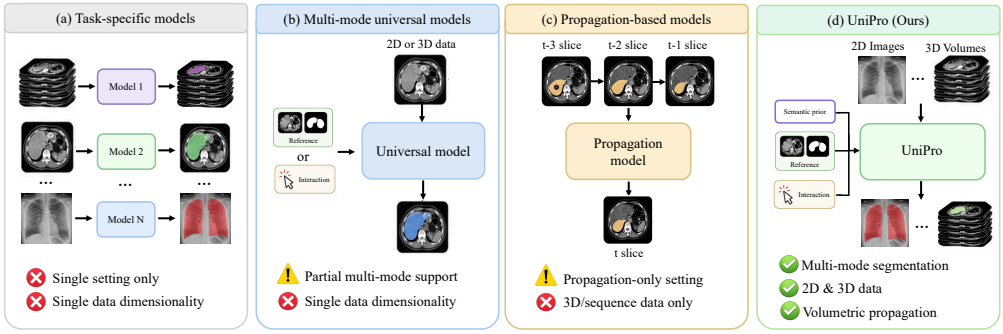


Figure 1: Comparison of medical image segmentation paradigms. Existing task-specific, universal, and propagation-based models cover only part of the design space, with limitations in segmentation modes, data dimensionality, or propagation capability. UniPro unifies multi-mode segmentation, native 2D and 3D data, and volumetric propagation in a single slice-based framework.

This fragmentation is also misaligned with real-world clinical practice. Clinicians work with both native 2D images and volumetric scans, for example reading chest X-rays and dermatology photographs in their original 2D form while reviewing CT and MRI studies slice-by-slice in 3D volumes [43]. A case may be initialized by semantic prediction, reference-guided segmentation, or user interaction [24]. Regardless of the initialization source, fine-grained refinement is naturally performed on 2D views, where local boundaries and subtle anatomical details are easier to inspect and correct [40]. For volumetric data, such 2D edits can be extended to the remaining slices through propagation, which sequentially predicts segmentation masks using initialized or previously segmented slices as references while preserving local editability. Together, these observations suggest that medical segmentation is inherently multi-mode and cross-dimensional, and that flexible initialization combined with editable volumetric propagation can naturally bridge these dimensions.

To address this challenge, we propose **UniPro**[1], a **uni**fied model for multi-mode medical image segmentation, with **pro**pagation as the bridge between 2D and 3D (Figure 1). Our key insight is that volumetric propagation and in-context segmentation share the same reference-conditioned prediction mechanism: both predict a target mask conditioned on image–mask references, differing only in whether the references come from other cases or from previously segmented neighboring slices within the same volume. Within a single slice-based framework, UniPro additionally supports semantic and interactive segmentation through class priors and user clicks, thereby unifying class-level, reference-based, click-based, and neighboring-slice conditioning without separate task-specific architectures. Reliable propagation remains challenging because sequential predictions can accumulate drift along the propagation chain, and slice-wise optimization alone does not sufficiently constrain the volumetric consistency. To improve propagation reliability, UniPro incorporates bidirectional and 3D supervision to regularize slice-wise propagation beyond per-slice losses. Together, these components enable UniPro to deliver coherent 3D segmentation from sparse 2D initialization while remaining locally editable—matching the multi-mode, cross-dimensional nature of clinical workflows.

In summary, our contributions are: **(1)** To our knowledge, **UniPro** is the first unified slice-based framework that jointly supports semantic, in-context, interactive, and propagation-based segmentation across native 2D images and 3D volumes. **(2)** We formulate volumetric propagation and in-context segmentation under a shared reference-conditioned prediction mechanism, allowing propagation to reuse the same conditioning pathway over neighboring slices. **(3)** We incorporate bidirectional and 3D supervision to regularize slice-wise propagation beyond per-slice losses, improving reliability for volumetric medical segmentation. **(4)** We evaluate UniPro on 14 medical imaging datasets across diverse modalities, anatomies, and segmentation settings, demonstrating strong performance and generalization, together with annotation-efficient 3D segmentation from sparse 2D initialization.

[1]Project repository: https://github.com/bangwayne/UniPro.

2 Related Work

2.1 Universal Medical Image Segmentation

Recent studies have explored unified medical segmentation frameworks that support multiple segmentation paradigms within a single model. Existing approaches typically achieve partial unification, such as combining semantic and in-context segmentation [21], semantic and interactive segmentation [22, 44, 18], or in-context and interactive segmentation [50, 55]. However, these frameworks are usually designed for a subset of segmentation modes and are often confined to a single data dimensionality. In contrast, UniPro aims to unify semantic, in-context, interactive, and propagation-based segmentation while supporting both native 2D images and 3D volumes through a slice-based framework.

2.2 2D and 3D Medical Image Segmentation

Medical image segmentation methods are commonly developed for either native 2D images or 3D volumetric scans. 2D models, such as nnU-Net [27], MedSAM [39], UniverSeg [8], and SAM-Med2D [16], are naturally suited for native 2D modalities such as X-ray, dermoscopy, ultrasound, and endoscopy, and can also be applied to 3D volumes by processing them slice by slice. However, when used for volumetric data, independent 2D prediction does not explicitly model cross-slice continuity and often requires dense per-slice prediction or correction to obtain coherent 3D masks. In contrast, 3D models, such as UNETR [25], nnInteractive [28], and SAM-MED3D [48], capture volumetric context and inter-slice dependencies, but are typically tied to volumetric inputs, often incur higher computational and memory costs than conventional 2D slice-based models, and provide limited flexibility for localized 2D slice-level editing. UniPro bridges this gap by retaining compatibility with native 2D images and slice-level editability, while using propagation to extend sparse 2D initialization and refinements into coherent 3D predictions.

2.3 Propagation-Based Models

Propagation-based models segment volumes sequentially by propagating mask information from annotated or previously segmented slices to adjacent slices. Compared with full-volume 3D processing, propagation-based models retain slice-level modeling and local editability through sequential slice-wise prediction [12]. Correspondence- and registration-based approaches [53, 2, 7] extend a single annotated slice to a full volume by learning inter-slice correspondences or displacement fields. Recent medical propagation models [40, 12, 3, 11] are often inspired by video segmentation models such as SAM2 [42]. However, they mainly operate on 3D volumes or video-like sequences and are usually designed for propagation-centric, interaction-based segmentation, rather than jointly supporting native 2D images and multiple segmentation modes. Moreover, most propagation methods lack explicit regularization for anatomical continuity in medical volumes, making sequential predictions prone to error accumulation and cross-slice drift [3]. In contrast, UniPro integrates propagation into a unified multi-mode framework by treating neighboring-slice propagation as reference-conditioned prediction, and further regularizes slice-wise propagation with bidirectional and 3D supervision.

3 Method

3.1 Overview

Figure 2 illustrates the UniPro framework. UniPro is built on a shared 2D encoder–decoder backbone that supports four segmentation settings: semantic, in-context, interactive, and propagation-based segmentation through different forms of conditioning applied to the same backbone. Specifically, UniPro supports the following settings:

- **Semantic segmentation**: the model predicts the mask from the input image conditioned on class-level prior queries.
- **In-context segmentation**: the model conditions on reference image–mask pairs for reference-guided prediction.

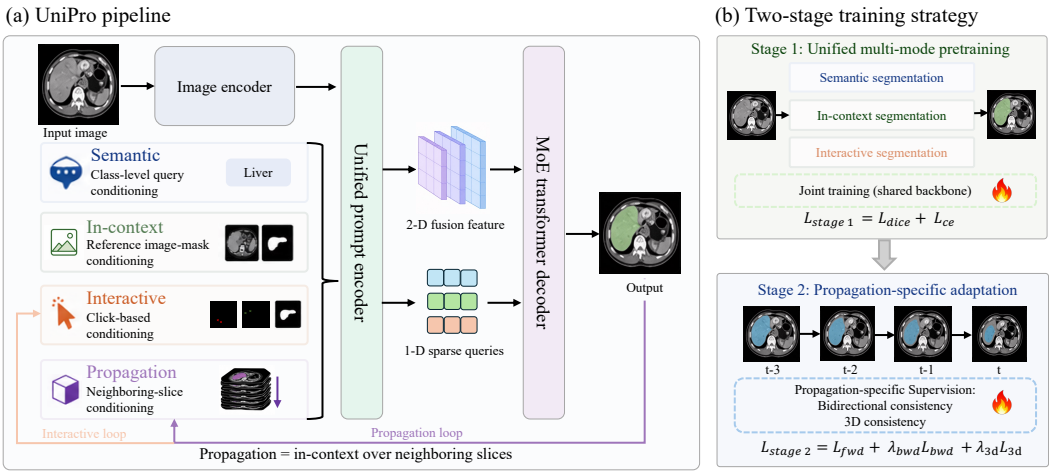


Figure 2: **UniPro overview. (a)** A unified slice-based segmentation pipeline for semantic, in-context, interactive, and propagation-based segmentation. UniPro uses a shared image encoder and decoder, with different settings instantiated through mode-specific conditioning inputs. Conditioning signals are converted into dense 2-D fusion features and sparse 1-D queries, which are jointly fed to the MoE Transformer decoder. Notably, propagation is formulated as in-context segmentation over neighboring slices, enabling coherent volumetric segmentation. **(b)** Two-stage training strategy. Stage 1 jointly pretrains semantic, in-context, and interactive segmentation with the shared backbone. Stage 2 further adapts the model to propagation with bidirectional and 3D consistency supervision.

- **Interactive segmentation**: the model takes user clicks for mask prediction and iterative refinement.
- **Propagation for 3D volumes**: the model performs slice-wise propagation from a seed slice (initialized by any of the above modes), treating each query slice as in-context segmentation conditioned on previously segmented neighboring slices.

To realize these capabilities within a single framework, UniPro is trained in two stages. In Stage 1, the shared backbone is jointly optimized for semantic, in-context, and interactive segmentation, learning a unified representation under diverse conditioning forms. In Stage 2, the same backbone is further trained for volumetric propagation with propagation-specific supervision that improves slice-wise propagation reliability beyond per-slice losses. Detailed architectures and training procedures are provided in the Appendix A.3.

3.2 Propagation as Reference-Conditioned In-Context Segmentation

Anchor-pinned reference set construction. Given a 3D volume $\mathbf{V} \in \mathbb{R}^{D\times 3\times H\times W}$ with D axial slices, UniPro produces the full segmentation by bidirectionally propagating from a seed slice z^*, defined as the median axial index of the target organ. The accompanying seed mask $\hat{\mathbf{M}}_{z^*}$ may originate from semantic, in-context, or interactive segmentation, or from ground-truth annotation. For notational simplicity, we describe only the forward direction. At each step $q > z^*$, propagation is cast as a reference-conditioned in-context prediction problem, where the query slice $\mathbf{I}_q \in \mathbb{R}^{3\times H\times W}$ is segmented using a dynamically maintained reference set

$$\mathcal{R}_q = \{(\mathbf{I}_{z^*}, \hat{\mathbf{M}}_{z^*})\} \cup \{(\mathbf{I}_{q-j}, \hat{\mathbf{M}}_{q-j})\}_{j=1}^{r-1}, \tag{1}$$

with $r = 5$ by default. The seed pair is permanently anchored in $\mathcal{R}_q$ and combined with the $r-1$ most recently propagated neighbors, providing a persistent, well-localized reference that mitigates cumulative semantic drift over long slice ranges. For brevity, we hereafter stack the reference images and masks in $\mathcal{R}_q$ as $\mathbf{I}_{\text{ref}} \in \mathbb{R}^{r\times 3\times H\times W}$ and $\mathbf{M}_{\text{ref}} \in \mathbb{R}^{r\times 1\times H\times W}$.

Dense reference-to-query feature matching. Given a query slice $\mathbf{I}_q$ together with its reference set $\mathcal{R}_q$, a shared image encoder extracts spatial key features from both sides:

$$\mathbf{K}^{\text{query}} = \text{Encoder}(\mathbf{I}_q) \in \mathbb{R}^{C\times hw}, \qquad \mathbf{K}^{\text{ref}} = \text{Encoder}(\mathbf{I}_{\text{ref}}) \in \mathbb{R}^{C\times rhw}. \tag{2}$$

In parallel, a lightweight mask encoder embeds the reference image–mask pairs into context-aware value features:

$$\mathbf{V}^{\mathrm{ref}} = \mathrm{MaskEncoder}\big(\mathrm{Concat}[\mathbf{I}_{\mathrm{ref}}, \mathbf{M}_{\mathrm{ref}}]\big) \in \mathbb{R}^{C\times rhw}. \tag{3}$$

Since $\mathbf{K}^{\mathrm{ref}}$ and $\mathbf{K}^{\mathrm{query}}$ reside in a common embedding space, pixel-wise correspondences between reference and query locations can be established directly. Formally, for a similarity function $c : \mathbb{R}^C \times \mathbb{R}^C \to \mathbb{R}$, we compute the affinity matrix and normalize it column-wise via softmax:

$$\mathbf{A}_{i,j} = c\big(\mathbf{K}^{\mathrm{ref}}_{:,i}, \mathbf{K}^{\mathrm{query}}_{:,j}\big), \qquad \mathbf{W}_{i,j} = \frac{\exp(\mathbf{A}_{i,j})}{\sum_n \exp(\mathbf{A}_{n,j})}, \tag{4}$$

where $\mathbf{A}, \mathbf{W} \in \mathbb{R}^{rhw\times hw}$. We instantiate c as a negative squared Euclidean similarity following Cheng et al. [15] (see Appendix A.1.3 for the explicit form). The normalized weights $\mathbf{W}$ then aggregate reference values into a query-aligned dense fusion feature carrying per-location semantic cues transferred from the reference set to the query:

$$\mathbf{F}_{\mathrm{fuse}} = \mathbf{V}^{\mathrm{ref}}\,\mathbf{W} \in \mathbb{R}^{C\times hw}. \tag{5}$$

Sparse foreground–background queries. To complement the dense fusion feature with object-level context, we condense the reference key features $\mathbf{K}^{\mathrm{ref}}$ into a small set of 1-D foreground–background queries $\mathbf{Q}^s \in \mathbb{R}^{n_s\times C}$ via masked average pooling. Here, "sparse" refers to this compact query set relative to the spatially dense 2-D feature map: the queries summarize object-level semantics, whereas the dense features preserve spatially aligned, location-specific details. The first $n_s/2$ queries summarize foreground semantics and the remaining half summarize background semantics. Let $\mathbf{M}^{\mathrm{pool}} \in [0,1]^{rhw}$ denote $\mathbf{M}_{\mathrm{ref}}$ downsampled and flattened to the key feature resolution, so that $\mathbf{M}^{\mathrm{pool}}_u$ indicates whether spatial location u is foreground. For the n-th query, we then define a binary selection weight $v^n_u \in \{0,1\}$ and compute

$$v^n_u = \begin{cases} 1, & n \le \frac{n_s}{2} \text{ and } \mathbf{M}^{\mathrm{pool}}_u \ge 0.5, \\ 1, & n > \frac{n_s}{2} \text{ and } \mathbf{M}^{\mathrm{pool}}_u < 0.5, \\ 0, & \text{otherwise}, \end{cases} \qquad \mathbf{Q}^s_{n,:} = \frac{\sum_{u=1}^{rhw} (\mathbf{K}^{\mathrm{ref}})^\top_{u,:}\, v^n_u}{\sum_{u=1}^{rhw} v^n_u} \in \mathbb{R}^C. \tag{6}$$

The sparse foreground–background queries $\mathbf{Q}^s$ and dense fusion features $\mathbf{F}_{\mathrm{fuse}}$ jointly condition the decoder to predict the slice-wise mask:

$$\hat{\mathbf{M}}_q = \mathrm{Decoder}(\mathbf{I}_q \mid \mathbf{F}_{\mathrm{fuse}},\, \mathbf{Q}^s) \in [0,1]^{H\times W}. \tag{7}$$

Stacking the predictions along the axial dimension yields the full 3D segmentation $\hat{\mathbf{V}} \in [0,1]^{D\times H\times W}$. Notably, this formulation unifies slice-wise propagation and in-context segmentation: both operate on a query image paired with a set of image–mask exemplars and invoke the same dense-plus-sparse prompting pipeline. The two modes differ only in how $\mathcal{R}_q$ is composed: from recently segmented neighboring slices for propagation, or from externally annotated exemplars for in-context segmentation, while the prediction mechanism remains unchanged.

