MISEG-AGENT: A HUMAN-IN-THE-LOOP AGENTIC FRAMEWORK FOR ERROR DIAGNOSIS AND EXECUTABLE PROMPT REVIEW IN 3D MEDICAL IMAGE SEGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

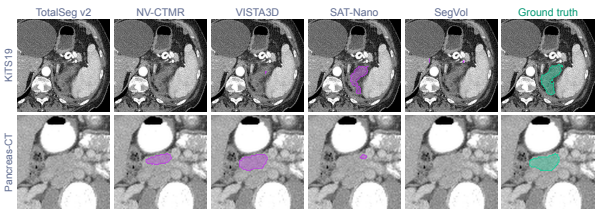
Pretrained models can provide useful initial masks for 3D medical image segmentation, but their accuracy and resource demands vary across targets, and residual errors may be scattered throughout a volume. Interactive models offer a way to refine these masks, yet users must still find errors, choose where to intervene, and assess the resulting segmentation. These decisions are interdependent: each accepted update changes the mask on which later recommendations rely. We introduce MISeg-Agent, a human-in-the-loop agentic framework that coordinates model selection, error diagnosis, point proposal, and review as correction proceeds, keeping proposed corrections tied to the current mask. The framework selects a compatible pretrained model using benchmark results, identifies likely missing and excess foreground from the image and current mask, and proposes slices and points for review. It then tests candidate point prompts with an interactive segmentation model and presents the results for human assessment. The diagnosis requires no access to the selected model's internal features, so the same review process accommodates different initial masks. In the evaluated CT cases, MISeg-Agent produces final masks that closely match reference annotations. By combining benchmark-informed model selection with error-guided review and tested corrective points, MISeg-Agent gives users specific locations and observed responses to assess before approving a mask update. Our code is available at `https://github.com/ICLR-2027-submission/MISEG-AGENT`.

1 INTRODUCTION

Segmentation masks support quantitative analysis and annotation in 3D medical imaging Shaker et al. (2024); Zhou et al. (2021); Hatamizadeh et al. (2022a;b); Milletari et al. (2016). Pretrained segmentation models increasingly provide useful initial predictions He et al. (2024); Zhao et al. (2023); Du et al. (2023); Wasserthal et al. (2023). However, segmentation quality varies across targets, and runtime and GPU memory requirements differ. A model suitable for one task or deployment setting may not suit another. Even strong initial masks can contain errors requiring correction (Fig. 1(a)). These errors may be sparse and distributed across slices, including missing foreground, neighboring tissue mislabeled as foreground, and fragmented structures. The practical challenge is therefore to select a pretrained model and help users correct its output with manageable effort.

Interactive models respond to clicks, scribbles, and other spatial prompts Wong et al. (2024); Isensee et al. (2025), but users must still locate false-negative (FN) and false-positive (FP) regions and choose corrective prompts. This can be difficult for users with limited anatomical experience. Moreover, a positive point may recover tissue but introduce leakage, while a negative point may remove both excess and valid foreground. Fig. 1(b) illustrates the risk of applying generated points without screening: using all candidates yields a mean Dice of 0.701, below both the initial mask (0.927) and mask-only refinement (0.879). At matched per-case point counts, mean Dice is higher with screened points than with the first k unscreened candidates from the same pool (0.892 versus 0.870), but remains below the initial-mask Dice. These automatic-refinement results motivate checking candidate responses and reviewing the resulting masks before accepting updates.

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



(a)

	Dice ↑	Δ Dice	Points
Initial mask	0.927 ± 0.040	$+0.000$	0.0
Mask-only nnInteractive	0.879 ± 0.124	-0.048	0.0
3D prompts (all)	0.701 ± 0.310	-0.226	18.6
3D prompts (first k)	0.870 ± 0.150	-0.057	1.7
3D prompts (screened)	0.892 ± 0.094	-0.035	1.7

(b)

Figure 1: Motivation for MISeg-Agent. (a) Segmentation performance of different pretrained segmentation models, showing that no single pretrained model is uniformly best across CT targets. (b) Mean Dice ($\pm$ standard deviation) over 120 cases for the initial mask and four nnInteractive Isensee et al. (2025) refinement settings. The first-k and screened rows use the same initial mask and per-case point count; both retain the initial mask when that count is zero. Δ Dice is the displayed mean minus the initial-mask mean. Points is the mean submitted count per volume. Screening details appear in Appendix E.1.

Existing quality-control methods use the image and mask to predict voxel-wise errors and identify regions for review Specktor-Fadida et al. (2025); Zaman et al. (2023). For interaction planning, MedSAM-Agent formulates interactive segmentation as autonomous multi-turn decision-making with reinforcement learning Liu et al. (2026). Our focus is human-controlled correction after initialization: connecting FN/FP diagnosis across heterogeneous mask sources to slice prioritization, corrective point proposal, and assessment of model responses. In this setting, point screening informs user decisions rather than automatically approving mask updates.

Correction is a sequential decision problem: the next action depends on the current mask, the observed model response, and user feedback. Edited points require retesting, while accepted mask updates require new diagnosis and recommendations. Rejected updates leave the current mask unchanged. These dependencies motivate agentic coordination of specialized tools through a shared case state, keeping guidance consistent with the current mask and user decisions.

We introduce MISeg-Agent, a human-in-the-loop framework that coordinates model selection, error diagnosis, point proposal, prompt review, and user-guided refinement. It selects a compatible pretrained model based on benchmarked segmentation accuracy, runtime, and GPU memory use, then converts predicted FN/FP regions into review locations and corrective points. Its Executable Review Agent uses nnInteractive Isensee et al. (2025) to compare each candidate's response with both the current mask and a mask-only output. Users choose which points to apply jointly and inspect the resulting mask before approving an update. To evaluate diagnosis across mask sources, we construct a benchmark using ground truth (GT) and rough masks from multiple pretrained models. The diagnosis input includes only the image and current mask, excluding source identity and internal features. Here, *model-agnostic* refers to this input interface, rather than guaranteed generalization to arbitrary unseen segmentation models.

Our contributions are threefold. (1) We formulate correction as a sequence of decisions based on the current mask, model responses, and user feedback, and implement a human-controlled review protocol that combines candidate testing with rules for updating diagnosis and recommendations. (2) We construct an error diagnosis benchmark using initial masks generated by multiple segmentation models. With the same validation inputs, source rankings differ between initialization and diagnostic supervision, supporting separate evaluation of these two roles. (3) We evaluate the complete workflow on 120 cases from seven CT datasets and report 360 participant–case sessions from three participants, measuring final-mask accuracy, voxel-wise error-diagnosis performance, and explicitly recorded review activity.

