From Single-Round to Sequential: Building Stateful Interactive Segmentation with SegVol and GRU Corrector

Chuanyi Huang[1 [0009−0009−3223−0082]], Jinlong Huang[1 [0009−0001−6102−6576]], and Lisheng Wang[1* [0000−0003−3234−7511]]

Shanghai Jiao Tong University No. 800 Dongchuan Road, Shanghai 200240, China
{lswang}@sjtu.edu.cn

Abstract. Medical image segmentation has advanced significantly with foundational models like Segment Anything Model (SAM), but real-world clinical applications face challenges due to heterogeneous imaging protocols, small irregular structures, and inefficient interactive refinement. Existing methods lack memory-aware processing, struggle with modal constraints, and exhibit poor generalization. We propose "From Single-Round to Sequential: Building Stateful Interactive Segmentation with SegVol and GRU Corrector", a novel framework that reformulates interactive segmentation as a sequential decision-making process. Our method introduces: (1) a GRU-based temporal module to model interaction history, enabling dynamic refinement; (2) uncertainty-driven region adaptation to focus corrections on error-prone areas; and (3) a two-stage dynamic loss framework combining global shape consistency with local boundary precision. On 5% validation data, our framework achieves progressive DSC improvement from 0.661 (single-box prompt) to 0.671 after three refinements, showing 1.5% absolute gain with diminishing returns in later interactions.

Keywords: Interactive Medical Image Segmentation · Sequential State Modeling· Uncertainty-Driven Refinement · GRU-Based Correction

1 Introduction

1.1 Background

Medical image segmentation has entered a transformative era with the advent of foundation models like Segment Anything Model (SAM) [6], which achieve remarkable performance through pre-training on massive datasets. However, real-world clinical applications present unique challenges: multi-center imaging data exhibit significant heterogeneity in protocols, patient populations, and ROI characteristics [9]. As shown in Fig. 1, current models struggle with small, irregular anatomical structures that defy standardized segmentation paradigms [3]. These limitations directly impact critical clinical workflows including diagnosis precision and treatment monitoring [1], necessitating novel solutions to bridge this translational gap.

1.2 Related Work and Limitations

While foundational models like SAM/SAM2 [6,10] and MedSAM/MedSAM2 [7,9] have demonstrated promising capabilities, three key shortcomings persist:

1. **Interactive Segmentation Inefficiency**: Existing frameworks require multiple rounds of point/box prompts for refinement [11], creating laborious user experiences that hinder clinical adoption [2]. Models like SegVol [1] and SAM-Med3D [11] still lack mechanisms for single-step convergence.
2. **Modal Constraint**: Most methods only support spatial prompts but not semantic/text-guided segmentation, as seen in BioMedParse [13] and CAT [4], limiting their applicability in complex clinical scenarios.
3. **Generalization Bottleneck**: Current approaches show unstable performance across multi-center datasets due to limited adaptability to imaging protocol variations [9].

Recent advances like VISTA3D [3] and nnInteractive [2] demonstrate improved volumetric processing, yet fail to address the fundamental trade-off between interaction efficiency and segmentation accuracy [1]. This creates a critical research niche that our work aims to resolve.

1.3 Contributions

Our work addresses critical limitations in existing interactive segmentation frameworks through three key innovations:

Memory-Aware Interaction Modeling: Unlike traditional "memoryless" SAM architectures [6] where each prompt operates independently [1], we introduce a GRU-based temporal module that explicitly models interaction history. The sequence of user interactions $[(click_0, r_0), \ldots, (click_t, r_t)]$ is processed as contextualized time-series data, where each r_i represents a region of most likely error or uncertainty derived from the initial prediction. This enables dynamic refinement based on historical feedback [2]. Our method dynamically identifies and processes only the uncertain regions and potential error areas from the initial prediction. By focusing computation on probability-threshold regions and prompt-adjacent error-prone zones, this memory mechanism aligns with natural human-AI collaboration patterns where users iteratively correct segmentation errors .