3.3 Propagation and Consistency Objectives

Stage II propagation objective. To train the model for volumetric propagation, we sample a short slice queue $\mathcal{Q} = \{(\mathbf{I}_t, \mathbf{M}_t)\}_{t=1}^T$ of length $T = 8$, where $\mathbf{I}_t$ is the image at slice t and $\mathbf{M}_t$ is the corresponding binary ground-truth mask. Beyond standard slice-wise supervision, our Stage II objective addresses two challenges specific to propagation: (1) error accumulation along the propagation direction, mitigated by backward propagation with seed reconstruction; and (2) lack of global volumetric consistency across slices, addressed by a lightweight 3D verification loss.

Forward propagation loss. Starting from the seed slice, the model obtains an initial mask $\hat{\mathbf{M}}_1$ and propagates the segmentation to subsequent slices. Each propagated slice is supervised by a standard 2D segmentation loss:

$$\mathcal{L}_{2D}(\hat{\mathbf{M}}_t, \mathbf{M}_t) = \mathcal{L}_{dice}(\hat{\mathbf{M}}_t, \mathbf{M}_t) + \mathcal{L}_{ce}(\hat{\mathbf{M}}_t, \mathbf{M}_t), \tag{8}$$

where $\hat{\mathbf{M}}_t$ is the prediction at slice t. The forward propagation loss aggregates the per-slice losses across the queue, with an additional weight λ_{init} on the seed slice:

$$\mathcal{L}_{\text{fwd}} = \frac{1}{T}\left(\lambda_{\text{init}}\mathcal{L}_{2D}(\hat{\mathbf{M}}_1, \mathbf{M}_1) + \sum_{t=2}^{T} \mathcal{L}_{2D}(\hat{\mathbf{M}}_t, \mathbf{M}_t)\right). \tag{9}$$

Backward propagation and seed reconstruction. Forward-only propagation suffers from one-way cascading errors: a drift introduced at an early slice is silently inherited by all subsequent predictions with no corrective signal. We therefore additionally propagate from slice T back to the seed, letting $\tilde{\mathbf{M}}_t$ denote the backward prediction at slice t; the reconstructed seed $\tilde{\mathbf{M}}_1$ then acts as a cycle-consistency checkpoint, whose supervision gradient penalizes forward drift at its source. We apply the same slice-wise supervision during backward propagation, with an additional weight λ_{seed} on the reconstructed seed slice to emphasize cycle consistency:

$$\mathcal{L}_{\text{bwd}} = \frac{1}{T-2} \sum_{t=2}^{T-1} \mathcal{L}_{2D}(\tilde{\mathbf{M}}_t, \mathbf{M}_t) + \lambda_{\text{seed}} \mathcal{L}_{2D}(\tilde{\mathbf{M}}_1, \mathbf{M}_1). \tag{10}$$

The last slice is directly supervised by $\mathcal{L}_{\text{fwd}}$, while the first slice is explicitly reconstructed during backward propagation. The weight λ_{seed} strengthens this reconstruction supervision to help limit accumulated propagation drift. Since our model is memory-based, forward and backward propagation invoke the same reference-conditioned function $f(\mathbf{I}_q, \mathcal{R}_q)$ and differ only in the content of $\mathcal{R}_q$, so $\mathcal{L}_{\text{bwd}}$ introduces no additional parameters. We further provide a theoretical analysis of $\mathcal{L}_{\text{bwd}}$ in Appendix B.

3D supervision. While slice-wise supervision improves local accuracy, it does not explicitly enforce volumetric coherence: predictions can be locally plausible on each slice yet form an incoherent 3D shape overall. We therefore introduce a 3D verification loss that operates directly on stacked 2D predictions, without introducing 3D convolutions or other volumetric modules:

$$\hat{\mathbf{M}}^{3D} = [\hat{\mathbf{M}}_1, \hat{\mathbf{M}}_2, \ldots, \hat{\mathbf{M}}_T], \qquad \mathbf{M}^{3D} = [\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_T]. \tag{11}$$

We then compare the predicted queue against the ground-truth queue using

$$\mathcal{L}_{\text{3D-ver}} = \mathcal{L}^{3D}_{dice}(\hat{\mathbf{M}}^{3D}, \mathbf{M}^{3D}) + \mathcal{L}^{3D}_{ce}(\hat{\mathbf{M}}^{3D}, \mathbf{M}^{3D}), \tag{12}$$

where $\mathcal{L}^{3D}_{dice}$ is computed over the entire queue jointly rather than averaging per-slice Dice scores, so errors concentrated in a few slices are not washed out by well-predicted ones. This term encourages coherent anatomical structure across the queue and discourages abrupt disappearance, leakage, or unrealistic shape variation between neighboring slices.

Overall Objective. To prevent the model from forgetting the native 2D datasets (e.g., X-ray, dermoscopy) seen in Stage I, we keep the Stage I objective active during Stage II training. The total Stage II loss combines both:

$$\mathcal{L}_{\text{stage II}} = \mathcal{L}_{\text{stage I}} + \mathcal{L}_{\text{fwd}} + \lambda_{\text{bwd}} \mathcal{L}_{\text{bwd}} + \lambda_{\text{3d}} \mathcal{L}_{\text{3d}}. \tag{13}$$

This combined objective jointly improves local segmentation accuracy, propagation consistency in both directions, seed preservation after a forward-backward cycle, and global volumetric coherence across the full slice queue, while preserving the four-mode segmentation capability of the shared backbone. In our experiments, we set $\lambda_{\text{init}} = 2$, $\lambda_{\text{seed}} = 2$, $\lambda_{\text{bwd}} = 0.7$, and $\lambda_{\text{3d}} = 5$.

4 Experiments

4.1 Experimental Setup

Datasets. We train UniPro on 9 publicly available datasets [38, 30, 31, 6, 26, 1, 29, 10, 41, 17, 46], covering diverse imaging modalities, including CT, MRI, ultrasound, X-ray, and dermoscopy, as well as various clinical targets such as organs, tumors, and lesions. For evaluation, we additionally include four external datasets, BTCV [33], ACDC [5], UW-SC [47], and BUS [51], to assess cross-dataset generalization, and one unseen-class 3D dataset, BraTS [4], to test adaptation to novel anatomical and pathological structures. Full dataset and preprocessing details are provided in Appendix C.1.

Baselines. We compare UniPro against three groups of baselines. For slice-based segmentation, we include semantic models nnU-Net (2D) [27], UniSeg [52], and Hermes [20], and in-context models UniverSeg [8], SegMIC [56], and Iris [21]. For volumetric segmentation, we report nnU-Net (3D) [27] as a fully supervised 3D upper-bound reference and nnInteractive [28] as a strong

Table 1: Propagation-based segmentation (3D Dice, %, ↑). All methods are initialized from a single positive click on one seed slice and propagated to the remaining slices.

Method	Internal					External		Unseen
	Abd-1K	CHAOS	M&Ms	LiTS	KiTS	BTCV	ACDC	BraTS
3D Upper Bound								
nnInteractive	76.38	83.04	78.58	49.76	63.08	67.44	79.87	51.25
General Pretrained Model								
SAM2	45.00	56.61	47.60	23.08	51.61	37.10	50.71	32.38
Propagation Model								
iSegFormer	83.00	82.27	82.55	51.46	68.77	64.78	77.05	42.25
PAM	85.59	86.41	79.69	60.83	63.20	77.49	79.06	39.85
MedSAM2	86.29	**90.64**	82.97	62.25	73.32	80.55	84.18	40.74
UniPro (prop)	**88.62**	89.70	**87.64**	**66.95**	**76.95**	**84.40**	**87.97**	**54.89**

interactive 3D reference. For propagation-based segmentation, we include SAM2 [42], PAM [12], MedSAM2 [40], and iSegFormer [35]. For UniPro itself, we report two inference modes: UniPro (no-prop), where each 2D slice is segmented independently under the corresponding semantic or in-context setting; and UniPro (prop), where only a seed slice is initialized by semantic, in-context or interactive segmentation and the remaining slices are segmented through the propagation mechanism, producing a coherent 3D segmentation from sparse 2D initialization.

Evaluation protocol. All slice-based baselines are trained on 2D slices extracted from our curated datasets under the same protocol as UniPro, ensuring a fair comparison at the slice level. With the exception of SAM2, the 3D and propagation baselines are trained on the same corresponding 3D datasets with matched preprocessing, optimization settings, and training budgets as UniPro, while preserving their method-specific input formats and inference interfaces. Performance is measured by the Dice coefficient: 2D Dice is computed per slice for native 2D datasets, and 3D Dice is computed over the entire volume formed by target-containing slices. Full implementation details are provided in the Appendix A.3.

4.2 Main Results on Interactive-to-Propagation Segmentation

Table 1 evaluates UniPro under a single-click propagation setting on 3D datasets, reflecting a practical annotation workflow where a user provides an initial click on one representative slice and the model completes the full volume automatically. UniPro (prop) achieves the best performance on 7 of 8 datasets, improving over the strongest propagation baseline MedSAM2 by 4.70 Dice points on LiTS (66.95% vs 62.25%) and 4.67 Dice points on M&Ms, and surpassing PAM and iSegFormer by larger margins on nearly all datasets. SAM2, despite large-scale natural image pretraining, performs poorly on medical volumes, indicating that natural-image pretraining alone does not transfer to volumetric medical segmentation. Notably, UniPro (prop) also outperforms the dedicated 3D interactive model nnInteractive, suggesting that a well-trained slice-based propagation can rival full-3D interactive approaches. UniPro further generalizes beyond the training distribution. On external datasets, it reaches 84.40% on BTCV and 87.97% on ACDC. Most strikingly, on BraTS (brain tumors, an anatomy unseen during training), UniPro achieves 54.89% compared to MedSAM2's 40.74%, a gain of 14.15 Dice points that suggests the reference-conditioned mechanism transfers beyond the anatomies seen during training. Together, these results demonstrate that UniPro delivers both reliable seed-slice interaction and robust volumetric propagation across domain shifts and novel anatomies, supporting accurate 3D segmentation from minimal user input. More statistical analysis of these results is provided in Appendix D.4.

4.3 Main Results on Semantic and In-Context Segmentation

Table 2 reports semantic and in-context segmentation across 3D and 2D datasets. On 3D datasets, UniPro (prop) achieves the best results on nearly all internal datasets in both semantic and in-context settings. In the semantic setting, UniPro (prop) reaches 87.73% on M&Ms and 89.49% on ACDC, surpassing the fully supervised 3D nnU-Net upper bound on both datasets despite using a 2D-based architecture. Propagation consistently improves over UniPro (no-prop) on six of the seven 3D datasets

Table 2: Semantic and in-context segmentation across internal and external datasets (Dice, %, ↑). All in-context models use one-shot inference; propagation is applied only to 3D volumes. nnU-Net (3D) is reported as a fully supervised upper bound. Best results in each block are in bold.

Method	3D Datasets							2D Datasets					
	Internal					External		Internal				External	
	Abd-1K	CHAOS	M&Ms	LiTS	KiTS	BTCV	ACDC	Breast	BKAI	Chest	ISIC	BUS	UW-SC
3D Upper Bound													
nnU-Net (3D)	88.45	90.15	85.47	63.14	81.28	86.02	88.59	–	–	–	–	–	–
Semantic segmentation													
nnU-Net (2D)	79.13	80.29	85.42	60.12	75.99	70.73	84.64	71.68	83.45	94.95	**86.76**	68.00	79.78
UniSeg	84.26	84.17	85.66	56.23	72.68	81.53	86.91	72.32	81.33	94.51	83.86	67.12	77.93
Hermes	86.43	86.25	86.15	60.62	75.17	84.21	86.59	**75.21**	**84.12**	**95.59**	85.33	**68.67**	82.31
UniPro (no-prop)	87.30	85.01	86.95	60.85	75.56	**84.36**	87.60	73.87	82.72	95.03	86.11	67.29	**82.57**
UniPro (prop)	**88.20**	**87.90**	**87.73**	**61.15**	**78.01**	83.59	**89.49**	–	–	–	–	–	–
In-context segmentation													
UniverSeg	64.21	67.61	63.38	45.42	60.70	58.93	59.81	56.90	73.91	94.88	79.60	45.23	68.31
SegMIC	74.12	72.60	82.91	48.35	61.56	72.77	78.14	64.18	77.53	**96.13**	81.20	68.39	80.60
Iris	85.22	84.31	84.77	59.10	74.29	79.94	83.51	**73.99**	**82.53**	95.40	**85.21**	**71.44**	82.57
UniPro (no-prop)	87.01	84.48	86.65	57.77	75.65	82.48	85.91	71.63	80.97	94.99	84.62	68.78	**84.92**
UniPro (prop)	**87.64**	**85.01**	**87.31**	**61.15**	**76.37**	**83.18**	**88.44**	–	–	–	–	–	–

Table 3: Ablation studies on Stage-2 propagation training.

(a) Effect of bidirectional and 3D verification supervision on ACDC.

Variant	Semantic	In-context	Interactive
w/o $\mathcal{L}_{\text{bwd}}$ & $\mathcal{L}_{\text{3D}}$	87.78	86.13	85.61
w/o $\mathcal{L}_{\text{bwd}}$	89.10	87.39	87.28
w/o $\mathcal{L}_{\text{3D}}$	87.92	86.51	86.20
Full model	89.49	88.44	87.97

(b) Trade-off between 3D propagation gains and native 2D segmentation accuracy.