2 METHOD: MISEG-AGENT

Let $X \in \mathbb{R}^{N_z \times N_y \times N_x}$ be a volume and q a target request. MISeg-Agent aims to obtain a final user-approved mask $M^\star$ through benchmark-informed model selection and targeted error correction. It first selects a compatible pretrained segmentation model to produce an initial rough mask

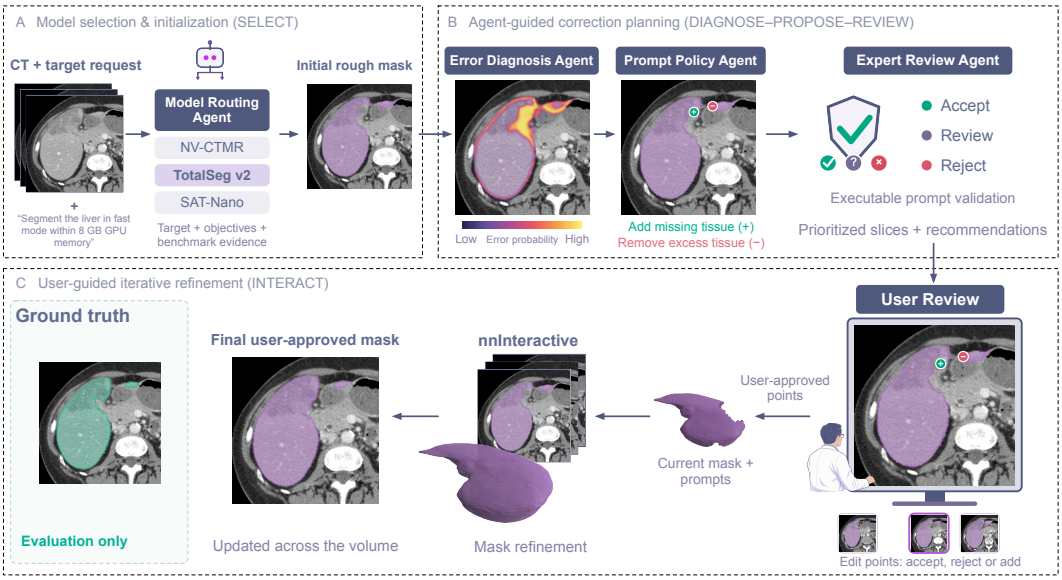


Figure 2: Overview of MISeg-Agent. Given a volume and target request, MISeg-Agent selects a compatible pretrained model to produce an initial mask, diagnoses FN/FP errors, proposes candidate points, reviews them with nnInteractive, and commits only user-approved refinements. "Expert" denotes the Executable Review Agent.

M^0, then directs human review to suspected errors and proposed corrections, providing explicit locations and candidate interactions for user inspection. Fig. 2 summarizes the SELECT–DIAGNOSE–PROPOSE–REVIEW–INTERACT workflow. Selection considers target compatibility, benchmarked segmentation quality, runtime, and GPU memory; diagnosis and proposal identify suspected errors and candidate interventions; executable review assesses their responses before human-controlled refinement.

The Interactive Coordinator maintains the current mask version, diagnosis maps, candidate points, probe results, and interaction history. Here, *agentic* coordination involves invoking tools and selecting candidate points according to the current case state. Uncertain candidates are referred to the user, who controls point submission and mask updates. The diagnostic predictor supplies evidence, proposal and review policies select recommendations, and the user controls point submission and mask commitment. The MISeg-Agent retests edited points and reruns error diagnosis after each user-approved mask update. When no candidate points remain, the user decides whether to continue or stop. Appendix C.1 summarizes these rules for state updates and user approval.

2.1 BENCHMARK-INFORMED MODEL ROUTING

Candidate models. Let $\mathcal{M}$ index a collection of pretrained segmentation models $F_m \in \mathcal{M}$, including both general-purpose and task-specific models. Our library contains NV-Segment-CTMR, NV-Segment-CT/VISTA3D (He et al., 2024), SAT-Nano (Zhao et al., 2023), SegVol (Du et al., 2023), and TotalSegmentator v2 (Wasserthal et al., 2023). These models are selected for initialization because they produce target-specific 3D CT masks under fixed inference settings without user-provided spatial prompts. We fix their checkpoints, target mappings, and inference settings before evaluation, and filter candidates using fixed compatibility criteria, including availability, dimensionality, CT compatibility, target coverage, and output-mask compatibility.

Benchmark-informed selection. Compatible models differ in segmentation quality, runtime, and GPU memory usage. Deployment can impose limits on runtime and GPU memory; we therefore

include these measured costs with segmentation quality as selection criteria. Routing uses target-specific benchmark evidence to adapt model selection to these competing objectives, allowing the same framework to accommodate different requirements. Let c denote the structured target parsed from q, o the selection objective, and $\mathcal{M}_c \subseteq \mathcal{M}$ the compatible candidate set. The routing context r holds an optional dataset hint and a stable case identifier for deterministic tie-breaking. Model selection and execution are expressed as $m_0 = \mathcal{R}(c, o, \mathcal{M}_c, r; \mathcal{S})$ and $M^0 = F_{m_0}(X, c)$, where $\mathcal{R}$ is the routing policy and $\mathcal{S}$ contains the model capabilities and benchmark evidence.

For compatible models, the benchmark reports mean and 10th-percentile Dice, runtime, and GPU memory use. A compatible model selected by the user takes precedence; otherwise, accuracy prioritizes quality and then lower-tail Dice, while balanced selection considers lower-tail performance and resource use among models in the highest quality grade. Fast and low-memory settings minimize the requested resource within that grade unless the user permits an unrestricted choice. This quality constraint avoids treating a reduction in resource use as sufficient reason to select a lower-grade model by default. Further routing analysis is provided in Appendix A.

2.2 Benchmarking Model-Agnostic Error Diagnosis

Targeted correction requires identifying likely errors before users search for them. Because routing can select different pretrained segmentation models, diagnosis must operate on heterogeneous masks without access to the source model's internals. We therefore evaluate *model-agnostic voxel-wise error diagnosis* as a component of the correction workflow. Given an image volume X and an imperfect binary mask $\widehat{M}$ produced by a black-box segmentation model, the goal is to predict for each voxel whether it is correct, FN, or FP with respect to the GT mask Y.

We treat each requested structure as a binary segmentation task. For multi-organ requests, we can apply the procedure independently to each target or to a user-specified foreground label. For each case, we construct diagnosis targets from the pair $(Y, \widehat{M})$. The binary error target is $E = Y \oplus \widehat{M}$, where $\oplus$ denotes the voxel-wise exclusive OR operation, and $E(v) = 1$ indicates that voxel v is incorrectly labeled in $\widehat{M}$. To distinguish missing foreground from excessive foreground, we define the FN polarity target $B = Y \odot (1 - \widehat{M})$, with polarity supervision restricted to error voxels where $E = 1$. Here, $\odot$ denotes element-wise multiplication. Thus, error voxels with $B = 1$ correspond to FNs, while error voxels with $B = 0$ correspond to FPs. This separation supports two correction decisions: error localization identifies where review is needed, while polarity indicates whether to add or remove foreground.

We construct the benchmark by running a fixed library of candidate pretrained segmentation models on each dataset–target group, producing the collection

$$\mathcal{D}_{\mathrm{ED}} = \{(X_i, Y_i, \widehat{M}_{i,m}, m, g_i) : i = 1, \ldots, N,\ m \in \mathcal{M}_{g_i}\}, \tag{1}$$

where m denotes the model that generated the rough mask, g_i denotes the dataset–target group, and $\mathcal{M}_{g_i}$ is the set of compatible models for that group. We retain the model identity for splitting and subgroup evaluation but do not provide it as an input to the error diagnosis model. Diagnosis therefore relies on the image and current mask, matching the information available after pretrained model routing.

To assess how the training mask source affects diagnosis, we use rough masks from three representative pretrained segmentation models, denoted by $\mathcal{M}_{\mathrm{ED}}$. For each source model $m \in \mathcal{M}_{\mathrm{ED}}$, we train a separate error diagnosis model using masks generated by that source. All models are evaluated on the same held-out cases with rough masks from all three sources. This common evaluation enables comparison of training sources across heterogeneous segmentation outputs. We evaluate binary error localization and FN/FP identification. These voxel-level metrics assess diagnostic overlap, rather than directly measuring the recall of prioritized slices or the quality of subsequent corrections.