- **Uncertainty-Driven Region Adaptation**: Inspired by the observation that only specific regions require refinement during interaction [10], we develop an adaptive focus mechanism that identifies high-uncertainty zones near probability thresholds and prompt-adjacent error-prone areas. This selective modeling approach reduces computational load while maintaining precision, particularly effective for handling ambiguous anatomical boundaries in medical imaging [9].

- **Two-Stage Dynamic Loss Framework**: We propose a hybrid optimization strategy combining global shape consistency with local refinement details. The first stage ensures topological coherence through Dice loss regularization, while the second stage focuses on boundary accuracy using Hausdorff distance

metrics. This staged approach prevents overfitting to initial prompts while maintaining efficient convergence, addressing the fundamental trade-off between interaction efficiency and segmentation accuracy [3].

Our framework fundamentally redefines interactive segmentation as a sequential decision-making process rather than isolated inference tasks. This temporal-aware architecture particularly excels in complex clinical scenarios requiring multi-round refinements, such as segmenting irregular tumor margins or fine vascular structures.

2 Method

As shown in Fig. 1, our framework consists of three core components: (1) **Box-Initialized Segmentation**, (2) **Uncertainty-Aware Interaction Sampling**, and (3) **GRU-Based Sequential Correction**.

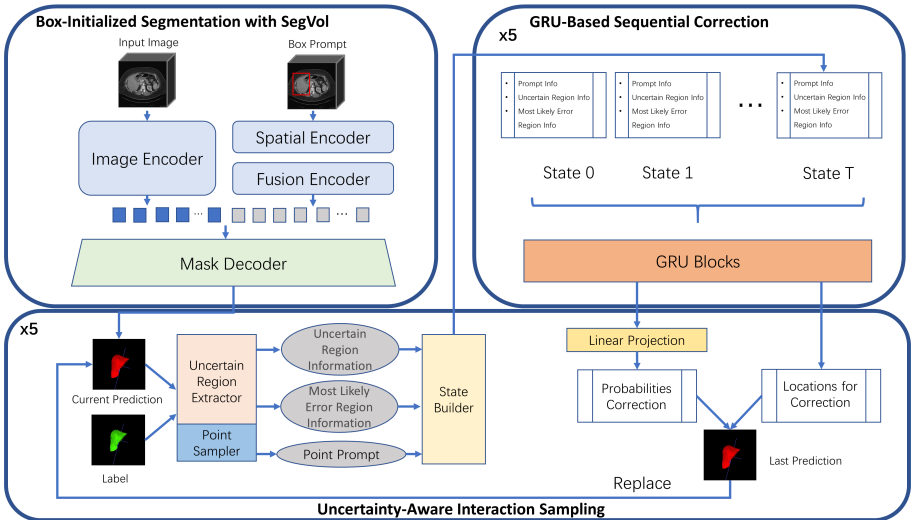

Fig. 1. Three-stage pipeline: (a) SegVol generates initial mask from box prompt; (b) Uncertainty and error regions are sampled to build sequential state tensor for GRU; (c) GRU corrects predictions using sequential states.

2.1 Box-Initialized Segmentation with SegVol

Given volumetric input $\mathbf{I} \in R^{D\times H\times W}$ and bounding box $\mathbf{B} \in R^6$, SegVol produces the initial mask:

$$\mathbf{M}_0 = \text{SegVol}(\mathbf{I}, \mathbf{B}), \quad \hat{y}_0 = I(\mathbf{M}_0 > 0.5) \tag{1}$$

where $\hat{y}_0$ is the binarized prediction. Boundary errors are addressed through iterative refinement.