Training Strategy	ACDC	ISIC
Stage-1 only	86.73	87.54
Stage-2 w/o 2D interleaving	89.46	82.40
Stage-2 w/ 2D interleaving	89.49	86.11

(e.g., +2.89 Dice points on CHAOS, +2.45 on KiTS), demonstrating that single-slice initialization combined with propagation effectively leverages volumetric continuity. Similar gains appear in the in-context setting, where UniPro (prop) outperforms strong baselines (UniverSeg, SegMIC, Iris) on most 3D datasets, including a gain of 2.42 Dice points over Iris on Abd-1K (87.64% vs 85.22%) and a gain of 4.93 Dice points on ACDC (88.44% vs 83.51%). This confirms that propagation serves as an effective intra-volume extension of in-context segmentation. On 2D datasets, UniPro remains competitive but does not always achieve the best performance: while it leads on UW-SC, specialized 2D models such as Hermes and Iris achieve the best results on most internal datasets. This reflects a training trade-off introduced by Stage-2 propagation adaptation: as the model is further trained for sequential slice prediction, it slightly drifts from the optimal solution for static 2D segmentation. We further discuss this trade-off in Discussion.

4.4 Ablation Study

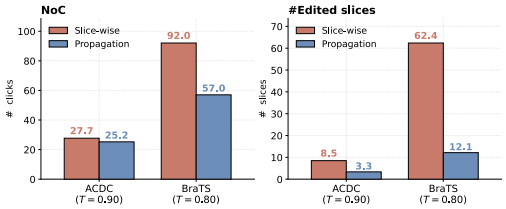
Effect of propagation supervision. Table 3(a) ablates the bidirectional consistency loss $\mathcal{L}_{\text{bwd}}$ and the 3D verification loss $\mathcal{L}_{\text{3D}}$ on ACDC under three propagation modes. Removing both reduces semantic Dice from 89.49% to 87.78%, with comparable drops on in-context (88.44% to 86.13%) and interactive (87.97% to 85.61%), confirming that propagation-specific supervision is essential beyond per-slice losses. Individually, $\mathcal{L}_{\text{3D}}$ contributes the larger share (a 1.57-Dice-point drop in semantic segmentation when removed) by enforcing cross-slice coherence that per-slice optimization cannot capture, while $\mathcal{L}_{\text{bwd}}$ adds a smaller but consistent gain (0.39 Dice points on top of $\mathcal{L}_{\text{3D}}$) by suppressing one-way drift. The two are complementary: $\mathcal{L}_{\text{3D}}$ regularizes global volumetric structure, $\mathcal{L}_{\text{bwd}}$ stabilizes the sequential propagation process.

Effect of interleaved Stage-2 training. We study the trade-off between learning 3D propagation and preserving native 2D segmentation ability. As shown in Table 3(b), Stage-2 training substantially improves ACDC (86.73% to 89.46%) but degrades ISIC by 5.14 Dice points (87.54% to 82.40%)

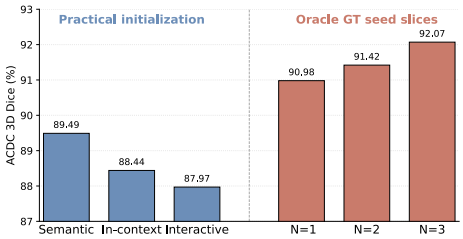
when trained only on propagation samples, indicating partial forgetting of 2D capability. Interleaving 2D slice-level batches during Stage-2 recovers most of the lost 2D performance (ISIC 82.40% to 86.11%) while leaving ACDC essentially unchanged (89.46% to 89.49%). However, ISIC remains 1.43 Dice points below the Stage-1 only model, suggesting an inherent trade-off between volumetric propagation and native 2D segmentation. A complementary ablation further supports the benefit of combining Stage-I pretraining with Stage-II propagation training: the full pipeline outperforms either stage alone under GT-seed initialization (Appendix D.6).

Effect of flexible interactive correction. We evaluate whether propagation reduces annotation effort during interactive 3D segmentation. Starting from an empty mask, the user iteratively places clicks until the 3D Dice reaches a target threshold ($T = 0.90$ for ACDC, $T = 0.80$ for the unseen BraTS) or a click budget of 100 is exhausted. We compare two settings: a slice-wise baseline, where each click updates only the clicked slice, and propagation, where the slice refined by user clicks is added to an anchor pool and the volume is re-segmented through propagation from the updated anchors. As shown in Figure 3a, propagation reaches the target Dice with fewer clicks (27.7 vs 25.2 on ACDC; 92.0 vs 57.0 on BraTS) and substantially fewer edited slices (8.5 vs 3.3 on ACDC; 62.4 vs 12.1 on BraTS). The gain is most pronounced on BraTS, where fragmented tumor regions make slice-by-slice annotation inefficient: propagation reduces the number of edited slices by over 80% (62.4 to 12.1) and clicks by nearly 40% (92.0 to 57.0). These results show that corrected slices serve as effective propagation anchors, reducing the numbers of clicks and edited slices in the simulated annotation protocol. Full simulation details are provided in Appendix D.1.

Effect of seed initialization. We analyze how reference quality and quantity affect volumetric propagation on ACDC. Here N denotes the number of GT slices retained as anchors in the sliding reference window during propagation. As shown in Figure 3b, practical seeds (semantic, in-context, interactive) reach 87.97–89.49% Dice from a single seed slice, only 1–3 Dice points below the oracle GT single-seed setting (90.98%). Increasing the number of GT reference slices monotonically improves performance, from 90.98% at $N = 1$ to 92.07% at $N = 3$, indicating that UniPro can effectively exploit additional high-quality anchors when available. Together, these results show that propagation is robust to the choice of practical initialization and scales gracefully with stronger references.



(a) Annotation efficiency under interactive correction: UniPro reaches the target 3D Dice with fewer clicks and edited slices than slice-wise editing.



(b) Seed initialization on ACDC: practical seeds (semantic, in-context, interactive) yield competitive single-seed propagation; multiple GT reference slices further improve performance.

Figure 3: Ablation analysis of UniPro's 3D propagation behavior.

Robustness to seed-slice selection. On ACDC, we vary the seed location across five quantiles of the target extent and a randomly selected target-containing slice, using ground-truth seed masks to isolate the effect of location. Dice remains within 89.46–91.03%, with random initialization achieving 90.12% compared with 90.98% for the median seed. Central slices generally yield slightly better results, consistent with their typically larger target regions providing more informative references, whereas boundary seeds show only modest degradation. Overall, propagation remains effective across the evaluated seed locations under GT-mask initialization; detailed results are provided in Appendix D.3.

4.5 Visualization

We complement the quantitative evaluation with qualitative comparisons and success–failure analysis. In the pancreas example (Figure 5), UniPro maintains coherent segmentation across distant slices, whereas the compared baselines exhibit leakage, incomplete coverage, or loss of the target. Figure 6

extends this analysis to irregular and multi-component tumors in LiTS, KiTS, and BraTS, showing the seed slice, selected propagated slices at increasing absolute distances, and corresponding 3D ground-truth and predicted masks. In the successful examples, UniPro segments the dominant irregular tumors in LiTS and KiTS and a relatively compact lesion in BraTS, despite brain anatomy and brain tumors being absent from training. Failures include missed small or spatially separated components in LiTS and KiTS and incomplete segmentation of a heterogeneous, fragmented BraTS lesion. Local errors also remain in the successful examples. These observations suggest that propagation depends not only on geometric complexity or anatomical familiarity, but also on component size, appearance consistency, and cross-slice continuity. Detailed visualizations and analyses are provided in Appendix E.

5 Discussion

We have presented UniPro, a unified framework that connects multiple segmentation paradigms across native 2D and 3D data. UniPro is particularly suited to heterogeneous segmentation and volumetric annotation workflows that combine automatic or reference-guided initialization, interactive correction, and propagation. Its shared interface allows difficult slices to be corrected locally and the corrections propagated through the volume. Dedicated models may remain preferable for fixed single-task applications.

Limitations and future work. Several limitations remain and motivate future work. **Trade-offs between 2D and 3D.** Stage-2 propagation adaptation improves 3D segmentation but can reduce native 2D accuracy, which interleaved 2D training only partly mitigates. Compared with the 3D nnU-Net baseline, UniPro uses less peak inference memory but requires longer inference time, partly due to sequential propagation (Appendix D.5). Future work could explore better multi-task optimization and propagation strategies that allow greater parallelization. **Gap to fully supervised 3D models.** UniPro trails the fully supervised 3D nnU-Net on most datasets in the semantic setting. It processes ordered 2D slices without explicit spacing or physical-distance inputs. Although isotropic spacing is not required, large through-plane gaps can weaken cross-slice correspondence. Incorporating spatial information, stronger cross-slice integration, or lightweight volumetric components may help close this gap. **Irregular and multi-component targets.** UniPro imposes no explicit convexity or single-component constraint. However, single-click initialization may miss disconnected regions, and weak cross-slice correspondence can hinder propagation. Additional interaction or alternative initialization may improve coverage. Robustness to fragmented targets and evaluation on untested structures, such as the intestine, remain directions for future work. **Towards a medical segmentation foundation model.** Our training corpus remains modest relative to that of large-scale models such as SAM2 [42]. Expanding data diversity and model capacity, together with broader evaluation across domains, is an important direction for improving generalization while retaining the unified segmentation interface.

6 Conclusion

We present UniPro, a unified slice-based framework that supports semantic, in-context, interactive, and propagation-based medical image segmentation within a single model. By formulating volumetric propagation as a special case of reference-conditioned in-context prediction, UniPro bridges native 2D images and 3D volumes through a shared backbone, enabling coherent volumetric segmentation from sparse 2D initialization. Extensive experiments across diverse modalities, anatomies, and segmentation settings show that UniPro achieves strong performance and cross-dataset generalization, while reducing clicks and edited slices in simulated interactive volumetric annotation. We believe this work points to a promising direction toward scalable, unified medical segmentation models that generalize across tasks, modalities, and dimensions.

Acknowledgments

This research has been partially funded by research grants to D. Metaxas through NSF: 2310966, 2235405, 2212301, 2003874, 1951890 and NIH 2R01HL127661.

Competing interests. The authors declare no competing interests.

References

[1] Walid Al-Dhabyani, Mohammed Gomaa, Hussien Khaled, and Aly Fahmy. Dataset of breast ultrasound images. *Data in brief*, 28:104863, 2020.

[2] Delin An, Pengfei Gu, Milan Sonka, Chaoli Wang, and Danny Z Chen. Sli2vol+: Segmenting 3d medical images based on an object estimation guided correspondence flow network. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3624–3634. IEEE, 2025.

[3] Yunhao Bai, Boxiang Yun, Zeli Chen, Qinji Yu, Yingda Xia, and Yan Wang. Revsam2: prompt sam2 for medical image segmentation via reverse-propagation without fine-tuning. *arXiv preprint arXiv:2409.04298*, 2024.

[4] Ujjwal Baid, Satyam Ghodasara, Suyash Mohan, Michel Bilello, Evan Calabrese, Errol Colak, Keyvan Farahani, Jayashree Kalpathy-Cramer, Felipe C Kitamura, Sarthak Pati, et al. The rsna-asnr-miccai brats 2021 benchmark on brain tumor segmentation and radiogenomic classification. *arXiv preprint arXiv:2107.02314*, 2021.

[5] Olivier Bernard, Alain Lalande, Clement Zotti, Frederick Cervenansky, Xin Yang, Pheng-Ann Heng, Irem Cetin, Karim Lekadir, Oscar Camara, Miguel Angel Gonzalez Ballester, et al. Deep learning techniques for automatic mri cardiac multi-structures segmentation and diagnosis: is the problem solved? *IEEE transactions on medical imaging*, 37(11):2514–2525, 2018.

[6] Patrick Bilic, Patrick Christ, Hongwei Bran Li, Eugene Vorontsov, Avi Ben-Cohen, Georgios Kaissis, Adi Szeskin, Colin Jacobs, Gabriel Efrain Humpire Mamani, Gabriel Chartrand, et al. The liver tumor segmentation benchmark (lits). *Medical image analysis*, 84:102680, 2023.

[7] Adeleh Bitarafan, Mohammad Farid Azampour, Kian Bakhtari, Mahdieh Soleymani Baghshah, Matthias Keicher, and Nassir Navab. Vol2flow: Segment 3d volumes using a sequence of registration flows. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 609–618. Springer, 2022.

[8] Victor Ion Butoi, Jose Javier Gonzalez Ortiz, Tianyu Ma, Mert R Sabuncu, John Guttag, and Adrian V Dalca. Universeg: Universal medical image segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21438–21451, 2023.

[9] Victor M Campello, Polyxeni Gkontra, Cristian Izquierdo, Carlos Martin-Isla, Alireza Sojoudi, Peter M Full, Klaus Maier-Hein, Yao Zhang, Zhiqiang He, Jun Ma, et al. Multi-centre, multi-vendor and multi-disease cardiac segmentation: the m&ms challenge. *IEEE Transactions on Medical Imaging*, 40(12):3543–3554, 2021.

[10] Sema Candemir, Stefan Jaeger, Kannappan Palaniappan, Jonathan P Musco, Rahul K Singh, Zhiyun Xue, Alexandros Karargyris, Sameer Antani, George Thoma, and Clement J McDonald. Lung segmentation in chest radiographs using anatomical atlases with nonrigid registration. *IEEE transactions on medical imaging*, 33(2):577–590, 2013.

[11] Yuwen Chen, Zafer Yildiz, Qihang Li, Yaqian Chen, Haoyu Dong, Hanxue Gu, Nicholas Konz, and Maciej A Mazurowski. Accelerating volumetric medical image annotation via short-long memory sam 2. *IEEE Transactions on Medical Imaging*, 2025.

[12] Zifan Chen, Xinyu Nan, Jiazheng Li, Jie Zhao, Haifeng Li, Ziling Lin, Haoshen Li, Heyun Chen, Yiting Liu, Lei Tang, et al. Pam: a propagation-based model for segmenting any 3d objects across multi-modal medical images. *npj Digital Medicine*, 2025.

[13] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022.

[14] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. *Advances in neural information processing systems*, 34:11781–11794, 2021.

[15] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Joon-Young Lee, and Alexander Schwing. Putting the object back into video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3151–3161, 2024.

[16] Junlong Cheng, Jin Ye, Zhongying Deng, Jianpin Chen, Tianbin Li, Haoyu Wang, Yanzhou Su, Ziyan Huang, Jilong Chen, Lei Jiang, et al. Sam-med2d. *arXiv preprint arXiv:2308.16184*, 2023.

[17] Noel Codella, Veronica Rotemberg, Philipp Tschandl, M Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, et al. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). *arXiv preprint arXiv:1902.03368*, 2019.

[18] Yuhang Ding, Liulei Li, Wenguan Wang, and Yi Yang. Clustering propagation for universal medical image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3357–3369, 2024.