2.3 Error Diagnosis Agent

Using the benchmark targets defined above, the Error Diagnosis Agent estimates voxel-wise correctness and FN/FP scores with a residual 3D U-Net. Its input tensor Z combines two representations of the same CT volume, one emphasizing soft tissue and the other retaining a broader range of intensities, with the current mask and its signed distance map (Appendix B). The scores guide error

prioritization and point polarity: positive points target likely missing foreground, and negative points target likely excess foreground. Because the target request is not an input, the model relies on the current mask to indicate which anatomy is being corrected.

The model maps Z to error, FN-polarity, and auxiliary foreground logits (z_e, z_b, z_g). Positive temperature parameters (T_e, T_b) rescale the logits; the probability notation below denotes model estimates, not an empirical calibration guarantee. We factorize the diagnosis scores into overall error score and conditional FN polarity:

$$e = \sigma(z_e/T_e), \quad \beta = \sigma(z_b/T_b), \quad (\pi_{\mathrm{cor}}, \pi_{\mathrm{FN}}, \pi_{\mathrm{FP}}) = \big(1-e, e\beta, e(1-\beta)\big). \tag{2}$$

Here, σ is the logistic function; e and β parameterize normalized three-class scores before candidate filtering. For binary masks, true error polarity is determined by the current mask once an error is known: FN outside and FP inside. The learned polarity head supplies a soft score, while proposal support enforces these mask-consistent directions. The workflow uses these scores without assuming calibration or a benefit from learned polarity.

Training supervises three tasks: error localization with binary cross-entropy and focal Tversky losses, FN/FP polarity on error voxels with class-balanced binary cross-entropy, and auxiliary foreground prediction with binary cross-entropy and soft Dice losses. The auxiliary head is used only during training and discarded at inference; Appendix B describes the training objective and loss terms.

2.4 Diagnosis-Conditioned Prompt Proposal

The Prompt Policy Agent converts voxel-wise diagnostic scores into explicit choices of review slices, point locations, and prompt labels. Positive points target predicted FN regions to recover missing foreground, while negative points target predicted FP regions to remove excess foreground.

At round t, the policy identifies FN candidates outside the current foreground and FP candidates inside it, retaining voxels that satisfy score and margin thresholds:

$$\begin{aligned} \Omega^t_{\mathrm{FN}} &= \{v : M^t(v) = 0,\ \pi^t_{\mathrm{FN}}(v) \geq \tau,\ \pi^t_{\mathrm{FN}}(v) - \pi^t_{\mathrm{cor}}(v) \geq \delta\}, \\ \Omega^t_{\mathrm{FP}} &= \{v : M^t(v) = 1,\ \pi^t_{\mathrm{FP}}(v) \geq \tau,\ \pi^t_{\mathrm{FP}}(v) - \pi^t_{\mathrm{cor}}(v) \geq \delta\}. \end{aligned} \tag{3}$$

Here, v indexes voxels, τ is an error-score threshold, and δ is the score margin over the correct class.

For each 18-connected component $C \subset \Omega^t_l$ containing at least four voxels, where $l \in \{\mathrm{FN}, \mathrm{FP}\}$, the evaluated policy selects one interior representative:

$$v^*_C = \arg\max_{v \in C} \left[\lambda_p \pi^t_l(v) + \lambda_2 \bar{d}_{\mathrm{2D}}(v, C) + \lambda_3 \bar{d}_{\mathrm{3D}}(v, C)\right], \tag{4}$$

where $d_{\mathrm{2D}}(v, C)$ is the interior distance in the component's axial cross-section through v, and $d_{\mathrm{3D}}(v, C)$ is its 3D interior distance. Both use physical voxel spacing and are normalized separately by their maxima over C. The weights $(\lambda_p, \lambda_2, \lambda_3) = (0.55, 0.30, 0.15)$ balance diagnostic score and interior depth. The selected point's axial coordinate determines its review slice. The candidate set $\mathcal{C}^t$ contains all remaining representatives after suppressing recorded candidate identifiers and points less than 12 mm from a same-label point in the feedback or refinement history. Every remaining representative enters executable review; Appendix E.2 summarizes the ordering and history rules. An empty set returns control to the user for inspection or a recommendation to stop, without certifying an error-free mask.

2.5 Executable Prompt Review

Error diagnosis identifies where a mask may be wrong, but does not establish how the interactive model will respond to a proposed correction. The Executable Review Agent (abbreviated "Expert" in figures and tables) screens candidate points by using nnInteractive to test each candidate point in 3D. "Expert" is a software-module label; its verdict is a screening result based on the model response, not an anatomical assessment by a clinician.

For candidate point $p_i = (v_i, y_i)$, where v_i is the point location and $y_i \in \{+, -\}$ is the prompt label, we compare its mask-conditioned prediction with a mask-only control:

$$\overline{M}^t = \mathcal{G}_{\mathrm{nnI}}(X, M^t, \varnothing), \qquad \widetilde{M}^t_i = \mathcal{G}_{\mathrm{nnI}}(X, M^t, \{p_i\}). \tag{5}$$

Each candidate is probed separately from the same current mask. The two reference masks answer different questions: M^t is the current mask being edited, while $\overline{M}^t$ shows the refinement model's response without the candidate. For binary masks viewed as voxel arrays, the signed change decomposes exactly as

$$\widetilde{M}_i^t - M^t = \underbrace{\overline{M}^t - M^t}_{\text{mask-only change}} + \underbrace{\widetilde{M}_i^t - \overline{M}^t}_{\text{control-referenced change}} . \tag{6}$$

This identity explains why both references are retained. A comparison only with M^t combines mask-only and candidate-conditioned changes; a comparison only with $\overline{M}^t$ can overlook damage already introduced by mask-only refinement. We therefore examine both the full output and the control-referenced additions $A_i^t = \widetilde{M}_i^t \setminus \overline{M}^t$ and removals $R_i^t = \overline{M}^t \setminus \widetilde{M}_i^t$, treating masks as sets in these expressions. These are observed output differences under the probe configuration, not a decomposition of GT Dice or a causal estimate of prompt benefit.

Review uses both this incremental comparison and the full candidate output. *Local responsiveness* measures agreement with the point label in its neighborhood. *Directional support* assesses FN evidence on incremental additions for positive points and FP evidence on incremental removals for negative points; positive-point additions are restricted to the candidate-containing component. *Foreground preservation* limits incremental loss of diagnosis-supported foreground; negative points also require retention relative to the current mask. A candidate is not accepted solely because it passes the incremental-support checks; it must also pass the applicable full-mask checks. Structural checks assess the response in 3D; positive points also receive a point-only probe to examine the responding component relative to the current foreground. Appendices C.2 and E.2 summarize the decision rules and main checks.

An automatic `accept` verdict means that the candidate passes the configured checks for the current image, mask, and diagnosis. These checks reuse the diagnosis that generated the candidate and can inherit its errors; they are not independent evidence of anatomical correctness and do not guarantee the joint effect of accepted points. The user therefore previews the joint refinement before approving a mask update. The module assigns `accept`, `reject`, or `review`; the last defers a candidate for user inspection. Unavailable evidence is recorded separately. These screening verdicts are distinct from user approval of points and commitment of the joint output. Probe predictions are never committed automatically.