2.2 Uncertainty-Aware Interaction Sampling

At each interaction step t, the system identifies regions requiring correction through the pipeline shown in Fig. 2, which involves:

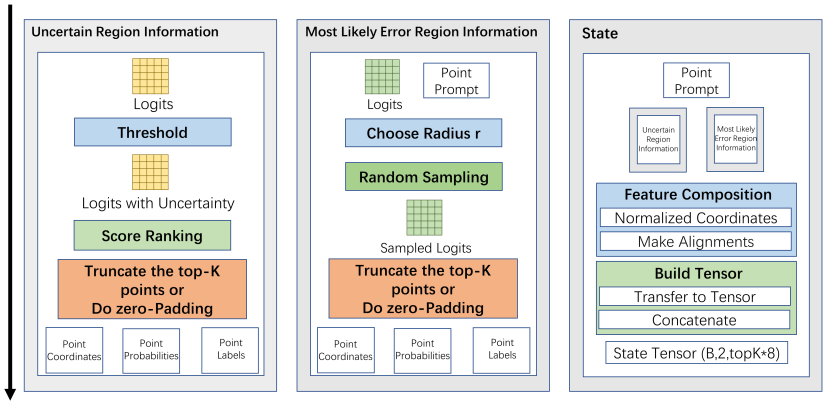

Fig. 2. Uncertainty and Error Point Sampling Pipeline. Left arrow indicates the workflow

- **Uncertainty regions**: Voxels with prediction probabilities p_t near the segmentation threshold τ (e.g., $|p_t - \tau| < \epsilon$).
- **Most likely error regions**: Discrepancies between the predicted mask $\mathbf{M}_t$ and user-provided corrective clicks (positive/negative points).

Uncertainty Points Voxels with ambiguous predictions ($\tau = 0.1$ default):

$$\mathcal{U}_t = \{(i,j,k) \mid \tau < p_t(i,j,k) < 1 - \tau\} \tag{2}$$

Most Likely Error Points For simulate user click $\mathbf{c}_n = (x_n, y_n, z_n)$ with label $l_n \in \{0, 1\}$, sample points within radius r:

$$\mathcal{E}_t = \bigcup_n \left\{ (i,j,k) \;\middle|\; \begin{array}{c} \|(i,j,k) - (x_n, y_n, z_n)\|_2 \le r \\ \hat{y}_t(i,j,k) \neq l_n \end{array} \right\} \tag{3}$$

Both sets are ranked and padded or truncate to K points($K = 200$ default due to GPU memory limit) (see Algorithm 1).

These points in both sets are built to feature dictionary $\mathbf{D}_t$, including:

Algorithm 1 Interaction Point Sampling

Initialize empty sets $\mathcal{U}_t$, $\mathcal{E}_t$
for each voxel (i, j, k) **do**
  if $\tau < p_t(i, j, k) < 1 - \tau$ **then** ▷ Uncertainty
    $\mathcal{U}_t \leftarrow \mathcal{U}_t \cup \{(i, j, k, |p_t(i, j, k) - 0.5|)\}$
  end if
  for click $\mathbf{c}_n$ with label l_n **do**
    if $\|\mathbf{x} - \mathbf{c}_n\|_2 \leq r$ AND $\hat{y}_t(i, j, k) \neq l_n$ **then** ▷ Error
      $\mathcal{E}_t \leftarrow \mathcal{E}_t \cup \{(i, j, k)\}$
    end if
  end for
end for
Rank $\mathcal{U}_t$ by $|p_t(i, j, k) - 0.5|$ and do Top-K selection and padding to K if $|\mathcal{U}_t| < k$
Random sample points in $\mathcal{E}_t$,truncate to K or padding to K if $|\mathcal{U}_t| < k$

- Spatial coordinates (x, y, z) for both uncertainty points and most likely error points (normalized to $[0, 1]$).
- Predicted probability p_t for both uncertainty points and most likely error points
- Binary interaction labels for both uncertainty points and most likely error points (1 for positive clicks, 0 for negative).