[19] Yuxin Du, Fan Bai, Tiejun Huang, and Bo Zhao. Segvol: Universal and interactive volumetric medical image segmentation. *arXiv preprint arXiv:2311.13385*, 2023.

[20] Yunhe Gao. Training like a medical resident: Context-prior learning toward universal medical image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11194–11204, 2024.

[21] Yunhe Gao, Di Liu, Zhuowei Li, Yunsheng Li, Dongdong Chen, Mu Zhou, and Dimitris N Metaxas. Show and segment: Universal medical image segmentation via in-context learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 20830–20840, 2025.

[22] Bangwei Guo, Meng Ye, Yunhe Gao, Bingyu Xin, Leon Axel, and Dimitris Metaxas. Verse: Integrating multiple queries as prompts for versatile cardiac mri segmentation. In *International Conference on Information Processing in Medical Imaging*, pages 373–387. Springer, 2025.

[23] Bangwei Guo, Xiao Chen, Boris Mailhe, Jia Yao, Yiqing Wang, Ankush Mukherjee, Yikang Liu, Zheyuan Zhang, Hang Yu, Terrence Chen, et al. Learning where to look: Anatomical grounding and guided attention for cardiac mri vision-language models. *arXiv preprint arXiv:2609.39899*, 2026.

[24] Bangwei Guo, Yunhe Gao, Meng Ye, Difei Gu, Yang Zhou, Leon Axel, and Dimitris N. Metaxas. K-prism: A knowledge-guided and prompt integrated universal medical image segmentation model. In *The Fourteenth International Conference on Learning Representations*, 2026.

[25] Ali Hatamizadeh, Yucheng Tang, Vishwesh Nath, Dong Yang, Andriy Myronenko, Bennett Landman, Holger R Roth, and Daguang Xu. Unetr: Transformers for 3d medical image segmentation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 574–584, 2022.

[26] Nicholas Heller, Niranjan Sathianathen, Arveen Kalapara, Edward Walczak, Keenan Moore, Heather Kaluzniak, Joel Rosenberg, Paul Blake, Zachary Rengel, Makinna Oestreich, et al. The kits19 challenge data: 300 kidney tumor cases with clinical context, ct semantic segmentations, and surgical outcomes. *arXiv preprint arXiv:1904.00445*, 2019.

[27] Fabian Isensee, Paul F Jaeger, Simon AA Kohl, Jens Petersen, and Klaus H Maier-Hein. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature methods*, 18(2):203–211, 2021.

[28] Fabian Isensee, Maximilian Rokuss, Lars Krämer, Stefan Dinkelacker, Ashis Ravindran, Florian Stritzke, Benjamin Hamm, Tassilo Wald, Moritz Langenberg, Constantin Ulrich, et al. nninteractive: Redefining 3d promptable segmentation. *arXiv preprint arXiv:2503.08373*, 2025.

[29] Stefan Jaeger, Alexandros Karargyris, Sema Candemir, Les Folio, Jenifer Siegelman, Fiona Callaghan, Zhiyun Xue, Kannappan Palaniappan, Rahul K Singh, Sameer Antani, et al. Automatic tuberculosis screening using chest radiographs. *IEEE transactions on medical imaging*, 33(2):233–245, 2013.

[30] A. Emre Kavur, N. Sinem Gezer, Mustafa Barış, Sinem Aslan, Pierre-Henri Conze, Vladimir Groza, Duc Duy Pham, Soumick Chatterjee, Philipp Ernst, Savaş Özkan, Bora Baydar, Dmitry Lachinov, Shuo Han, Josef Pauli, Fabian Isensee, Matthias Perkonigg, Rachana Sathish, Ronnie Rajan, Debdoot Sheet, Gurbandurdy Dovletov, Oliver Speck, Andreas Nürnberger, Klaus H. Maier-Hein, Gözde Bozdağı Akar, Gözde Ünal, Oğuz Dicle, and M. Alper Selver. CHAOS Challenge - combined (CT-MR) healthy abdominal organ segmentation. *Medical Image Analysis*, 69:101950, April 2021. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2020.101950. URL `http://www.sciencedirect.com/science/article/pii/S1361841520303145`.

[31] Ali Emre Kavur, M. Alper Selver, Oğuz Dicle, Mustafa Barış, and N. Sinem Gezer. CHAOS - Combined (CT-MR) Healthy Abdominal Organ Segmentation Challenge Data, April 2019. URL `https://doi.org/10.5281/zenodo.3362844`.

[32] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[33] Bennett Landman, Zhoubing Xu, J Igelsias, Martin Styner, T Langerak, and Arno Klein. Miccai multi-atlas labeling beyond the cranial vault–workshop and challenge. In *Proc. MICCAI Multi-Atlas Labeling Beyond Cranial Vault—Workshop Challenge*, volume 5, page 12, 2015.

[34] Songhao Li, Jonathan Xu, Tiancheng Bao, Yuxuan Liu, Yuchen Liu, Yihang Liu, Lilin Wang, Wenhui Lei, Sheng Wang, Yinuo Xu, et al. A co-evolving agentic ai system for medical imaging analysis. *arXiv preprint arXiv:2509.20279*, 2025.

[35] Qin Liu, Zhenlin Xu, Yining Jiao, and Marc Niethammer. isegformer: interactive segmentation via transformers with application to 3d knee mr images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 464–474. Springer, 2022.

[36] Qin Liu, Zhenlin Xu, Gedas Bertasius, and Marc Niethammer. Simpleclick: Interactive image segmentation with simple vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22290–22300, 2023.

[37] Shengyuan Liu, Liuxin Bao, Qi Yang, Wanting Geng, Boyun Zheng, Chenxin Li, Wenting Chen, Houwen Peng, and Yixuan Yuan. Medsam-agent: Empowering interactive medical image segmentation with multi-turn agentic reinforcement learning. *arXiv preprint arXiv:2602.03320*, 2026.

[38] Jun Ma, Yao Zhang, Song Gu, Cheng Zhu, Cheng Ge, Yichi Zhang, Xingle An, Congcong Wang, Qiyuan Wang, Xin Liu, Shucheng Cao, Qi Zhang, Shangqing Liu, Yunpeng Wang, Yuhui Li, Jian He, and Xiaoping Yang. Abdomenct-1k: Is abdominal organ segmentation a solved problem? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6695–6714, 2022. doi: 10.1109/TPAMI.2021.3100536.

[39] Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. Segment anything in medical images. *Nature Communications*, 15(1):654, 2024.

[40] Jun Ma, Zongxin Yang, Sumin Kim, Bihui Chen, Mohammed Baharoon, Adibvafa Fallahpour, Reza Asakereh, Hongwei Lyu, and Bo Wang. Medsam2: Segment anything in 3d medical images and videos. *arXiv preprint arXiv:2504.03600*, 2025.

[41] Phan Ngoc Lan, Nguyen Sy An, Dao Viet Hang, Dao Van Long, Tran Quang Trung, Nguyen Thi Thuy, and Dinh Viet Sang. Neounet: Towards accurate colon polyp segmentation and neoplasm detection. In *International Symposium on Visual Computing*, pages 15–28. Springer, 2021.

[42] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[43] Dinggang Shen, Guorong Wu, and Heung-Il Suk. Deep learning in medical image analysis. *Annual review of biomedical engineering*, 19:221–248, 2017.

[44] Yiqing Shen, David Dreizin, Blanca Inigo, and Mathias Unberath. Protosam-3d: Interactive semantic segmentation in volumetric medical imaging via a segment anything model and mask-level prototypes. *Computerized Medical Imaging and Graphics*, 121:102501, 2025.

[45] Konstantin Sofiiuk, Ilya A Petrov, and Anton Konushin. Reviving iterative training with mask guidance for interactive segmentation. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 3141–3145. IEEE, 2022.

[46] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5(1): 1–9, 2018.

[47] University of Waterloo. Skin cancer detection. `https://uwaterloo.ca/vision-image-processing-lab/research-demos/skin-cancer-detection`.

[48] Haoyu Wang, Sizheng Guo, Jin Ye, Zhongying Deng, Junlong Cheng, Tianbin Li, Jianpin Chen, Yanzhou Su, Ziyan Huang, Yiqing Shen, et al. Sam-med3d: a vision foundation model for general-purpose segmentation on volumetric medical images. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

[49] Jakob Wasserthal, Hanns-Christian Breit, Manfred T Meyer, Maurice Pradella, Daniel Hinck, Alexander W Sauter, Tobias Heye, Daniel T Boll, Joshy Cyriac, Shan Yang, et al. Totalsegmentator: robust segmentation of 104 anatomic structures in ct images. *Radiology: Artificial Intelligence*, 5(5):e230024, 2023.

[50] Hallee E Wong, Jose Javier Gonzalez Ortiz, John Guttag, and Adrian V Dalca. Multiverseg: Scalable interactive segmentation of biomedical imaging datasets with in-context guidance. *arXiv preprint arXiv:2412.15058*, 2024.

[51] Moi Hoon Yap, Gerard Pons, Joan Marti, Sergi Ganau, Melcior Sentis, Reyer Zwiggelaar, Adrian K Davison, and Robert Marti. Automated breast ultrasound lesions detection using convolutional neural networks. *IEEE journal of biomedical and health informatics*, 22(4): 1218–1226, 2017.

[52] Yiwen Ye, Yutong Xie, Jianpeng Zhang, Ziyang Chen, and Yong Xia. Uniseg: A prompt-driven universal segmentation model as well as a strong representation learner. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 508–518. Springer, 2023.

[53] Pak-Hei Yeung, Ana IL Namburete, and Weidi Xie. Sli2vol: Annotate a 3d volume from a single slice with self-supervised learning. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 69–79. Springer, 2021.

[54] Yichi Zhang, Qingcheng Liao, Le Ding, and Jicong Zhang. Bridging 2d and 3d segmentation networks for computation-efficient volumetric medical image segmentation: An empirical study of 2.5 d solutions. *Computerized Medical Imaging and Graphics*, 99:102088, 2022.

[55] Ziyu Zhang, Yi Yu, Simeng Zhu, Ahmed Aly, Yunhe Gao, Ning Gu, and Yuan Xue. Atlas is your perfect context: One-shot customization for generalizable foundational medical image segmentation. *arXiv preprint arXiv:2512.18176*, 2025.

[56] Jianwei Zhao, Fan Yang, Xin Li, Zhicheng Jiao, Qiang Zhai, Xiaomeng Li, De Wu, Huazhu Fu, and Hong Cheng. Segmic: A universal model for medical image segmentation through in-context learning. *Pattern Recognition*, 171:112179, 2026.

A Method

A.1 Problem Definition

A.1.1 Unified multi-mode segmentation.

UniPro formulates medical image segmentation as a unified condition-driven prediction problem, where different forms of guidance specify what or where to segment within the same shared architecture. Given an input image $\mathbf{I}$ and a mode-specific condition $\mathcal{G}$, the model predicts

$$\hat{\mathbf{M}} = f_\theta(\mathbf{I} \mid \mathcal{G}). \tag{14}$$

Under this formulation, UniPro supports three primary 2D segmentation modes (semantic, in-context, and interactive) and further extends in-context segmentation to volumetric propagation as a fourth mode.

- **Mode-1: Semantic segmentation.** Semantic segmentation uses learned class-level priors as the condition. Given a learnable class embedding matrix $\mathbf{P} \in \mathbb{R}^{N_{\text{cls}} \times (p \times C)}$, each class n is represented by $\mathbf{p}_n \in \mathbb{R}^{p \times C}$, which contains p semantic query vectors. The prediction for class n is formulated as
$$\hat{\mathbf{M}}_n = f_\theta(\mathbf{I} \mid \mathbf{p}_n). \tag{15}$$
- **Mode-2: In-context segmentation.** In-context segmentation defines the target object using annotated reference examples instead of a predefined class label. Given a query image $\mathbf{I}_q$ and a support set $\mathcal{S} = \{(\mathbf{I}^{(i)}_{\text{ref}}, \mathbf{M}^{(i)}_{\text{ref}})\}_{i=1}^{K}$, UniPro predicts
$$\hat{\mathbf{M}}_q = f_\theta(\mathbf{I}_q \mid \mathcal{S}). \tag{16}$$
- **Mode-3: Interactive segmentation.** Interactive segmentation uses user-provided prompts, such as positive and negative clicks, as the condition. Given a click set $\mathcal{C} = \{c_i\}_{i=1}^{N_c}$, the model predicts
$$\hat{\mathbf{M}} = f_\theta(\mathbf{I} \mid \mathcal{C}). \tag{17}$$
This mode can either produce masks from scratch or refine predictions initialized by semantic or in-context segmentation.
- **Mode-4: Propagation-based volumetric segmentation.** Propagation is treated as an intra-volume special case of in-context segmentation. Instead of using external annotated support images, the support set is constructed from previously segmented slices within the same 3D volume. Given a volume $\mathbf{V} = \{\mathbf{I}_z\}_{z=1}^{D}$ and a seed prediction $\hat{\mathbf{M}}_{z^*}$ initialized by any of the 2D modes or provided as ground truth, UniPro segments the remaining slices by repeatedly applying
$$\hat{\mathbf{M}}_z = f_\theta(\mathbf{I}_z \mid \mathcal{R}_z), \tag{18}$$
where $\mathcal{R}_z = \{(\mathbf{I}_r, \hat{\mathbf{M}}_r)\}_{r \in \mathcal{N}(z)}$ denotes the intra-volume reference set formed by the seed slice and nearby propagated slices. Thus, Mode-4 shares the same reference-conditioned formulation as Mode-2, but restricts the support examples to the current volume and applies the prediction sequentially along the slice dimension. This enables UniPro to handle 3D volumetric segmentation without introducing a separate 3D-specific architecture.

In the following sections, we describe how UniPro realizes these modes within a shared architecture by converting heterogeneous segmentation conditions into a unified condition representation.

A.1.2 Semantic segmentation

Given an input image $\mathbf{I} \in \mathbb{R}^{3 \times H \times W}$, the image encoder extracts a visual feature map:

$$\mathbf{F} = \text{Encoder}(\mathbf{I}), \quad \mathbf{F} \in \mathbb{R}^{C \times h \times w}, \tag{19}$$

where C denotes the feature dimension and h, w denote the spatial resolution of the encoded features. In semantic segmentation, the condition is provided by learned class-level sparse queries; since no spatial reference mask or user interaction is available, this mode relies solely on sparse queries. We define a learnable semantic prompt matrix $\mathbf{P} \in \mathbb{R}^{N_{\text{cls}} \times (p \times C)}$, where each class n is associated

with a prompt set $\mathbf{p}_n \in \mathbb{R}^{p\times C}$. These prompts are optimized during training and encode class-level anatomical priors. To segment class n, UniPro feeds the image feature $\mathbf{F}$ and the corresponding semantic prompt $\mathbf{p}_n$ into the shared decoder:

$$\hat{\mathbf{M}}_n = \mathrm{Decoder}(\mathbf{F} \mid \mathbf{p}_n). \tag{20}$$

A.1.3 In-context segmentation

In-context segmentation uses annotated reference examples as the conditioning information. Let the support set be $\mathcal{S} = \{(\mathbf{I}_{\mathrm{ref}}, \mathbf{M}_{\mathrm{ref}})\}$, where $\mathbf{I}_{\mathrm{ref}} \in \mathbb{R}^{N_{\mathrm{ref}}\times 3\times H\times W}$ denotes the reference images and $\mathbf{M}_{\mathrm{ref}} \in \mathbb{R}^{N_{\mathrm{ref}}\times 1\times H\times W}$ denotes the corresponding segmentation masks. This conditioning signal is encoded into two complementary components: sparse foreground–background queries and spatially aligned dense fusion features.