2.6 USER REVIEW AND MASK UPDATES

MISeg-Agent uses guidance from diagnosis, point proposal, and executable review to focus human inspection while preserving user authority over anatomical decisions. The interface presents the image, current mask, FN/FP overlays, prioritized slices, candidate points, and probe evidence, helping users navigate to suspected errors and assess proposed actions in their anatomical context. Users determine whether a highlighted region requires correction and whether the proposed point has an appropriate location and label. Agent suggestions can be inaccurate or incomplete, so users may accept, reject, relabel, or move candidates and add points for errors not covered by the proposals. An edited point is treated as an explicit human override and requires a new executable probe before it can be considered reviewed.

User-approved points are applied jointly with nnInteractive, and users inspect the resulting mask before accepting or rejecting the update. This review assesses their combined effect, which may differ from the responses to individual points. Rejection retains the current mask; acceptance commits the update. Diagnosis and recommendations are then refreshed for the new mask before further automated guidance is provided. Appendix C.1 provides further details of the interaction workflow.

3 EXPERIMENTS AND RESULTS

3.1 DATASETS AND IMPLEMENTATION DETAILS

We evaluate MISeg-Agent on seven public CT datasets: AortaSeg24 Imran et al. (2026), KiTS19 Heller et al. (2021), MSD Liver Simpson et al. (2019), MSD Pancreas Simpson et al. (2019), MSD Spleen Simpson et al. (2019), Pancreas-CT Roth et al. (2016), and SLIVER07 Van Ginneken

et al. (2007), comprising 1,194 cases in total. For training the error-diagnosis model, 1,074 cases are used for training and 120 cases are reserved for validation. We evaluate one binary foreground per volume. KiTS19, MSD Liver, and MSD Pancreas include organ and tumor labels; AortaSeg24 includes main-aorta Zones 0–9 but excludes branch and iliac classes; the other datasets use the organ label. Appendices C.3 and B give the exact label mapping and summarize preprocessing and training settings. Inference, runtime, and GPU-memory evaluation use the same 120-case validation cohort on an NVIDIA GeForce RTX 4090 Laptop GPU with 16 GB of VRAM. We used `gpt-5.6-terra` for the language-model component; the framework also accepts other compatible models selected by the user.

Table 1: Segmentation Dice across seven CT datasets. Cells show mean Dice and ± standard deviation on separate lines; standard deviations use smaller type. Overall MISeg-Agent summarizes 360 participant–case sessions; automatic and point-only results summarize 120 cases. These workflows differ in initialization and human input. NV-CTMR: NV-Segment-CTMR; VISTA: NV-Segment-CT/VISTA3D; Panc.: pancreas.

Model	Aorta Seg24	KiTS19	MSD Liver	MSD Panc.	MSD Spleen	Panc. CT	SLIVER 07	All
Cases	13	28	18	39	7	12	3	120
Automatic initialization								
TotalSeg. v2	0.871 ±0.040	0.855 ±0.120	0.968 ±0.011	0.722 ±0.196	0.952 ±0.009	0.755 ±0.174	0.968 ±0.007	0.829 ±0.166
NV-CTMR	0.837 ±0.051	0.918 ±0.094	0.970 ±0.009	0.912 ±0.021	0.963 ±0.007	0.877 ±0.016	**0.973 ±0.007**	0.915 ±0.063
SAT-Nano	0.818 ±0.040	0.952 ±0.014	0.965 ±0.009	0.875 ±0.032	0.957 ±0.005	0.841 ±0.066	0.967 ±0.003	0.904 ±0.062
VISTA	0.859 ±0.038	0.885 ±0.107	0.971 ±0.009	0.871 ±0.051	0.960 ±0.004	0.864 ±0.022	**0.973 ±0.005**	0.895 ±0.073
SegVol	0.844 ±0.057	0.496 ±0.399	0.946 ±0.050	0.796 ±0.127	0.959 ±0.007	0.814 ±0.038	**0.973 ±0.005**	0.769 ±0.262
Unfiltered points; no initial mask or human editing								
ScribblePrompt	0.004 ±0.005	0.070 ±0.041	0.030 ±0.028	0.014 ±0.024	0.012 ±0.025	0.006 ±0.006	0.033 ±0.056	0.028 ±0.037
nnInteractive	0.072 ±0.161	0.608 ±0.284	0.734 ±0.369	0.197 ±0.187	0.306 ±0.428	0.120 ±0.235	0.643 ±0.554	0.369 ±0.364
Human-guided complete workflow								
MISeg-Agent	**0.916 ±0.150**	**0.955 ±0.130**	**0.972 ±0.008**	**0.931 ±0.180**	**0.965 ±0.008**	**0.920 ±0.190**	0.972 ±0.008	**0.943 ±0.144**

3.2 RESULTS

We evaluate three aspects: final masks under the specified workflows, voxel-wise Dice for predicted error regions, and recorded activity during human interaction. The 120-case cohort is used for validation; these descriptive comparisons do not isolate the effect of the agentic organization. Appendix C.3 maps each result to its protocol and evaluation unit.

Segmentation Performance. The five pretrained initializers run under fixed inference settings. The ScribblePrompt and nnInteractive baselines receive the original agent-generated points without Expert filtering, human editing, or rough-mask initialization. These point sets contain 20.27 ± 30.13 generated points per case (Appendix D). In MISeg-Agent, the framework selects a pretrained model to generate the initial mask. It then refines this mask using reviewed points, with updates committed only after human approval.

The strongest initializer in Table 1 varies across targets. MISeg-Agent reports 0.943 ± 0.144 overall, compared with 0.915 ± 0.063 for NV-Segment-CTMR, and the highest mean on six of seven datasets. Differences from the best automatic model are largest on AortaSeg24 and Pancreas-CT and smaller on MSD Liver and Spleen. SLIVER07 scores are 0.972 for MISeg-Agent and 0.973 for the best baseline. These describe different workflows, with routing and human-input contributions combined. With unfiltered agent-generated points and no rough-mask prior, ScribblePrompt and nnInteractive obtain 0.028 ± 0.037 and 0.369 ± 0.364, respectively. Under the separate participant-

supplied point-budget protocol in Appendix D, nnInteractive reaches 0.912 ± 0.057 with five points and 0.928 ± 0.041 with ten. These results show the dependence on prompt protocol; the unfiltered-point setting is not a general performance estimate for either interactive model.

Error Diagnosis Benchmark. We compare error-diagnosis models trained on three rough-mask sources under common validation inputs. Separate Error Diagnosis (ED) models are trained on masks from NV-CTMR, TotalSegmentator, and SegVol, which have descending overall segmentation Dice. Every ED model is evaluated on the same held-out collection containing masks from all three sources for the same 120 cases. The model receives only the image and rough mask, without the source model's identity. The pooled evaluation combines matched-source and cross-source inputs; each training source remains represented.

Table 2: Model-agnostic error-diagnosis performance under different rough-mask training sources.