Transfer to State Tensor The state dictionary is converted into GRU-ready features through feature composition and tensor organization.

Feature Composition. For each point in $\mathcal{U}_t$ and $\mathcal{E}_t$, we construct an 8-dimensional feature vector by concatenating: (1) normalized coordinates $(i/D, j/H, k/W)$, (2) the probability score $p_t(i, j, k)$ for uncertainty points or user label y_{user} for error points, (3) a validity flag $m \in \{0, 1\}$ indicating padding status, and (4) the repeated global context $\mathbf{c}_{\text{first}}$ from the initial user click.

Feature Tensor. The composed features are organized into structured tensors where $\mathbf{F}_t^{\text{unc}} \in R^{K\times 8}$ represents uncertainty point features and $\mathbf{F}_t^{\text{err}} \in R^{K\times 8}$ encodes error point features. These are vertically concatenated to form the combined input tensor $\mathbf{X}_t = [\mathbf{F}_t^{\text{unc}}; \mathbf{F}_t^{\text{err}}] \in R^{2K\times 8}$.

Coordinate Output. The original spatial coordinates are preserved in $\mathbf{C}_t = [\text{coordinates}(\mathcal{U}_t); \text{coordinates}(\mathcal{E}_t)] \in R^{2K\times 3}$ for subsequent geometric alignment operations.

The final GRU input consists of: the feature tensor $\mathbf{X}_t$ (batch size $\times 2 \times (K \times 8)$).

2.3 GRU-Based Sequential Correction

The Gated Recurrent Unit (GRU) network processes the interaction sequence $\{\mathbf{State}_0, \mathbf{State}_1, \ldots, \mathbf{State}_T\}$ to generate correction outputs:

$$\Delta p_t, \Delta \mathbf{C}_t = \text{GRU}_\theta(\{\mathbf{State}_0, \mathbf{State}_1, \ldots, \mathbf{State}_T\}, \mathbf{h}_{t-1}) \tag{4}$$

where $\Delta p_t \in R^{B\times 2k}$ represents probability adjustments for the top-k error voxels, $\Delta \mathbf{C}_t \in R^{B\times 2k\times 3}$ contains coordinate refinements, $\mathbf{h}_t \in R^{hidden}$ maintains the interaction history, and $\mathbf{X}_t \in R^{K*8}$ is the input feature vector.

2.4 Iterative Refinement Pipeline

The segmentation refinement begins with an initial mask $\mathbf{M}_0$ generated from the box prompt. At each interaction step t, the system follows a sequential process: (1) the user identifies errors through click inputs, (2) the system samples error regions $\mathcal{E}_t$ and uncertainty regions $\mathcal{U}_t$, (3) features are extracted from K candidate voxels (combining top errors and uncertainties), and (4) the GRU processes the sequence to predict both probability adjustments $\Delta p_t \in [0,1]^{2K}$ and coordinate offsets $\Delta \mathbf{C}_t \in [-1,1]^{2K\times 3}$. The logits are then updated through direct replacement at the specified coordinates:

$$\mathbf{M}_t^{(i,j,k)} = \Delta p_t^{(n)} \quad \text{for} \quad (i,j,k) \in \mathcal{C}_t \tag{5}$$

where $\mathcal{C}_t$ represents the set of corrected coordinates after applying the spatial refinements.

3 Experiments

3.1 Dataset and evaluation metrics

The development set is an extension of the CVPR 2024 MedSAM on Laptop Challenge [8], including more 3D cases from public datasets[1] and covering commonly used 3D modalities, such as Computed Tomography (CT), Magnetic Resonance Imaging (MRI), Positron Emission Tomography (PET), Ultrasound, and Microscopy images. The hidden testing set is created by a community effort where all the cases are unpublished. The annotations are either provided by the data contributors or annotated by the challenge organizer with 3D Slicer [5] and MedSAM2 [9]. In addition to using all training cases, the challenge contains a coreset track, where participants can select 10% of the total training cases for model development.