Given a query image $\mathbf{I}_q$ and reference images $\mathbf{I}_{\mathrm{ref}}$, the shared image encoder first extracts query and reference features:

$$\mathbf{F}_q = \mathrm{Encoder}(\mathbf{I}_q), \quad \mathbf{F}_k = \mathrm{Encoder}(\mathbf{I}_{\mathrm{ref}}), \tag{21}$$

while the reference image–mask pairs are encoded into value features by a lightweight mask encoder:

$$\mathbf{F}_v = \mathrm{MaskEncoder}(\mathrm{Concat}[\mathbf{I}_{\mathrm{ref}}, \mathbf{M}_{\mathrm{ref}}]). \tag{22}$$

For dense prompts, these encoded features are flattened as

$$\mathbf{K}^{\mathrm{ref}} = \mathrm{Flat}(\mathbf{F}_k) \in \mathbb{R}^{C\times N_{\mathrm{ref}}hw}, \quad \mathbf{V}^{\mathrm{ref}} = \mathrm{Flat}(\mathbf{F}_v) \in \mathbb{R}^{C\times N_{\mathrm{ref}}hw}, \quad \mathbf{K}^{q} = \mathrm{Flat}(\mathbf{F}_q) \in \mathbb{R}^{C\times hw}. \tag{23}$$

Since $\mathbf{K}^{\mathrm{ref}}$ and $\mathbf{K}^{q}$ are extracted by the same encoder, they lie in a shared feature space where query–reference correspondences can be computed. The value features $\mathbf{V}^{\mathrm{ref}}$, derived from reference image–mask pairs, are then transferred to the query space through these correspondences, providing dense mask-aware guidance. Following [14, 15], we adopt the negative squared Euclidean distance to compute the affinity matrix $\mathbf{A}$:

$$\mathbf{A}_{i,j} = -\left\|\mathbf{K}^{\mathrm{ref}}_{:,i} - \mathbf{K}^{q}_{:,j}\right\|_2^2 = 2\,\mathbf{K}^{\mathrm{ref}\top}_{:,i}\mathbf{K}^{q}_{:,j} - \left\|\mathbf{K}^{\mathrm{ref}}_{:,i}\right\|_2^2 - \left\|\mathbf{K}^{q}_{:,j}\right\|_2^2, \tag{24}$$

where the last term can be omitted as shown in [14], which improves efficiency and reduces computational cost. The affinity matrix is then normalized along the reference dimension to obtain the soft assignment weights:

$$\mathbf{W}_{i,j} = \frac{\exp(\mathbf{A}_{i,j})}{\sum_n \exp(\mathbf{A}_{n,j})}, \quad \mathbf{A}, \mathbf{W} \in \mathbb{R}^{N_{\mathrm{ref}}hw\times hw}. \tag{25}$$

The affinity weights $\mathbf{W}$ project the value features $\mathbf{V}^{\mathrm{ref}}$ into the query-aligned space, producing the 2D fusion feature map:

$$\mathbf{F}_{\mathrm{fuse}} = \mathbf{V}^{\mathrm{ref}}\mathbf{W}, \quad \mathbf{F}_{\mathrm{fuse}} \in \mathbb{R}^{C\times hw}. \tag{26}$$

To facilitate subsequent operations, feature channels at all scales are projected to 256 via a linear layer before being fed into the decoder.

For the sparse foreground–background queries, we construct a set of n_s 1-D query vectors $\mathbf{Q}^s \in \mathbb{R}^{n_s\times C}$ from the encoded reference features $\mathbf{K}^{\mathrm{ref}}$, where the first half represents foreground queries and the second half represents background queries. A pooling mask $\mathbf{M}^{\mathrm{pool}} \in [0,1]^{N_{\mathrm{ref}}hw}$ is obtained by downsampling and flattening $\mathbf{M}_{\mathrm{ref}}$. The binary selection vector for the n-th object query is defined as

$$\mathbf{v}_i^n = \begin{cases} 0, & n \le \frac{n_s}{2} \text{ and } \mathbf{M}_i^{\mathrm{pool}} < 0.5, \\ 0, & n > \frac{n_s}{2} \text{ and } \mathbf{M}_i^{\mathrm{pool}} \ge 0.5, \\ 1, & \text{otherwise.} \end{cases} \tag{27}$$

The corresponding object query is computed by masked average pooling:

$$\mathbf{Q}^s_{n,:} = \frac{\sum_{i=1}^{N_{\mathrm{ref}}hw} (\mathbf{K}^{\mathrm{ref}})^{\top}_{i,:}\mathbf{v}_i^n}{\sum_{i=1}^{N_{\mathrm{ref}}hw} \mathbf{v}_i^n}, \quad \mathbf{Q}^s_{n,:} \in \mathbb{R}^C. \tag{28}$$

In summary, the reference examples act as the condition for in-context segmentation. They are encoded into sparse foreground–background queries that capture object-level semantics and dense fusion features that provide spatially aligned mask guidance. The shared decoder then predicts the query mask conditioned on both components:

$$\hat{\mathbf{M}}_q = \mathrm{Decoder}(\mathbf{F}_{\mathrm{fuse}} \mid \mathbf{Q}^s). \tag{29}$$

A.1.4 Interactive segmentation

Interactive segmentation conditions the prediction on user-provided clicks. Given an input image $\mathbf{I}$ and a click set $\mathcal{C} = \{c_i\}_{i=1}^{N_c}$, we follow [45, 36] and assemble a three-channel click prompt $\mathbf{I}_{\text{click}} \in \mathbb{R}^{3\times H\times W}$: the first two channels are disk-encoded maps for positive and negative clicks, and the third channel carries the mask predicted in the previous interaction round, which serves as a memory of past corrections. This prompt provides a 2D dense signal for spatial refinement and is processed by a lightweight prompt encoder.

The encoded click feature is then fused with the visual feature, where the choice of visual feature depends on whether interactive segmentation operates standalone or as a refinement step on top of another mode:

$$\mathbf{F}_{\text{fuse}} = \begin{cases} \text{PromptEncoder}(\mathbf{I}_{\text{click}}) + \text{Encoder}(\mathbf{I}), & \text{Mode-3 or refining Mode-1,} \\ \text{PromptEncoder}(\mathbf{I}_{\text{click}}) + \mathbf{V}^{\text{ref}}\mathbf{W}, & \text{refining Mode-2,} \end{cases} \qquad \mathbf{F}_{\text{fuse}} \in \mathbb{R}^{C\times hw}. \tag{30}$$

When the model runs as a standalone interactive segmenter (Mode-3) or refines a semantic prediction (Mode-1), the click feature is added to the image feature directly. When refining an in-context prediction (Mode-2), it is instead added to the query-aligned reference feature $\mathbf{V}^{\text{ref}}\mathbf{W}$, so that user clicks correct the reference-conditioned output rather than the raw image feature.

In parallel, each click is converted into a sparse object query. Following [22], a click at image coordinates (x, y) is mapped to its feature-grid location $(x', y') = (\lfloor x/s \rfloor, \lfloor y/s \rfloor)$ with downsampling ratio s. We then average-pool features within a $(2r+1)\times(2r+1)$ window centered at (x', y'), apply a linear projection, and add a SAM-style positional embedding [32]. Stacking the results over all clicks yields the click query set $\mathbf{Q}^c \in \mathbb{R}^{N_c\times C}$.

The shared decoder consumes the fused feature together with the appropriate sparse queries, where the queries depend on which mode is being refined:

$$\hat{\mathbf{M}} = \begin{cases} \text{Decoder}(\mathbf{F}_{\text{fuse}} \mid \mathbf{p}_n, \mathbf{Q}^c), & \text{refining Mode-1,} \\ \text{Decoder}(\mathbf{F}_{\text{fuse}} \mid \mathbf{Q}^s, \mathbf{Q}^c), & \text{refining Mode-2,} \\ \text{Decoder}(\mathbf{F}_{\text{fuse}} \mid \mathbf{Q}^c), & \text{Mode-3.} \end{cases} \tag{31}$$

In this way, interactive segmentation reuses exactly the same dense-fusion and sparse-query decoding interface as the other modes: it can stand on its own as a click-driven segmenter, or attach to Mode-1 and Mode-2 as a lightweight refinement step that injects user feedback without altering the backbone or decoder.

A.1.5 Propagation-based volumetric segmentation

Propagation is treated as an intra-volume special case of in-context segmentation. Rather than drawing support examples from an external annotated set, the reference set is built from previously segmented slices within the same 3D volume. Given a volume $\mathbf{V} = \{\mathbf{I}_z\}_{z=1}^{D}$ and a seed prediction $\hat{\mathbf{M}}_{z^*}$ initialized by any of the 2D modes (semantic, in-context, or interactive) or provided as ground truth, UniPro segments the remaining slices by repeatedly applying

$$\hat{\mathbf{M}}_z = f_\theta(\mathbf{I}_z \mid \mathcal{R}_z), \tag{32}$$

where $\mathcal{R}_z = \{(\mathbf{I}_r, \hat{\mathbf{M}}_r)\}_{r\in\mathcal{N}(z)}$ denotes the intra-volume reference set formed by the seed slice together with nearby slices that have already been propagated.

Propagation therefore shares exactly the same reference-conditioned formulation as in-context segmentation, but with two distinguishing properties: the support examples are restricted to the current volume, and predictions are produced sequentially along the slice dimension. Notably, when the seed slice is itself initialized by the semantic class query $\mathbf{p}_n$, the entire pipeline becomes fully automatic, allowing UniPro to perform 3D volumetric segmentation without introducing any 3D-specific architecture. Further details on reference set construction and bidirectional inference are provided in the main text (Section 3.2).

A.2 Model architectures

A.2.1 Shared slice-based image backbone.

UniPro adopts a shared slice-based backbone to process both native 2D images and individual slices from 3D volumes within a unified framework. Given an input slice, the image encoder extracts a hierarchy of multi-scale visual features that serve as the common representation for all segmentation settings. Following standard medical segmentation practice, we use a lightweight UNet-style backbone [27] and retain feature maps at three spatial resolutions, corresponding to $1/16$, $1/8$, and $1/4$ of the input size. These multi-scale features provide complementary semantic and spatial information, and are shared across semantic, in-context, interactive, and propagation-based segmentation.

A.2.2 Reference mask and interaction encoding.

To support reference-conditioned prediction, UniPro further encodes auxiliary conditioning inputs associated with the query slice. Specifically, reference masks are processed by a lightweight mask encoder that produces hierarchical features aligned with the image backbone at the same three resolutions ($1/4$, $1/8$, and $1/16$ of the input resolution). This design enables reference annotations to be fused with visual features in a spatially consistent manner. For interactive segmentation, user prompts are encoded with an analogous lightweight prompt encoder so that clicks or other interaction signals can be mapped into the same multi-scale feature space. By using aligned hierarchical encoders for reference masks and interaction prompts, UniPro preserves a unified conditioning interface across different segmentation settings.

A.2.3 Decoder

As shown in Figure 4, we use $L = 6$ layers in total in the transformer decoder. Similar to Mask2Former [13], we adopt a round-robin strategy for multi-scale interaction between image features and integrated queries. The blocks are scheduled in a cyclic order across scales (i.e., $1 \to 2 \to 3 \to 1 \to 2 \to 3$), ensuring balanced cross-scale information exchange throughout the decoding process. Finally, the decoded features are passed through a lightweight mask decoder that projects them back to the spatial resolution of the input, producing the final segmentation masks.

A.2.4 Mixture-of-Experts FFN Layers

UniPro uses a shared transformer decoder across all segmentation modes. While the attention layers provide a common mechanism for exchanging information between query tokens and fused image features, different conditions, including semantic queries, reference-based object queries, and click queries, may require different nonlinear transformations. To improve mode adaptability without introducing separate decoders, we replace the standard feed-forward network (FFN) in the decoder with a Mixture-of-Experts FFN (MoE-FFN).

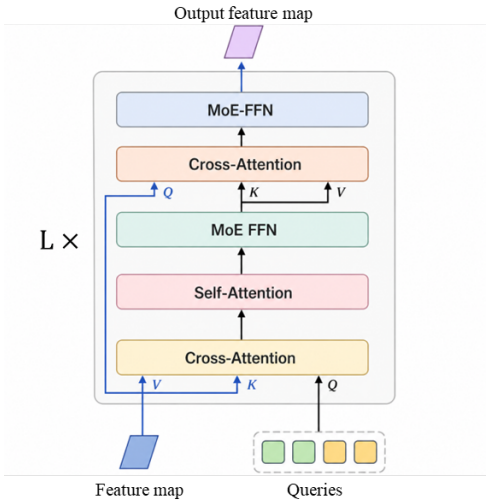


Figure 4: Transformer decoder with MoE-FFN. Queries attend to image features via cross-attention, followed by self-attention and MoE-FFN.

Each expert is implemented as an independent two-layer FFN:

$$\mathrm{FFN}_m(\mathbf{x}) = \mathbf{W}_m^{(2)} \sigma\left(\mathbf{W}_m^{(1)} \mathbf{x}\right), \quad m = 1, \ldots, M, \tag{33}$$

where M is the number of experts, $\mathbf{W}_m^{(1)}$ and $\mathbf{W}_m^{(2)}$ are learnable linear projections of the m-th expert, and $\sigma(\cdot)$ denotes the activation function.

Given a query representation $\mathbf{Q} \in \mathbb{R}^{q \times C}$, a lightweight gating network predicts token-wise expert weights:

$$\boldsymbol{\alpha} = \mathrm{softmax}\left(G(\mathbf{Q})\right), \quad \boldsymbol{\alpha} \in \mathbb{R}^{q \times M}. \tag{34}$$

Each query token is processed by all experts, and their outputs are combined using the gating weights:

$$\text{MoE-FFN}(\mathbf{Q})_i = \sum_{m=1}^{M} \alpha_{i,m} \text{FFN}_m(\mathbf{Q}_i), \quad i = 1, \ldots, q. \tag{35}$$

The output is then added to the input through a residual connection followed by layer normalization:

$$\mathbf{Q}' = \text{LayerNorm}\left(\mathbf{Q} + \text{MoE-FFN}(\mathbf{Q})\right). \tag{36}$$

Only the FFN layers are expert-specialized, while the cross-attention and self-attention layers remain shared across all modes. This design keeps the decoder compact and unified, while allowing different query tokens to adaptively select suitable nonlinear transformations for heterogeneous conditioning signals.