ED training source	FN Dice	FP Dice	Macro Dice	Binary Dice
TotalSegmentator	0.420 ± 0.160	0.680 ± 0.140	0.550 ± 0.120	0.560 ± 0.150
NV-CTMR	0.440 ± 0.150	0.680 ± 0.130	0.560 ± 0.110	0.570 ± 0.140
SegVol	$\mathbf{0.550 \pm 0.130}$	$\mathbf{0.750 \pm 0.110}$	$\mathbf{0.650 \pm 0.100}$	$\mathbf{0.660 \pm 0.120}$

Table 2 reports FN Dice and FP Dice for missing and excess foreground, respectively. Macro Dice averages the two, $\text{Macro Dice} = 0.5(\text{FN Dice} + \text{FP Dice})$, while Binary Dice measures their union regardless of polarity. They quantify voxel-level error overlap, rather than downstream correction benefit. Initialization and diagnostic supervision are different selection problems. The SegVol-trained diagnosis model has the highest values on all four metrics, although SegVol has the lowest segmentation Dice among these sources. Thus, the observed initialization ranking does not transfer to choosing the training-mask source for error diagnosis in this cohort. MISeg-Agent treats these as separate choices: routing uses segmentation quality and runtime and GPU-memory measurements, while ED selection uses error-localization performance. Error prevalence, extent, and target composition remain possible explanations for the observed ranking.

Human Interaction and Review Activity. Three participants independently corrected the same 120 cases without access to reference annotations during interaction: a postdoctoral researcher (A), a doctoral student (B), and an undergraduate researcher with limited familiarity with medical images (C). All sessions followed the review workflow described in Appendix E.2. Table 3 reports the resulting final-mask accuracy and recorded interaction activity under this assisted annotation setting. The 120 cases contain 30,720 axial slices in total, based on the $256 \times 256 \times 256$ volumes used in the interaction interface. Across this shared case set, Participants A, B, and C explicitly confirmed 741, 654, and 734 reviewed slices, respectively (Appendix C.4). Across the 360 participant–case sessions, these 2,129 slice confirmations correspond to 5.91 per session and 10.09 per recorded review minute.

Table 3: Participant-level final-mask Dice and recorded interaction activity. Dice is mean ± standard deviation over 120 cases per participant or 360 participant–case sessions pooled. Slices/total reports explicitly confirmed review slices / working-grid slices (30,720 per participant; 92,160 pooled). Review min records intervals for inspection, point editing, and decisions; Slices/min divides confirmed slices by Review min. Points are manual additions; Accept and Reject count user decision actions. The Pooled row combines A–C.

User	Sessions	Dice ↑	Points	Accept	Reject	Slices / total	Review min	Slices/ min
A	120	0.957 ± 0.079	39	126	327	741 / 30,720	73.54	10.08
B	120	0.947 ± 0.148	24	103	427	654 / 30,720	61.64	10.61
C	120	0.925 ± 0.185	26	232	191	734 / 30,720	75.92	9.67
Pooled	360	0.943 ± 0.144	89	461	945	2,129 / 92,160	211.10	10.09

Annotation-time Context. For context, we report annotation times reported in related medical image annotation studies. For MSD Spleen, MONAI Label Diaz-Pinto et al. (2024) reports about 25 minutes per volume with a paintbrush tool and 12.5 minutes with a contouring tool. The IMVS user study Reen et al. (2026) reports manual annotation times of 8.11 and 73.53 minutes for MSD Pancreas and LiTS, reduced to 5.26 and 47.56 minutes with a copy-and-edit workflow. Across our 360 participant-case sessions, mean recorded human review and editing time was 35.18 seconds, with dataset-specific means of 19.34, 24.08, and 37.77 seconds for MSD Spleen, MSD Pancreas, and MSD Liver, respectively. These results suggest that MISeg-Agent can substantially reduce active human interaction for segmentation review and correction, supporting efficient human-in-the-loop annotation. The reported times reflect user review and editing activity (Appendix C.4); because annotation targets, completion criteria, and timing protocols differ across studies, these values provide context rather than a controlled comparison of annotation speed.

Case Study. Fig. 3 illustrates targeted correction in KiTS19 and Pancreas-CT. In KiTS19, diagnosis highlights a large missed target region. The panel follows proposed candidates through automated review to the point retained by the user; refinement recovers the missing structure and improves visual agreement with GT. In Pancreas-CT, diagnosis identifies suspected missing and excess foreground for positive and negative point proposals. Executable review informs user selection, and the approved refinement follows the GT more closely. These selected examples illustrate the workflow, rather than isolating component contributions or establishing typical-case performance.

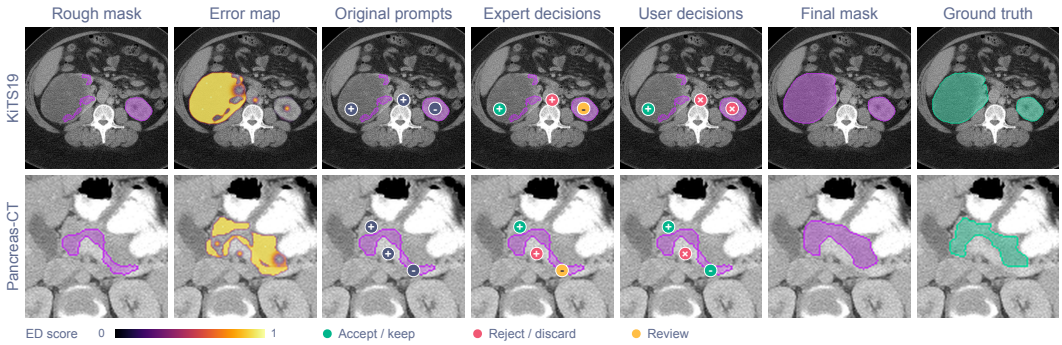


Figure 3: Qualitative case studies of MISeg-Agent on KiTS19 and Pancreas-CT. Each row shows a selected axial slice. Plus and minus signs denote foreground and background prompts. Green, red, and orange markers indicate accept, reject, and review, respectively; the color bar shows the error-diagnosis (ED) score.

4 Conclusions and Discussion

MISeg-Agent coordinates 3D mask correction through benchmark-informed model selection, error diagnosis, point proposal, executable review, and user-controlled refinement. It identifies suspected errors in the current mask, tests proposed points with an interactive model, and presents the observed responses for user assessment. After each user-approved mask update, the MISeg-Agent reruns error diagnosis and generates new recommendations before providing further guidance. In the evaluated CT cases, the workflow directed users to localized segmentation errors and supported targeted corrections, resulting in final masks with high overlap with the reference annotations.

The CT findings require validation on independent cohorts and other modalities. Differences in initialization and human input prevent attribution to individual components, and the three-participant study is too limited to draw strong conclusions about workload or user-dependent effects. Screening may inherit diagnostic errors, while favorable single-point probes do not guarantee the quality of their joint refinement; user preview therefore remains essential. Point prompts were evaluated; diagnosis after successive edits remains untested. Future work will pursue matched controls for the complete human-guided workflow, larger controlled user studies, additional interaction types beyond point prompts, and adaptive planning across successive edits.

AI USE STATEMENT

The authors used AI-based writing assistance tools only to check grammar, improve clarity, and refine the presentation of the manuscript. All technical ideas, method design, experiments, results, analyses, claims, and final writing decisions were reviewed, verified, and approved by the authors. AI tools were not used to generate experimental results, fabricate data, or make autonomous scientific conclusions.

REFERENCES

Andres Diaz-Pinto, Sachidanand Alle, Vishwesh Nath, Yucheng Tang, Alvin Ihsani, Muhammad Asad, et al. MONAI Label: A framework for AI-assisted interactive labeling of 3D medical images. *Medical Image Analysis*, 95:103207, 2024. doi: 10.1016/j.media.2024.103207.

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

Ali Hatamizadeh, Vishwesh Nath, Yucheng Tang, Dong Yang, Holger R Roth, and Daguang Xu. Swin UNETR: Swin Transformers for semantic segmentation of brain tumors in MRI images. In *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Interventions Brainlesion Workshop*, pp. 272–284, 2022a.