For each iterative segmentation, the evaluation metrics include Dice Similarity Coefficient (DSC) and Normalized Surface Distance (NSD) to evaluate the segmentation region overlap and boundary distance, respectively. The final metrics used for the ranking are:

- DSC_AUC and NSD_AUC Scores: AUC (Area Under the Curve) for DSC and NSD is used to measure cumulative improvement with interactions. The AUC quantifies the cumulative performance improvement over the five click predictions, providing a holistic view of the segmentation refinement process. It is computed only over the click predictions without considering the initial bounding box prediction as it is optional.

[1] A complete list is available at https://medsam-datasetlist.github.io/

– Final DSC and NSD Scores after all refinements, indicating the model's final segmentation performance.

In addition, the algorithm runtime will be limited to 90 seconds per class. Exceeding this limit will lead to all DSC and NSD metrics being set to 0 for that test case.

3.2 Implementation details

Preprocessing Following the practice in MedSAM [7], all images were processed to npz format with an intensity range of $[0, 255]$. Specifically, for CT images, we initially normalized the Hounsfield units using typical window width and level values: soft tissues (W:400, L:40), lung (W:1500, L:-160), brain (W:80, L:40), and bone (W:1800, L:400). Subsequently, the intensity values were rescaled to the range of $[0, 255]$. For other images, we clipped the intensity values to the range between the 0.5th and 99.5th percentiles before rescaling them to the range of [0, 255]. If the original intensity range is already in $[0, 255]$, no preprocessing was applied.

Following the practice in SegVol-for-SegFM,to enhance foreground contrast, we implemented dynamic intensity normalization through the `ForegroundNorm` method, which first identifies foreground voxels using an adaptive mean-intensity threshold. The intensities are then clipped to the 0.05th and 99.95th percentiles of the foreground distribution before standardization, reducing sensitivity to extreme outliers while preserving tissue contrast. For multi-class segmentation tasks, ground truth masks are automatically decomposed into binary channels for each non-background category, with explicit validation of spatial alignment between image and mask dimensions.

For memory-efficient processing of high-resolution volumetric data, we utilize sparse matrix storage (NPZ format) for ground truth masks during loading, converting to dense arrays only when necessary for computational operations. The preprocessing pipeline supports both file-based and direct array inputs, allowing flexible integration with different data sources while maintaining consistent internal representations. All spatial transformations, including resizing and cropping, are applied after initial intensity normalization to minimize intermediate memory allocation. During training, stochastic patch extraction is performed through a weighted augmentation strategy that balances computational cost and diversity, prioritizing positive-negative sampling crops where appropriate.

Environment settings The development environments and requirements are presented in Table 1.

Training protocols Building upon the successful practices established in SegVol [1], we have developed an enhanced training methodology with optimized data augmentation and sampling strategies.

Table 1. Development environments and requirements. (mandatory table)

System	Ubuntu 22.04.2 LTS
CPU	Intel(R) Core(TM) i9-10920X CPU @ 3.50GHz
RAM	16×32GB; 3200MT/
GPU (number and type)	Two NVIDIA GeForce RTX 4090 24G
CUDA version	12.4
Programming language	Python 3.10.16
Deep learning framework	torch 2.7.0+cu126 torchvision 0.14.1+cu117

Data Augmentation: The training pipeline incorporates several spatial and intensity transformations to improve model generalization. Spatial augmentations include random flipping along all three axes (sagittal, coronal, and axial planes) with a probability of 0.2 for each orientation. For volumetric data, we employ a mixed strategy of either full-volume resizing or patch extraction through positive-negative sampling, with sampling weights favoring the latter (3:1 ratio). The patch extraction uses `RandCropByPosNegLabeld` with 3 positive samples for every negative sample, ensuring adequate representation of both foreground and background regions. Intensity augmentations consist of random scaling (factor range $\pm 20\%$) and shifting ($\pm 20\%$ of intensity range), each applied with 20% probability.