A.3 Implementation

A.3.1 Training details.

We train UniPro with a two-stage training strategy. In the first stage, the model is trained for 100 epochs on single-slice segmentation tasks to learn robust mode-conditioned segmentation capability. Each training batch is randomly assigned to one of the three segmentation modes, with probabilities of 0.3, 0.3, and 0.4 for semantic segmentation, in-context segmentation, and interactive segmentation, respectively. This stage enables the shared backbone and decoder to learn a unified representation across different conditioning formats. In this stage, models are trained on 8 Quadro RTX 8000 GPUs using the AdamW optimizer with a base learning rate of 1×10^{-4}. We use a per-GPU batch size of 8, resulting in a total batch size of 64. A cosine annealing learning-rate scheduler with 10 warm-up epochs is adopted, where the minimum learning rate is scaled by 1×10^{-5}. During training, images are resized to 512×512, with data augmentations including random flipping, affine transformations, brightness and contrast adjustment, Gaussian blur, Gaussian noise, and grid distortion. The segmentation objective combines binary cross-entropy loss and Dice loss:

$$\mathcal{L}_{\text{stage1}} = \mathcal{L}_{\text{2d}} = \mathcal{L}_{\text{dice}} + \mathcal{L}_{\text{ce}}. \tag{37}$$

In the second stage, we fine-tune the entire model, including the image encoder, for 10 epochs with propagation-based segmentation. This adapts the shared representation to sequential prediction with multiple, potentially imperfect image–mask references. Since propagation training is performed on 3D datasets, directly fine-tuning on propagation batches may cause the model to forget the previously learned 2D segmentation capability and degrade its performance on native 2D datasets. To mitigate this issue, we interleave propagation training with regular single-slice training: each propagation batch is paired with a standard 2D training batch sampled with the same mode probabilities as in Stage 1. The propagation batch is constructed from an ordered slice queue, where the model predicts target slices conditioned on previous image-mask references. This stage enables the model to learn slice-wise mask propagation for volumetric segmentation while preserving its compatibility with native 2D inputs and the three original segmentation modes.

In this stage, models are trained using the AdamW optimizer with a base learning rate of 5×10^{-4}. We use a per-GPU propagation batch size of 1. Propagation is performed on 3D chunks with a queue size of 8 slices. For each target slice, the reference window contains 5 image-mask pairs. When more than 5 previous predictions are available, we use a sliding reference window while always retaining the seed slice as an anchor. This anchor-pinned design provides a stable reference throughout propagation and helps reduce semantic drift across long slice sequences. For propagation training, data augmentation is applied consistently to the entire 3D chunk rather than independently to each slice. Specifically, all slices within the same queue undergo the same spatial transformations, ensuring that inter-slice anatomical correspondence is preserved during propagation learning.

The Stage-2 objective combines auxiliary 2D supervision and propagation-based volumetric losses:

$$\mathcal{L}_{\text{stage2}} = \mathcal{L}_{2d} + \mathcal{L}_{\text{prop}} = \mathcal{L}_{2d} + \mathcal{L}_{\text{fwd}} + \lambda_{\text{bwd}}\mathcal{L}_{\text{bwd}} + \lambda_{\text{3d}}\mathcal{L}_{\text{3d}}. \tag{38}$$

Here, $\mathcal{L}_{2d}$ denotes the auxiliary single-slice supervision used to preserve 2D segmentation capability, while $\mathcal{L}_{\text{fwd}}$ and $\mathcal{L}_{\text{bwd}}$ correspond to forward and backward propagation losses. $\mathcal{L}_{\text{3d}}$ is a volume-level consistency loss computed on stacked slice predictions. In our experiments, we set $\lambda_{\text{bwd}} = 0.7$ and $\lambda_{\text{3d}} = 5.0$.

A.3.2 Inference details.

During inference, each input image is resized so that its longer side equals 512 pixels while preserving the aspect ratio, and the predicted masks are remapped back to the original resolution before evaluation. For single-slice (2D) segmentation, UniPro directly predicts a mask for each input image using the chosen mode (semantic, in-context, or interactive), with the same per-mode forward pass as during training. For volumetric (3D) segmentation, we first select the middle slice of the target organ as the seed slice and initialize its mask via semantic, in-context, or interactive segmentation, depending on the evaluation setting. Starting from this seed, UniPro performs bidirectional propagation along the axial direction—forward to subsequent slices and backward to preceding slices—where each target slice is segmented using the anchor-pinned reference set containing the seed and nearby propagated slices. For the main evaluation, we compute Dice over slices that contain the target object. For 3D datasets, this corresponds to volumetric Dice computed over voxels within target-containing slices; for 2D datasets, Dice is computed on target-containing images/slices. This protocol is designed for a unified multi-mode evaluation setting. In interactive segmentation, user input is naturally defined on target-containing slices, since a positive click cannot be placed on an absent object and target-absent slices are typically skipped in practical annotation workflows. Applying the same target-present restriction across 2D and 3D datasets, and across semantic, in-context, and interactive modes, ensures that all modes are evaluated on the same foreground-relevant slice set. To further assess whether this foreground-region evaluation affects the conclusions, we additionally report full-volume Dice over all slices in Appendix D.2.

A.4 Click encoding and simulation.

User interactions are represented as binary disk maps with a radius of 1 pixel during both training and inference. To simulate corrective interactions, we follow the standard protocol of generating clicks from the discrepancy between the current prediction and the ground truth [45, 36]. We place each new click at the centroid of the largest connected error region, so that the simulated clicks better reflect how users typically correct substantial local mistakes in medical images. We consider three click-based operating settings. In semantic or in-context initialization, the model first produces an initial segmentation and then performs two rounds of click-guided refinement. In pure interactive segmentation , the model applies three clicks sequentially, producing an updated mask after each interaction and thus yielding three intermediate refinement results.

For the single-click propagation comparison in Table 1, all methods receive the same initial positive click on the same seed slice for each volume–target pair. Each method generates its own seed mask, whose quality is part of the end-to-end click-to-volume evaluation. In multi-click evaluations, the click-generation policy and budget are matched within each setting, while corrective click locations depend on each method's current prediction.

A.5 Additional implementation details for baseline models

All trainable baselines use the same dataset splits, annotations, and preprocessing as UniPro, with matched training budgets and method-specific input formats. Universal 2D baselines, including Hermes and Iris, are jointly trained on the same multi-dataset training collection as UniPro. In contrast, nnU-Net is trained separately for each dataset following its standard self-configuring pipeline. Trainable propagation baselines use the corresponding 3D training datasets. No additional medical training data or annotations are introduced for UniPro.

Initialization is method-specific. For SAM2, we directly use the official sam2.1_hiera_base_plus checkpoint pretrained on large-scale natural images, without any additional fine-tuning on medical data. SAM2 is therefore excluded from the matched medical-training protocol above. For MedSAM2, the encoder is initialized with MAE-pretrained ViT-Base weights, after which the entire network is fine-tuned end-to-end on our curated medical datasets. Other medical segmentation baselines, such as Hermes, are trained from scratch on the same curated datasets. Released medical checkpoints are not used, to avoid potential overlap with the evaluation data. Positive and negative clicks are simulated consistently across methods using the policy in Appendix A.4.

B Theoretical Analysis of Bidirectional Supervision

We provide a simple idealized analysis to illustrate the intuition behind bidirectional supervision. This analysis is conditional on local stability assumptions and should be viewed as an explanatory perspective rather than a formal generalization guarantee.

B.1 Setup and Notation

Let $\mathbf{M}_0 \in \mathcal{Y}_0$ denote the ground-truth mask on the seed slice, represented as a probability map in $[0,1]^{H\times W}$; we use the Euclidean norm $\|\cdot\|$ throughout. Let $\mathcal{T}_t : \mathcal{Y}_t \to \mathcal{Y}_{t+1}$ be the true forward transfer between slice t and $t{+}1$, so that $\mathbf{M}_{t+1} = \mathcal{T}_t(\mathbf{M}_t)$.

Our model is memory-based: a single reference-conditioned function $f(\mathbf{I}_q, \mathcal{R}_q)$ predicts the mask at a query slice $\mathbf{I}_q$ given a reference set $\mathcal{R}_q$ of previously segmented slices. Forward propagation at step t corresponds to $P_t(\cdot) := f(\mathbf{I}_{t+1}, \mathcal{R}_t^{\mathrm{fwd}})$ with $\mathcal{R}_t^{\mathrm{fwd}}$ containing past forward predictions, while backward reconstruction at step t is $Q_t(\cdot) := f(\mathbf{I}_t, \mathcal{R}_t^{\mathrm{bwd}})$ with $\mathcal{R}_t^{\mathrm{bwd}}$ containing future backward predictions. The operators P_t and Q_t use the same network f but differ in query slice and reference-set content.

The predicted forward trajectory is $\hat{\mathbf{M}}_0 := \mathbf{M}_0$, $\hat{\mathbf{M}}_{t+1} := P_t(\hat{\mathbf{M}}_t)$ for $t = 0, \ldots, n{-}1$. The backward reconstruction chain and its output are

$$Q^{(n:0)} := Q_0 \circ Q_1 \circ \cdots \circ Q_{n-1}, \qquad \tilde{\mathbf{M}}_0 := Q^{(n:0)}(\hat{\mathbf{M}}_n), \tag{39}$$

and the bidirectional consistency loss is $\mathcal{L}_{\mathrm{bi}} := \|\tilde{\mathbf{M}}_0 - \mathbf{M}_0\|$.

B.2 Assumptions

(A1) Bounded one-step forward error. There exist $\delta_t \geq 0$ such that $\|P_t(\mathbf{M}_t) - \mathbf{M}_{t+1}\| \leq \delta_t$ for all t.

(A2) Lipschitz forward operator. There exist $L_t \geq 0$ such that $\|P_t(\mathbf{a}) - P_t(\mathbf{b})\| \leq L_t\|\mathbf{a} - \mathbf{b}\|$ for all $\mathbf{a}, \mathbf{b}$.

(A3) Local non-collapse of the backward chain. For intuition, assume that in a local neighborhood of the true endpoint $\mathbf{M}_n$, the backward chain is approximately non-collapsing and can reconstruct the seed mask up to error β. That is, there exist $\alpha > 0$ and $\beta \geq 0$ such that

$$\|Q^{(n:0)}(\mathbf{z}) - Q^{(n:0)}(\mathbf{M}_n)\| \geq \alpha\|\mathbf{z} - \mathbf{M}_n\|, \quad \forall \mathbf{z}, \tag{A3a}$$

$$\|Q^{(n:0)}(\mathbf{M}_n) - \mathbf{M}_0\| \leq \beta. \tag{A3b}$$

This assumption is not guaranteed globally for learned networks. It only formalizes the desirable regime in which backward reconstruction preserves endpoint discrepancies and penalizes forward trajectories that cannot be mapped back to the reliable seed slice.

B.3 Main Result

Proposition 1 (Idealized endpoint-error bound)**.** *Under Assumption (A3),*

$$\|\hat{\mathbf{M}}_n - \mathbf{M}_n\| \;\leq\; \frac{\mathcal{L}_{\mathrm{bi}} + \beta}{\alpha}. \tag{40}$$

Proof. Applying (A3a) with $\mathbf{z} = \hat{\mathbf{M}}_n$,

$$\alpha\|\hat{\mathbf{M}}_n - \mathbf{M}_n\| \;\leq\; \|Q^{(n:0)}(\hat{\mathbf{M}}_n) - Q^{(n:0)}(\mathbf{M}_n)\|. \tag{41}$$

By the triangle inequality,

$$\|Q^{(n:0)}(\hat{\mathbf{M}}_n) - Q^{(n:0)}(\mathbf{M}_n)\| \leq \|Q^{(n:0)}(\hat{\mathbf{M}}_n) - \mathbf{M}_0\| + \|Q^{(n:0)}(\mathbf{M}_n) - \mathbf{M}_0\| \tag{42}$$

$$\leq \mathcal{L}_{\mathrm{bi}} + \beta. \tag{43}$$

Dividing by α yields Eq. (40). □

Proposition 2 (One-step accumulation bound)**.** *Under (A1)–(A2) with uniform constants $L_t \le L$ and $\delta_t \le \delta$,*

$$\|\hat{\mathbf{M}}_n - \mathbf{M}_n\| \;\le\; \delta \sum_{i=0}^{n-1} L^i. \tag{44}$$

Proof. Let $e_t := \|\hat{\mathbf{M}}_t - \mathbf{M}_t\|$. Then

$$e_{t+1} \le \|P_t(\hat{\mathbf{M}}_t) - P_t(\mathbf{M}_t)\| + \|P_t(\mathbf{M}_t) - \mathbf{M}_{t+1}\| \le L_t e_t + \delta_t. \tag{45}$$

With $e_0 = 0$, unrolling gives $e_n \le \delta \sum_{i=0}^{n-1} L^i$. □

Corollary 1 (Combined conditional bound)**.** *Under (A1)–(A3),*

$$\|\hat{\mathbf{M}}_n - \mathbf{M}_n\| \;\le\; \min\left\{ \delta \sum_{i=0}^{n-1} L^i,\; \frac{\mathcal{L}_{\mathrm{bi}} + \beta}{\alpha} \right\}. \tag{46}$$

Interpretation. This analysis should be understood as an idealized intuition rather than a strict guarantee. The first term, $\delta \sum_{i=0}^{n-1} L^i$, describes standard forward error accumulation, where local errors may grow along long propagation chains. The second term, $(\mathcal{L}_{\mathrm{bi}} + \beta)/\alpha$, suggests that, when the backward chain is locally stable and non-collapsing, a smaller bidirectional reconstruction loss is consistent with a smaller endpoint error. The key takeaway is that bidirectional supervision acts as a cycle-consistency regularizer: it penalizes forward trajectories whose accumulated errors prevent reconstruction back to the reliable seed slice. Thus, while it does not universally guarantee low propagation error, it provides an additional long-range constraint that helps explain the empirical reduction of drift in volumetric propagation.

C Dataset

C.1 Dataset Details

We use a diverse collection of public medical image segmentation datasets covering 3D CT/MRI volumes and native 2D images across abdominal, cardiac, brain, skin, breast, lung, and endoscopic targets. Below we briefly describe each dataset used in training and evaluation.

AbdomenCT-1K [38]. A large-scale abdominal CT benchmark with expert annotations for the liver, kidney, spleen, and pancreas, aggregated from multiple medical centers. To prevent severe class imbalance with other datasets in the training pool, we use only a subset of the released data, split into 352 training, 20 validation, and 76 test volumes.

Combined (CT-MR) Healthy Abdominal Organ Segmentation (CHAOS) [31, 30]. A multi-modal abdominal dataset from the ISBI 2019 CHAOS Challenge, comprising CT and MRI scans of healthy subjects with expert annotations for the liver, right kidney, left kidney, and spleen. We use only the MRI subset, split into 36 training, 9 validation, and 15 test volumes.