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*, pp. 1748–1758, 2022b.

Yufan He, Pengfei Guo, Yucheng Tang, Andriy Myronenko, Vishwesh Nath, Ziyue Xu, Dong Yang, Can Zhao, Benjamin Simon, Mason Belue, Stephanie Harmon, Baris Turkbey, Daguang Xu, and Wenqi Li. VISTA3D: A unified segmentation foundation model for 3D medical imaging. *arXiv preprint arXiv:2406.05285*, 2024.

Nicholas Heller, Fabian Isensee, Klaus H. Maier-Hein, Xiaoshuai Hou, Chunmei Xie, Fengyi Li, Yang Nan, Guangrui Mu, Zhiyong Lin, Miofei Han, et al. The state of the art in kidney and kidney tumor segmentation in contrast-enhanced CT imaging: Results of the KiTS19 challenge. *Medical Image Analysis*, 67:101821, 2021. doi: 10.1016/j.media.2020.101821.

Muhammad Imran, Jonathan R. Krebs, Vishal Balaji Sivaraman, Teng Zhang, Amarjeet Kumar, Walker R. Ueland, Michael J. Fassler, Chang Wang, Jinlong Huang, Xiao Sun, et al. Multiclass segmentation of aortic branches and zones in computed tomography angiography: The AortaSeg24 challenge. *Medical Image Analysis*, 113:104188, 2026. doi: 10.1016/j.media.2026.104188.

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.

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. doi: 10.48550/ARXIV.2602.03320.

Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In *Proceedings of the Fourth International Conference on 3D Vision*, pp. 565–571, 2016.

Abhilaksh Singh Reen, Kushal Borkar, and Ritvik Mahapatra. IMVS: Interactive medical volume segmentation with test-time adaptation: A new method for annotating radiology datasets. *arXiv preprint arXiv:2609.16775*, 2026.

Holger Roth, Amal Farag, Evrim B. Turkbey, Le Lu, Jiamin Liu, and Ronald M. Summers. Data from pancreas-CT, 2016. URL `https://wiki.cancerimagingarchive.net/x/eIlXAQ`.

Abdelrahman Shaker, Muhammad Maaz, Hanoona Rasheed, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. UNETR++: Delving into efficient and accurate 3D medical image segmentation. *IEEE Transactions on Medical Imaging*, 43(9):3377–3390, 2024.

Amber L Simpson, Michela Antonelli, Spyridon Bakas, et al. A large annotated medical image dataset for the development and evaluation of segmentation algorithms. *arXiv preprint arXiv:1902.09063*, 2019.

Bella Specktor-Fadida, Liat Ben-Sira, Dafna Ben-Bashat, and Leo Joskowicz. SegQC: A segmentation network-based framework for multi-metric segmentation quality control and segmentation error detection in volumetric medical images. *Medical Image Analysis*, 103:103638, 2025. doi: 10.1016/J.MEDIA.2025.103638.

Bram Van Ginneken, Tobias Heimann, and Martin Styner. 3D segmentation in the clinic: A grand challenge. In *Proceedings of the International Conference on Medical Image Computing and Computer-assisted Intervention Workshop on 3D Segmentation in the Clinic: A Grand Challenge*, volume 1, pp. 7–15, 2007.

Jakob Wasserthal, Hanns-Christian Breit, Manfred T. Meyer, Maurice Pradella, Daniel Hinck, Alexander W. Sauter, Tobias Heye, Daniel Boll, Joshy Cyriac, Shan Yang, Michael Bach, and Martin Segeroth. TotalSegmentator: Robust segmentation of 104 anatomic structures in CT images. *Radiology: Artificial Intelligence*, 5(5):e230024, 2023.

Hallee E Wong, Marianne Rakic, John Guttag, and Adrian V Dalca. ScribblePrompt: Fast and flexible interactive segmentation for any biomedical image. In *Proceedings of the European Conference on Computer Vision*, pp. 207–229, 2024.

Fahim A. Zaman, Lichun Zhang, Honghai Zhang, Milan Sonka, and Xiaodong Wu. Segmentation quality assessment by automated detection of erroneous surface regions in medical images. *Computers in Biology and Medicine*, 164:107324, 2023. doi: 10.1016/J.COMPBIOMED.2023.107324.

Ziheng Zhao, Yao Zhang, Chaoyi Wu, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. One model to rule them all: Towards universal segmentation for medical images with text prompts. *arXiv preprint arXiv:2312.17183*, 2023.

Hong-Yu Zhou, Jiansen Guo, Yinghao Zhang, Lequan Yu, Liansheng Wang, and Yizhou Yu. nnFormer: Interleaved Transformer for volumetric segmentation. *arXiv preprint arXiv:2109.03201*, 2021.

A Full Routing Analysis

The routing policy in main-paper Section 2.1 uses benchmark evidence summarized for each dataset–target group g. For a compatible model m, quality and lower-tail Dice grades are defined by

$$G^{\mathrm{qual}}_{m,g} = Q(\overline{D}_{m,g}), \qquad G^{\mathrm{rel}}_{m,g} = Q(D^{(0.1)}_{m,g}), \tag{A.1}$$

where $\overline{D}_{m,g}$ and $D^{(0.1)}_{m,g}$ are the mean and 10th-percentile Dice, respectively. For either statistic, $Q(d)$ is weak for $d < 0.75$, fair for $0.75 \le d < 0.85$, good for $0.85 \le d < 0.90$, strong for $0.90 \le d < 0.95$, and excellent for $d \ge 0.95$. These grades summarize average and lower-tail overlap, not reliability on a new case. Dataset-specific grades require a matching target and at least five benchmark cases; otherwise pooled target grades are used.

A compatible user-selected model takes precedence. Otherwise, accuracy prioritizes quality and then lower-tail grade; balanced keeps the highest quality grade and removes candidates dominated in lower-tail grade, runtime, and memory. Fast and low-memory objectives minimize the requested resource within the highest quality grade, unless the user permits an unrestricted choice. Ties are resolved deterministically using model identifiers and the case identifier. Table S1 reports the resulting assignments under the supplied dataset hints.

Table S1: Routing assignments by selection objective. Entries are model (case count); N is the number of cases. Unrestricted objectives remove the quality-grade constraint.

(a) Accuracy and balanced objectives

Dataset	N	Accuracy	Balanced
AortaSeg24	13	VISTA (8), TS (5)	VISTA (9), TS (4)
KiTS19	28	SAT (28)	SAT (28)
MSD Liver	18	VISTA (3), CTMR (3), SAT (8), TS (4)	CTMR (6), SAT (3), SV (4), TS (5)
MSD Pancreas	39	CTMR (39)	CTMR (39)
MSD Spleen	7	VISTA (3), CTMR (4)	CTMR (2), SV (5)
Pancreas-CT	12	CTMR (12)	CTMR (12)
SLIVER07	3	CTMR (1), SAT (2)	CTMR (1), SV (1), TS (1)

(b) Runtime and memory objectives

		Quality-constrained		Unrestricted	
Dataset	N	Fast	Low memory	Fast	Low memory
AortaSeg24	13	VISTA (13)	TS (13)	SV (13)	SV (13)
KiTS19	28	SAT (28)	SAT (28)	SV (28)	SV (28)
MSD Liver	18	SV (18)	SV (18)	SV (18)	SV (18)
MSD Pancreas	39	CTMR (39)	CTMR (39)	SV (39)	SV (39)
MSD Spleen	7	SV (7)	SV (7)	SV (7)	SV (7)
Pancreas-CT	12	CTMR (12)	CTMR (12)	SV (12)	SV (12)
SLIVER07	3	SV (3)	SV (3)	SV (3)	SV (3)

Models: CTMR = NV-Segment-CTMR; VISTA = NV-Segment-CT/VISTA3D; SAT = SAT-Nano; TS = TotalSegmentator v2; SV = SegVol.