Multi-Centre, Multi-Vendor & Multi-Disease Cardiac Image Segmentation Challenge (M&Ms) [9]. A multi-centre, multi-vendor cardiac MRI dataset from the MICCAI 2020 Challenge, comprising short-axis cine scans of patients with cardiomyopathies and healthy controls. Expert annotations of the left ventricle, right ventricle, and left ventricular myocardium are provided at end-diastolic and end-systolic phases. We follow the official train/test split.

The Liver Tumor Segmentation Benchmark (LiTS) [6] . A multi-center benchmark of 201 contrast-enhanced abdominal CT scans (131 training, 70 testing) covering patients with diverse liver tumors, including hepatocellular carcinoma and metastases from colorectal, breast, and lung cancers, with expert annotations for liver and tumor regions. We use the 131 public cases with an 80%/20% split, retaining only the tumor labels.

The Kidney Tumor Segmentation 2019 dataset (KiTS) [26]. A dataset of CT scans from 300 patients who underwent nephrectomy for kidney tumors at the University of Minnesota Medical Center (2010–2018), of which 210 cases are publicly released. We split the public portion 80%/20% for training and testing, retaining only the tumor labels.

Dataset of Breast Ultrasound Images (Breast) [1]. A collection of 780 grayscale breast ultrasound images from 600 female patients (aged 25–75) at Baheya Hospital, Egypt, categorized as normal, benign, or malignant, with expert-annotated lesion masks. We use only the 647 benign and malignant cases with an 80%/20% train/test split.

Chest X-ray Masks and Labels (Chest) [29]. A chest radiograph dataset jointly released by the National Library of Medicine (Bethesda, USA) and Shenzhen No.3 People's Hospital (China), covering a broad range of abnormalities such as effusions and miliary patterns. We use only the lung segmentation masks, with an 80%/20% train/test split.

BKAI-IGH NeoPolyp Dataset (BKAI) [41]. A colonoscopy dataset of 1,200 images (1,000 white-light and 200 FICE) released by the BKAI Research Center and the Institute of Gastroenterology and Hepatology, Vietnam, with expert-verified segmentation masks distinguishing neoplastic from non-neoplastic polyps. We adopt an 80%/20% split on the official training set of 1,000 images.

International Skin Imaging Collaboration (ISIC) [46, 17]. A large-scale dermoscopic image collection from the ISIC Skin Lesion Analysis Challenges, with expert-annotated lesion boundary masks spanning diverse skin tones, lighting conditions, and lesion morphologies. We use the ISIC 2018 segmentation subset and follow the official train/validation/test split.

Beyond the Cranial Vault (BTCV) [33]. An abdominal CT dataset with multi-organ annotations, used as an external 3D test set to evaluate cross-benchmark generalization. To align the label space with AbdomenCT-1K, we merge the left and right kidneys into a single class and retain only the organs present in AbdomenCT-1K (liver, kidney, spleen, and pancreas).

Automatic Cardiac Diagnosis Challenge (ACDC) [5]. A cardiac MRI dataset acquired at the University Hospital of Dijon with 1.5T and 3.0T Siemens scanners, providing short-axis cine sequences annotated for the same structures as M&Ms at end-systolic and end-diastolic phases. For validation, we use 200 ED/ES phase volumes from the 100 patients in the official training set; for final evaluation, we use 100 phase volumes from the 50 patients in the official test set.

University of Waterloo Skin Cancer (UW-SC) [47]. A collection of 167 dermoscopic skin lesion images with expert-verified segmentation masks, used solely as an external test set.

Breast Ultrasound Dataset B (BUS) [51]. A breast ultrasound dataset of 163 images with expert-annotated lesion regions. We merge benign and malignant lesions into a single class and use it solely as an external test set.

Brain Tumor Segmentation (BraTS) [4]. A multi-modal brain tumor MRI dataset, used as an unseen 3D test set to evaluate generalization to an anatomical target and imaging domain absent from the training pool. We use only the T2-weighted modality. The original annotations contain three tumor sub-regions (enhancing tumor, tumor core, and peritumoral edema); to reduce the difficulty of this unseen-class evaluation, we merge them into a single tumor region label.

C.2 Dataset statistics

Table 4 summarizes the datasets used to train and evaluate our framework. We draw from a diverse collection of publicly available medical image segmentation benchmarks covering a wide range of imaging modalities and anatomical regions. The training pool comprises nine datasets spanning abdominal CT/MRI (AbdomenCT-1K, CHAOS, LiTS, KiTS), cardiac MRI (M&Ms), ultrasound (Breast Cancer), dermoscopy (ISIC), endoscopy (BKAI), and chest X-ray. These datasets differ substantially in scale—ranging from a few hundred to several thousand 3D volumes or 2D slices—as well as in annotation granularity, from single-organ masks to multi-organ delineations. To rigorously

assess generalization, we further include held-out test sets covering both external domains (BTCV, ACDC, UW-SC, BUS), which share class definitions with the training data but originate from different cohorts or acquisition centers, and unseen classes (BraTS), which evaluate the framework's ability to segment anatomical structures absent from training. This split allows us to disentangle cross-center robustness from cross-class transferability.

Table 4: Datasets statistics.

Dataset	# cls	Modality	3D Volumes			2-D Slices		
			Train	Validate	Test	Train	Validate	Test
AbdomenCT-1K[38]	4	CT	352	20	76	39956	2557	8665
CHAOS[31, 30]	4	MRI	36	9	15	4984	1309	1945
M&Ms [9]	3	MRI	300	-	340	2475	-	2821
LiTS [6]	1	CT	104	-	27	5687	-	1625
KiTS [26]	1	CT	168	-	42	5269	-	1609
Chest X-ray [1, 29]	1	X_ray	-	-	-	450	-	116
Breast Cancer [1]	1	Ultrasound	-	-	-	518	-	129
BKAI [41]	2	Endoscopy	-	-	-	800	-	200
ISIC [46, 17]	1	Dermoscopy	-	-	-	2594	100	1000
BTCV [33]	4	CT	-	-	30	-	-	3791
ACDC [5]	3	MRI	-	200	100	-	1841	1001
UW-SC [47]	1	Dermoscopy	-	-	-	-	-	167
BUS [1]	1	Ultrasound	-	-	-	-	-	163
BraTS [4]	1	MRI	–	–	74	-	-	4761

C.3 Data preprocessing

Preprocessing is dataset-specific and retains both isotropic and anisotropic volumes. AbdomenCT-1K, CHAOS, BTCV, and KiTS are resampled to an isotropic spacing of $1.5 \times 1.5 \times 1.5$ mm. M&Ms and ACDC are resampled in-plane to 1.5×1.5 mm while preserving their native through-plane spacing of 5–10 mm, yielding strongly anisotropic volumes. LiTS uses $0.7676 \times 0.7676 \times 1.0$ mm spacing and is mildly anisotropic, whereas BraTS uses isotropic $1.0 \times 1.0 \times 1.0$ mm spacing. These spacings refer to the volumes before the final 2D input resizing. For intensity preprocessing, CT volumes are clipped to $[-991, 500]$ HU for AbdomenCT-1K and BTCV, and $[-200, 250]$ HU for KiTS and LiTS. MRI volumes from CHAOS, M&Ms, ACDC, and BraTS are clipped to their volume-wise 0th–99th intensity percentiles. All volumes are then rescaled to $[0, 255]$ using volume-wise min–max normalization. UniPro processes ordered slices without explicit voxel-spacing or physical-distance inputs; isotropic spacing is therefore not required, although large through-plane gaps may weaken cross-slice correspondence.

D More Experiments

D.1 Effect of flexible interactive correction

We compare two interactive 3D segmentation paradigms under the same annotation budget: *slice-wise interaction*, where the user corrects foreground slices independently, and *click-and-propagate*, where clicks on selected slices are propagated to the rest of the volume. Both protocols use the same robot click generator, the same per-slice click cap, the same total click budget, and the same volume-level stopping criterion. Therefore, differences in annotation cost reflect the interaction protocol rather than the click oracle.

Robot click generation. Clicks are generated by a deterministic robot user. For each slice, the first click is sampled from the largest connected component of the foreground ground truth. Subsequent clicks are placed at the centroid of the largest connected component in the current 2D error map, computed between the thresholded prediction and the ground truth. A click is labeled positive if the selected error component is a false-negative region, and negative if it is a false-positive region. No human input is used at any stage.

Common budget and stopping rules. For each $(\text{volume}, \text{organ})$ pair, both protocols are given a total budget of $B = 100$ clicks. Within a single slice, at most $C = 5$ clicks can be used. A slice-level

click loop stops early once the 2D Dice reaches target T. For the seed slice in the propagation protocol, we use a stricter threshold $\tau_{2D} = 0.95$ to obtain a cleaner reference mask. The full $(\text{volume}, \text{organ})$ evaluation stops when either the volume-level 3D Dice reaches the dataset-specific target T, or the click budget B is exhausted. We set $T = 0.90$ for ACDC and $T = 0.80$ for BraTS20.

Slice-wise interaction. This protocol represents conventional per-slice correction. The user first provides one positive click on every foreground slice, producing an initial stack of independent 2D predictions. The method then enters a refinement loop. At each step, we select the slice with the largest 2D error among slices that have not reached the per-slice click cap, and add clicks according to the robot click rule. The process repeats until the 3D Dice target T is reached or the total budget is exhausted. Importantly, no information is shared across slices: each slice is segmented and refined independently.

Click-and-propagate. This protocol evaluates the proposed propagation-based annotation workflow. The seed slice is chosen as the median foreground slice. The robot user places up to $C = 5$ clicks on the seed slice, with early stopping at $\tau_{2D} = 0.95$. The resulting seed mask is then used as the initial reference to propagate predictions to the remaining slices. If the propagated volume does not reach the target 3D Dice, the protocol enters an iterative refinement loop. At each iteration, we select the unclicked slice with the largest 2D error, add up to $C = 5$ clicks on that slice, and pin every clicked slice so that its corrected mask is not overwritten by later propagation. Propagation is then re-run using the top-$K = 5$ pinned slices, ranked by their 2D Dice, as references. The loop continues until the same termination conditions are met: reaching the dataset-specific 3D Dice target T or exhausting the $B = 100$ click budget.

Datasets and evaluated pairs. We evaluate on the full test split of ACDC and BraTS. For ACDC, we evaluate 100 volumes with three cardiac organs, resulting in 300 $(\text{volume}, \text{organ})$ pairs, with target $T = 0.90$. For BraTS20, we evaluate 74 volumes with one tumor class, resulting in 74 pairs, with target $T = 0.80$. We use a lower target for BraTS because it is an unseen dataset. For each dataset, we report two annotation-cost metrics. The first is the **Number of Clicks (NoC)**, defined as the number of clicks required to first reach the target 3D Dice T, censored at $B = 100$ if the target is not reached. This censoring makes the mean NoC conservative. The second is the **number of edited slices**, defined as the number of distinct slices that receive at least one click. NoC measures direct click effort, while the number of edited slices captures slice-level scrolling and inspection effort, which is an important practical cost in volumetric annotation. Both metrics are averaged across all evaluated pairs in each dataset.

D.2 Full-Volume Dice versus Target-Containing-Slice Dice

Our main experiments report Dice on target-containing slices to maintain a unified evaluation protocol across 2D/3D data and semantic, in-context, and interactive modes. We additionally compute standard full-volume Dice over all slices. As shown in Table 5, the two metrics remain close across datasets, indicating that target-containing-slice evaluation does not substantially inflate performance in most cases. Larger gaps appear on KiTS and LiTS, where targets are smaller, sparser, or more prone to axial false positives; tumor cases are similarly more sensitive because small leakage into target-absent slices can noticeably reduce full-volume Dice. These results support using target-containing-slice Dice as a foreground-region metric for interactive evaluation, while reporting full-volume Dice as the standard volumetric metric.

D.3 Robustness to Seed-Slice Selection

Our default propagation protocol selects the seed slice as the median axial index among slices containing the target organ. This choice provides a simple and deterministic initialization. To examine whether the propagation performance depends critically on this seed location, we conduct a controlled robustness study on ACDC by selecting seed slices at different relative positions within the target-organ slice range. In this analysis, the selected seed slice is initialized with the ground-truth mask for all seed choices. This removes the confounding effect of seed-slice segmentation quality and isolates the effect of seed location on propagation. Specifically, we evaluate seeds at the 20%, 40%, 50%, 60%, and 80% axial quantiles of the target-organ extent, as well as a deterministic random seed

Table 5: Comparison between standard full-volume Dice (%) and target-containing-slice Dice. Results are averaged over organs within each dataset when applicable. Δ denotes target-containing-slice Dice minus full-volume Dice, reported in percentage points.

Dataset	Full-vol Dice	Target-slice Dice	Δ
Abd-1K	86.11	88.62	+2.51
CHAOS	87.05	89.70	+2.65
M&M	86.06	87.64	+1.58
LiTS	63.32	66.95	+3.63
KiTS	73.05	76.95	+3.90
BTCV	82.19	84.40	+2.21
ACDC	87.80	87.97	+0.17

Table 6: Seed-position robustness on ACDC under ground-truth seed-mask initialization. GT-init removes seed-slice segmentation error and isolates the effect of seed location on propagation. All results are 3D Dice on target-containing slices.

Seed slice	$q = 0.2$	$q = 0.4$	$q = 0.5$	$q = 0.6$	$q = 0.8$	Random
Mean Dice	89.81	91.03	90.98	90.53	89.46	90.12
Δ vs. median	-1.17	+0.05	0.00	-0.45	-1.52	-0.86

slice sampled independently for each volume. All remaining slices are then segmented by the same propagation procedure.

As shown in Table 6, UniPro is not highly sensitive to the exact seed-slice location when the seed mask is fixed to the ground truth. Across different seed choices, the mean Dice remains in a narrow range from 89.46 to 91.03. Seeds near the two ends of the target-organ extent, such as $q = 0.2$ and $q = 0.8$, show slightly larger drops, which is expected because boundary seeds provide less balanced context for covering the full organ range. In contrast, seeds around the middle of the organ extent remain stable, with $q = 0.4$ even slightly outperforming the median seed. These results support using the median slice as a simple, deterministic, and standardized choice for evaluation, rather than an oracle condition on which the method critically depends.

D.4 Statistical Analysis of Propagation Performance

We assess the statistical reliability of the propagation results by reporting variability across evaluated volume–target pairs and paired comparisons with MedSAM2. Table 7(a) summarises UniPro's mean Dice, standard deviation, and 95% confidence intervals computed using Student's t-distribution. Paired t-tests show significant improvements on five of the eight datasets at the unadjusted $p < 0.05$ level (Table 7(b)), while differences on CHAOS, LiTS, and KiTS are not statistically significant. The wider confidence intervals on LiTS, KiTS, and BraTS indicate greater uncertainty in these settings. All statistics are based on a single training run per method and describe test-pair variability rather than variation across training runs.

We additionally compare UniPro with Sli2Vol+ and Vol2Flow on AbdomenCT-1K liver propagation using the same seed slice and ground-truth seed mask. UniPro achieves 94.03% Dice, compared with 86.05% for Sli2Vol+ and 83.81% for Vol2Flow (Table 8).

D.5 Inference Runtime and Memory

We compare UniPro with a 3D nnU-Net of similar parameter count on AbdomenCT-1K test volumes using a single NVIDIA Quadro RTX 8000 GPU and a batch size of 1. The 3D nnU-Net uses 128^3 sliding-window patches at 1.5 mm isotropic spacing, while UniPro processes chunks of eight 3-channel 512×512 slices. Table 9 reports median per-volume inference time and peak GPU memory under these native input configurations. UniPro uses approximately 46% less peak memory but takes longer per volume (28.0 s versus 9.3 s), reflecting a memory–runtime trade-off. Sequential propagation limits parallelization because later predictions depend on earlier outputs.

Table 7: Statistical analysis of propagation performance. (a) UniPro's mean Dice (%), SD, and 95% CI. (b) Paired comparisons with MedSAM2, where Δ denotes the Dice difference in percentage points. Bold p-values indicate significance at the unadjusted $p < 0.05$ level.

(a) UniPro performance

Dataset	Mean	SD	95% CI
Abd-1K	88.62	12.64	[87.19, 90.05]
CHAOS	89.70	9.05	[87.37, 92.04]
M&Ms	87.64	10.31	[87.01, 88.28]
LiTS	66.95	28.60	[55.15, 78.76]
KiTS	76.95	18.94	[71.04, 82.85]
BTCV	84.40	14.70	[81.32, 87.48]
ACDC	87.97	13.06	[86.49, 89.45]
BraTS	54.89	23.03	[49.56, 60.23]

(b) UniPro versus MedSAM2

Dataset	Δ	p-value
Abd-1K	$+2.33$	$\mathbf{0.0165}$
CHAOS	-0.94	0.5254
M&Ms	$+4.67$	$\mathbf{5.49 \times 10^{-11}}$
LiTS	$+4.70$	0.3136
KiTS	$+3.63$	0.1911
BTCV	$+3.85$	$\mathbf{0.0035}$
ACDC	$+3.79$	$\mathbf{0.0066}$
BraTS	$+14.15$	$\mathbf{1.14 \times 10^{-4}}$

Table 8: Liver propagation on AbdomenCT-1K with the same seed slice and ground-truth seed mask for all methods. Dice scores are percentages.

Method	Dice (↑)
Vol2Flow	83.81
Sli2Vol+	86.05
UniPro (GT seed)	**94.03**

These measurements compare the two native deployment configurations and do not imply a general reduction in computation or inference latency.

D.6 Contributions of Stage-I Pretraining and Stage-II Propagation Training

To assess the contributions of the two training stages, we train all variants using only the cardiac subset of our training data and evaluate on ACDC. We compare Stage I only, Stage II only without Stage-I pretraining, and the full two-stage pipeline. Apart from the ablated stage, all other training and evaluation settings are kept identical across variants. At inference time, every variant receives the same seed slice and ground-truth seed mask for each volume–target pair, isolating propagation performance from differences in initialization quality. This cardiac-only ablation uses a different protocol from Table 3(b), which evaluates 2D interleaving under multi-dataset training with semantic seed predictions.

As shown in Table 10, Stage I alone enables propagation, while Stage II alone can learn propagation without unified pretraining. Combining the two stages achieves the highest Dice, improving over Stage I only and Stage II only by 3.69 and 2.86 Dice points, respectively. These results support the benefit of combining unified pretraining with propagation-specific adaptation in this controlled cardiac setting.

E Visualization

E.1 Comparison with propagation baselines

We further illustrate the behavior of UniPro on pancreas segmentation, one of the most challenging targets in abdominal imaging due to its small size, elongated and curved shape, low contrast against surrounding soft tissues, and high inter-patient variability in both location and morphology. As shown in Figure 5, all methods are initialized from a single positive click on the seed slice ($z = 34$) and required to propagate the resulting mask through the entire volume. UniPro recovers a coherent pancreas across distant slices ($z = 22$ to $z = 48$) and reaches a 3D Dice of 0.834, while propagation baselines fail in characteristic ways: SAM2 (0.410) leaks heavily into adjacent organs and tissues; nnInteractive (0.445), despite being a dedicated 3D interactive model, loses the target on slices far from the seed; iSegFormer (0.503) and PAM (0.541) exhibit progressive boundary inflation as

Table 9: Inference runtime and memory on AbdomenCT-1K using one NVIDIA Quadro RTX 8000 GPU and batch size 1. Time is the median per volume; memory is peak GPU memory during inference.

Model	Parameters	Native input	Time (s)	Memory (GB)
3D nnU-Net	31.84M	128^3 patch	9.3	3.60
UniPro	32.84M	$8 \times 3 \times 512 \times 512$ chunk	28.0	1.93

Table 10: Training-stage ablation using only cardiac training data. All variants are evaluated on ACDC with the same ground-truth seed masks. Dice scores are percentages.

Training strategy	ACDC Dice (GT seed)
Stage I only	87.29
Stage II only, without Stage-I pretraining	88.12
Full two-stage UniPro	**90.98**

propagation proceeds; and MedSAM2 (0.665), the strongest medical propagation baseline, still under-segments the head and tail of the pancreas. The 3D reconstructions in the rightmost column further highlight that UniPro produces a continuous, anatomically plausible pancreas, whereas baselines show fragmented or distorted volumes. This case illustrates that the reference-conditioned propagation in UniPro remains stable on small, low-contrast, elongated structures where slice-by-slice drift typically dominates.

E.2 Success and failure cases on tumor datasets

Figure 6 presents selected success and failure cases on LiTS, KiTS, and BraTS. Each example shows the initialized seed slice, predictions on four additional slices at increasing absolute slice-index distances from the seed, and the corresponding 3D surfaces. The successful LiTS and KiTS examples recover the dominant tumor component across slices despite its irregular shape. The successful BraTS example similarly captures the main lesion, although neither brain anatomy nor brain tumors are included in training. These examples nevertheless retain local errors, including missed small foci and false-positive extensions.

The failure cases highlight limitations on fragmented and heterogeneous targets. In LiTS and KiTS, predictions cover part of the initialized region but miss small, spatially separated components on more distant slices. The BraTS failure captures only part of a heterogeneous lesion and omits substantial surrounding and disconnected regions. Together, these observations suggest that appearance consistency and cross-slice traceability influence propagation beyond whether the anatomy was seen during training. They motivate improved handling of newly appearing components and weak correspondence between neighboring slices.

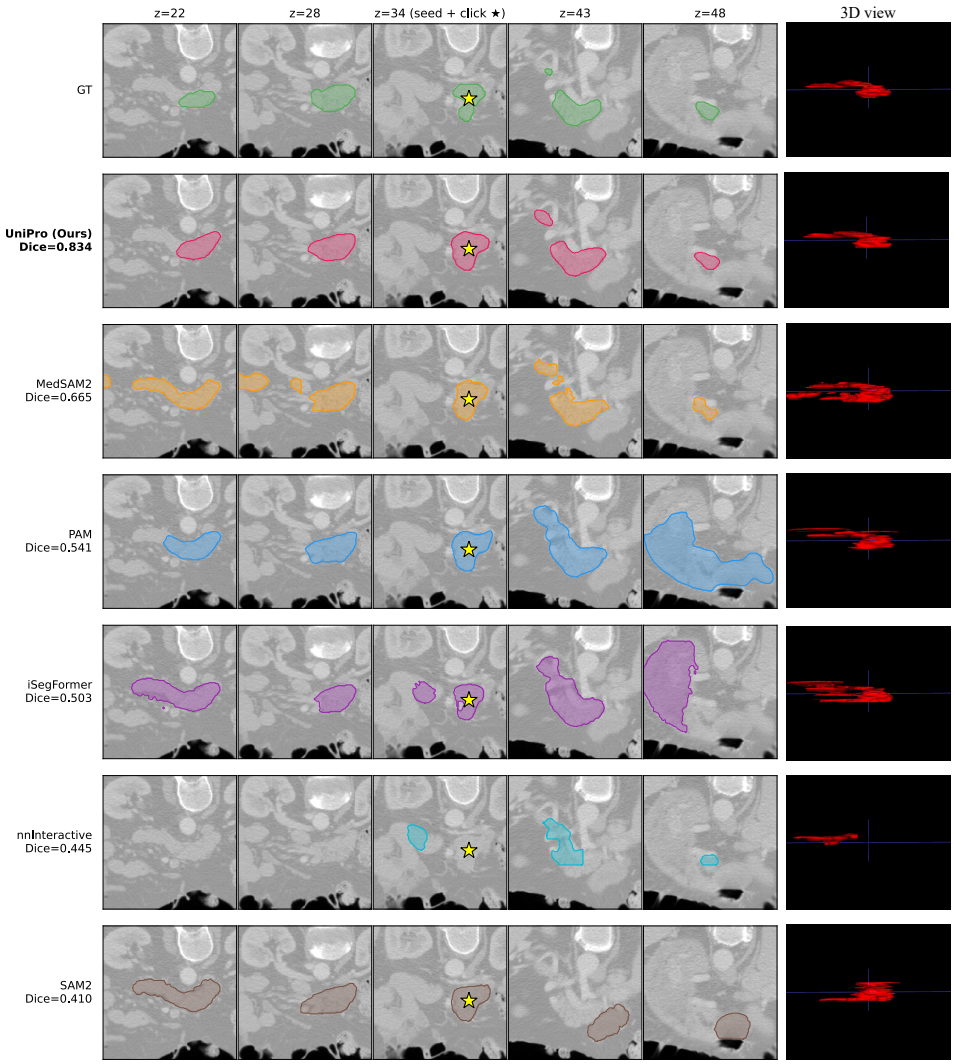


Figure 5: Qualitative comparison on pancreas segmentation under the single-click propagation setting. Each row shows a different method propagating from a single positive click (★) on the seed slice ($z = 34$) to representative slices at $z = 22, 28, 43, 48$, with the resulting 3D reconstruction shown in the rightmost column. UniPro produces masks that closely follow the ground-truth pancreas boundary across the entire volume, while baselines exhibit boundary leakage (PAM, iSegFormer), partial coverage (nnInteractive, MedSAM2), or large over-segmentation toward neighboring organs (SAM2). Per-volume 3D Dice is reported next to each method name.

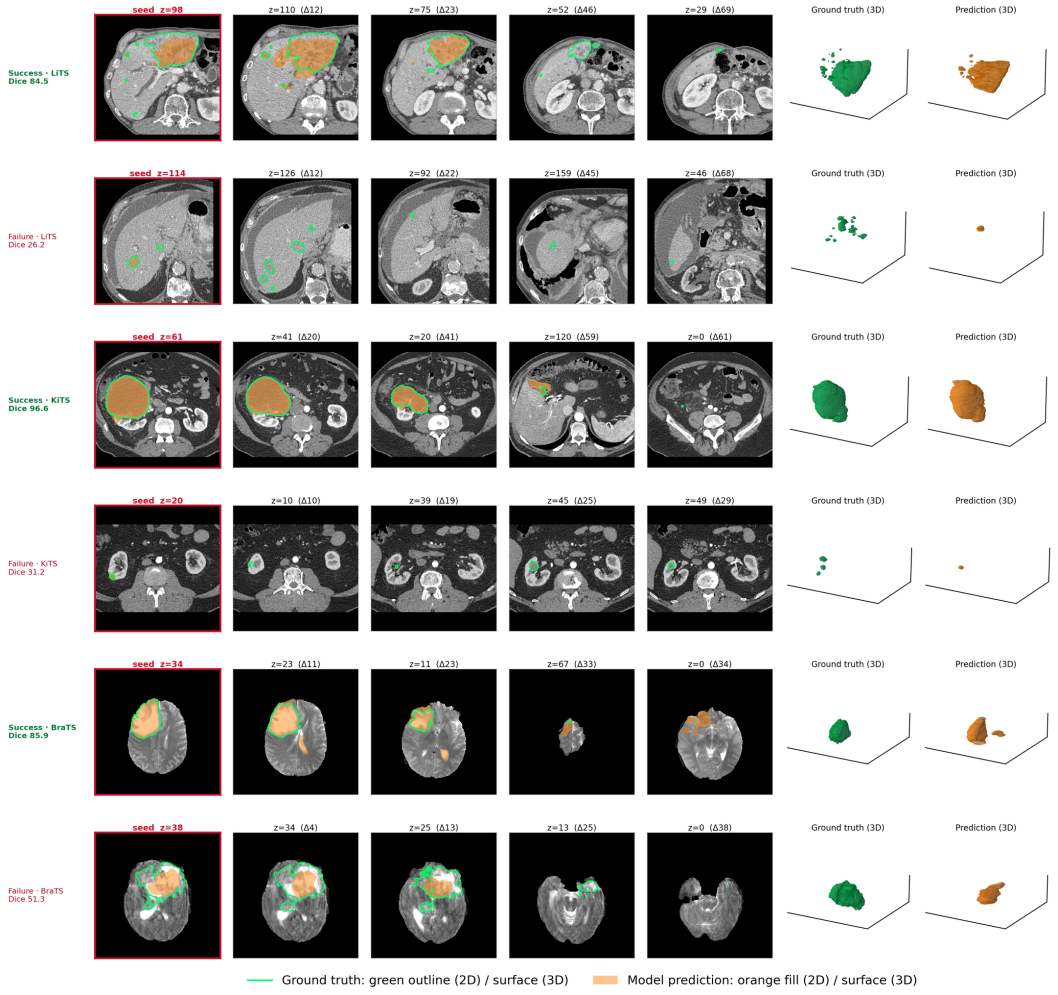


Figure 6: Selected success and failure cases on LiTS, KiTS, and BraTS. Each row shows the seed slice (red border), four additional slices ordered by absolute distance from the seed, and 3D ground-truth and predicted surfaces. Green contours and orange overlays denote ground truth and predictions; 3D surfaces use the same colors. z is the slice index and Δ the absolute index distance in the preprocessed volume.

F The Use of Large Language Models (LLMs)

In this work, large language models (LLMs) were used as general-purpose assistants for code development, manuscript writing, and figure illustration. Specifically, LLMs aided in routine coding tasks (e.g., debugging, boilerplate generation, and refactoring), language polishing and formatting of the manuscript. They did not contribute to research ideation, experimental design, analysis, or result interpretation.

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]

2. **Limitations**

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

   Answer: [Yes]

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]

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]

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]

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]

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: [Yes]

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]

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]

10. **Broader impacts**

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

    Answer: [N/A]

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]

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: [N/A]

13. **New assets**

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

    Answer: [N/A]

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]

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]

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: [Yes]