Assignments change with both the target and the selection objective; no one model is chosen for every setting. Fig. S1 shows the runtime and memory measurements used alongside the quality grades. When benchmark evidence is insufficient, routing falls back to capability constraints or user selection.

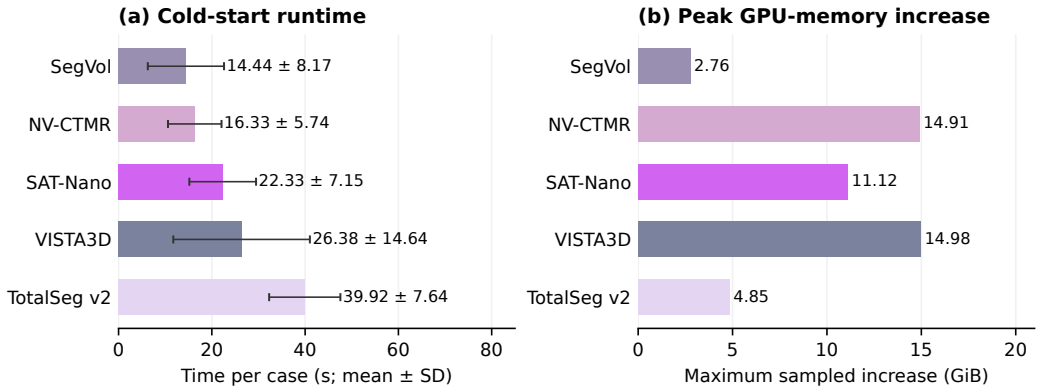


Figure S1: Runtime and GPU-memory measurements underlying model routing. Five models evaluated on 120 cases using an RTX 4090 Laptop GPU (16,376 MiB). Left: cold-start wall time, mean ± std, including loading and the full processing pipeline. Right: maximum sampled GPU-memory increase above baseline across cases under Windows WDDM, not per-process allocation. Lower is better.

B Error-Diagnosis Inputs and Training Objective

A 3D residual U-Net combines CT appearance with the current segmentation and its signed distance map as a four-channel input:

$$Z(X, M) = [W_{\text{soft}}(X),\ W_{\text{wide}}(X),\ M,\ \phi(M)]\,, \tag{B.1}$$

where W_{soft} and W_{wide} are fixed soft-tissue and wide-field CT windowing operators normalized to $[0, 1]$. The signed distance $\phi \in [-1, 1]$ is computed in physical units and clipped at a fixed radius, with negative values inside M and positive values outside. The target request is not part of this input, so targeted diagnosis relies on the current mask indicating the intended anatomy.

The targets E, B, and Y denote binary segmentation error, FN polarity, and foreground ground truth, respectively; (z_e, z_b, z_g) are the corresponding logits. The training objective combines error localization, polarity classification, and auxiliary foreground supervision:

$$\begin{aligned}\mathcal{L}_{\text{total}} = \lambda_e[\mathcal{L}_{\text{BCE}}(z_e, E) + \mathcal{L}_{\text{FT}}(z_e, E)] + \lambda_b \mathcal{L}_{\text{bal-BCE}}(z_b, B; E) \\ + \lambda_g[\mathcal{L}_{\text{BCE}}(z_g, Y) + \mathcal{L}_{\text{Dice}}(z_g, Y)]\end{aligned} \tag{B.2}$$

where $\mathcal{L}_{\text{BCE}}$ denotes binary cross-entropy loss and $\mathcal{L}_{\text{FT}}$ denotes focal Tversky loss. The nonnegative weights $(\lambda_e, \lambda_b, \lambda_g)$ sum to one and balance the three supervision objectives. The auxiliary term includes $\mathcal{L}_{\text{Dice}}$, the soft Dice loss between the foreground prediction $\sigma(z_g)$ and the GT mask Y. The term $\mathcal{L}_{\text{bal-BCE}}$ is class-balanced BCE restricted to voxels where $E = 1$. The auxiliary foreground head provides anatomical supervision during training and is discarded at inference. Training masks are outputs from candidate pretrained models. They represent initialization errors; errors after human edits may follow a different distribution. The models were trained on volumes resampled to $256 \times 256 \times 256$ voxels for 300 epochs using AdamW and cosine learning-rate decay. Main-paper Table 2 evaluates each trained model on masks from all three rough-mask sources, allowing initialization quality and diagnosis-training utility to be compared separately.

C Review Evidence and Evaluation Scope

C.1 Evidence validity and user control

A mask update changes the input on which later diagnosis and recommendations depend. The coordinator therefore links each diagnosis to a mask version and each point probe to both that version and the point's location and label. Editing a point requires a new probe; committing a mask preview requires new diagnosis; rejecting a preview retains the current mask. If no proposal is eligible, the

workflow returns control to the user rather than declaring the mask correct. These dependencies explain the state-dependent coordination described in the main paper.

Users select the points to submit, including any edited or manually added points. The interactive model applies the selected points jointly; its combined output is shown before the user commits or rejects the mask update. Individual point probes do not establish the quality of that joint output. The interaction record preserves mask revisions, point changes, and user decisions for review.

C.2 What executable review checks

Each candidate is tested from the current mask both with and without that point. The current mask anchors checks of the complete output, while the mask-only response shows changes produced by the interactive model without the candidate. Changes between the candidate and mask-only outputs are checked for the requested direction: additions should agree with predicted missing foreground for a positive point, and removals with predicted excess foreground for a negative point. Local response, preservation of diagnosis-supported foreground, and large structural changes are also checked. These are observed model responses, not ground-truth accuracy or a causal estimate of a point's benefit.

The automatic screen can accept a candidate, reject a decisive inconsistency, or defer uncertain evidence for user review; it does not commit a mask. Screening uses the image, current mask, diagnosis, and model responses, without GT. The participant configuration is summarized in Appendix E.2.

C.3 Targets and evaluation units

Each evaluated volume has one binary foreground. KiTS19, MSD Liver, and MSD Pancreas combine organ and tumor labels (IDs 1 and 2); AortaSeg24 combines main-aorta Zones 0–9 while excluding branch and iliac classes. MSD Spleen, Pancreas-CT, and SLIVER07 use the organ label. These definitions apply to overlap evaluation and participant review.

Table S2 separates results that use different inputs and evaluation units. Main-paper Fig. 1(b) contrasts the initial mask with nnInteractive refinement without points, with all generated points, with a count-matched unscreened subset, and with automatically screened points. The MISeg-Agent rows in main-paper Tables 1 and 3 instead summarize the human-guided workflow.

Table S2: Inputs and evaluation units for the reported results. The same 120 cases recur across several protocols; the 360 participant–case sessions are not 360 independent patients.

Result	**Input and intervention**	**Evaluation unit**
Main Table 1: initializers	Fixed pretrained model; no points or human edits	120 cases
Main Table 1: point-only models	Original generated points; no initial mask or human edits	120 cases
Main Fig. 1(b): automatic refinement	Common rough prior; mask-only, all points, or count-matched unscreened and screened points	120 cases
Main Tables 1 and 3: MISeg-Agent	Routed prior; participant-selected or added points; user-approved output	360 sessions on 120 cases
Main Table 2: diagnosis	Models trained on three rough-mask sources; pooled evaluation on masks from all three	Same 120 cases
Table S3: point-budget references	Participant-supplied points; no initial mask	120 cases

C.4 Participant activity and timing

Table 3 counts distinct axial slices reviewed within each participant–case session. Each participant's 120 resampled working-grid volumes contain $120 \times 256 = 30,720$ axial slices, and the pooled denominator of $92,160$ repeats the same cases across three participants. Points are manual additions; Accept and Reject count user decisions, which may cover multiple candidates. The pooled Dice summaries use 360 participant–case observations on 120 cases.

Review min sums recorded intervals for inspection, point editing, and decisions. The participant rows total 211.10 minutes, or 35.18 seconds per session across 360 sessions; 2,129 slice confirmations divided by 211.10 minutes gives the pooled 10.09 Slices/min in main-paper Table 3. These intervals characterize logged interaction rather than end-to-end completion time.

D Interactive Point References

Table S3 reports interactive segmentation with points manually placed by participants based on their assessment of each case. The fixed budgets are one, five, or ten points per volume; the per-slice setting uses one participant-supplied point per target slice. The models receive CT and point coordinates/labels, without an initial mask. These reference settings differ from the original generated points used by the point-only baselines in main-paper Table 1.

Table S3: Interactive segmentation with participant-supplied points. Dice and Points used are mean $\pm$ standard deviation over 120 cases. The settings differ in point budget and are separate from the MISeg-Agent correction workflow.

Model	N	Point budget	Dice ↑	Points used
ScribblePrompt	120	1	0.025 ± 0.029	1.000 ± 0.000
ScribblePrompt	120	5	0.094 ± 0.052	5.000 ± 0.000
ScribblePrompt	120	10	0.161 ± 0.069	10.000 ± 0.000
nnInteractive	120	1	0.689 ± 0.233	1.000 ± 0.000
nnInteractive	120	5	0.912 ± 0.057	5.000 ± 0.000
nnInteractive	120	10	0.928 ± 0.041	10.000 ± 0.000
ScribblePrompt	120	1 / target slice	0.717 ± 0.125	95.850 ± 31.620

In these settings, nnInteractive reaches 0.912 Dice with five participant-supplied points, while ScribblePrompt reaches 0.717 under the per-slice protocol, averaging 95.85 points per volume; these are different budgets and prompt protocols. For the main-paper point-only baselines, the original candidate sets contain 2,432 generated points across 120 cases (806 positive and 1,626 negative; 20.27 ± 30.13 points per case), without rough-mask initialization or human editing.

E Screening Context for the Reported Workflows

E.1 Automatic screen in main-paper Fig. 1(b)

The screened row uses a fixed automatic rule applied to candidate points from the same pool as both unscreened rows. Each point is probed separately from the common rough mask; the rule checks point-label response, whole-mask plausibility, and FN/FP support before accepting it for submission. Accepted points are then submitted together in their recorded order; if none are accepted, the initial mask is retained. The all-points and screened rows submit different numbers of points; the matched-budget unscreened row submits the first k candidates in proposal order, where k is the screened count for that case. This contrast evaluates point selection at a fixed count, but the rule does not include the mask-only incremental checks of the participant workflow.

E.2 Participant screening and proposal order

The participant workflow uses the candidate construction in main-paper Eq. 4. Eligible components are prioritized using diagnostic score and component size; previously considered same-label points within 12 mm are suppressed. All remaining representatives enter review, without a top-k cutoff.

For each point, the policy checks local response and whole-mask plausibility against the current mask, then checks incremental additions or removals against the mask-only response and considers supported-foreground loss. Incremental directional support below 0.90 for positive points or 0.98 for negative points, and supported-foreground loss above 0.005, flag uncertain responses for user review. A positive-point response is also checked for spatial separation from the current foreground. The automatic verdict is `accept` when the required checks pass, `reject` for clear inconsistencies in the model response, or `review` when evidence remains uncertain. These verdicts concern the output obtained with a single point; the user still chooses submitted points and approves or rejects their joint refinement.

F Additional Qualitative Case Studies

Figure S2 extends the KiTS19 and Pancreas-CT examples in main-paper Fig. 3 with cases from AortaSeg24, MSD Liver, MSD Pancreas, MSD Spleen, and SLIVER07. The panel follows rough masks through diagnosis and interaction to final masks and GT. Selected slices illustrate the workflow, not success frequency or whole-volume correctness.

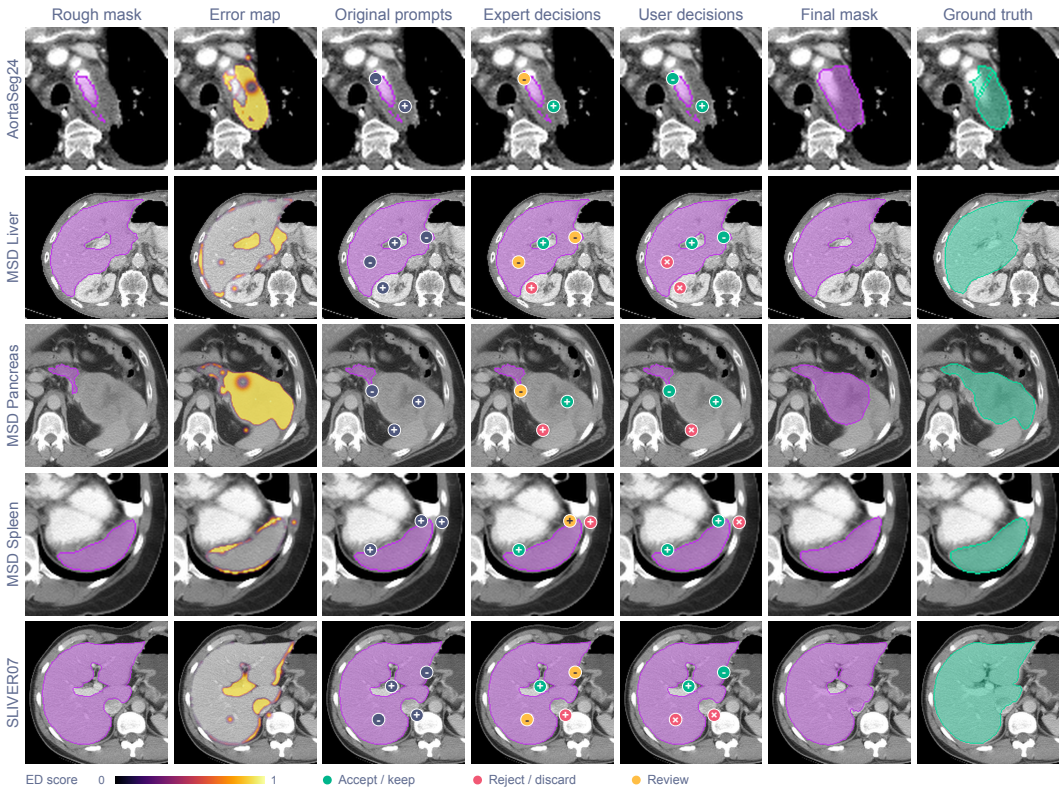


Figure S2: Additional qualitative case studies from the five CT datasets not shown in the main-paper case-study panel.