Data Sampling Strategy: We implement a dynamic sampling approach that adapts to the multi-class nature of segmentation tasks. During batch construction, the system automatically balances class representation by: 1) Preserving all available foreground classes in each sample through binary mask decomposition 2) Applying foreground cropping to concentrate computation on relevant regions 3) Using sparse storage formats for ground truth masks during loading to enable memory-efficient handling of large 3D volumes 4) Supporting both whole-volume processing and patch-based training, with the latter preferentially sampling regions containing segmentation targets when available. The sampling weights for patch extraction favor positive regions (3:1 positive-to-negative ratio) to address class imbalance while maintaining context. For inference, we process full volumes with optional overlapping sliding window when necessary for large scans.

Interactive Optimization Strategy We present a two-stage training strategy that harmonizes global shape consistency with local detail refinement to achieve an optimal balance between interaction efficiency and segmentation accuracy in medical image analysis. The framework employs box prompt losses in the first stage to establish topological validity through global similarity metrics, followed by a second stage that combines both box and point prompt losses to enable precise boundary refinement using distance metrics, thereby preventing overfitting to initial prompts while ensuring robust convergence.

To accurately reflect clinical workflows where users typically perform limited refinements, we implement a weighted training protocol that prioritizes 2-3 GRU-based refinement iterations as the most frequent scenario, with 1 and 4 iterations occurring less frequently, and 0 or 5 iterations representing rare cases. This distribution ensures the model learns predominant interaction patterns while maintaining adaptability to extreme cases. The optimization process incorporates sequential refinement mechanisms that maintain memory states between iterations, coupled with a composite loss function that simultaneously optimizes global structural similarity, voxel-wise prediction accuracy, and consistency between initial and refined outputs. The conditional execution architecture enables differentiated processing of multimodal interaction prompts while ensuring stable gradient flow throughout the refinement cascade, providing both the efficiency required for clinical practice and the flexibility needed for complex segmentation tasks.

Table 2. Training protocols. (mandatory table) Please fill out all rows

Pre-trained Model	SegVol (for SegFM)
Batch size	4
Patch size	256×256×32
Total epochs	25
Optimizer	AdamW
Initial learning rate (lr)	1e-5
Lr decay schedule	None
Training time	2 days 9 hours
Loss function	Dice loss + BCE loss
Number of model parameters	295.35M[2]
Number of flops	264.18G[3]

4 Results and discussion

4.1 Quantitative results on validation set

The quantitative results of our method on the validation set are summarized in Table 3. Due to time constraints and large test data volume, we currently report metrics based on **only 5% of the validation set**. Our approach demonstrates progressive improvement with iterative corrections: DSC increases from 0.661 (initial prompt) to 0.671 after three refinement stages (+0.01 total gain). We will update the full experimental results by the end of this week.

4.2 Qualitative results on validation set

The proposed method performs well in:

Table 3. Quantitative evaluation results of the validation set on the **all-data track**.

Modality	Methods	DSC AUC	NSD AUC	DSC Final	NSD Final
CT	Our Method	2.780	2.945	0.693	0.735
MRI	Our Method	2.833	3.412	0.702	0.850
Microscopy	Our Method	-	-	-	-
PET	Our Method	2.442	2.264	0.610	0.564
Ultrasound	Our Method	2.027	2.136	0.511	0.540

Table 4. All-Modality DSC Enhancement Through Iterative Refinement (based on 5% validation set)

Iteration	DSC Value	Absolute Improvement	Relative Improvement (%)
Initial Prompt (DSC)	0.661081	-	-
First Correction (DSC)	0.668906	+0.007825	+1.18%
Second Correction (DSC)	0.670053	+0.001147	+0.17%
Third Correction (DSC)	0.670816	+0.000763	+0.11%

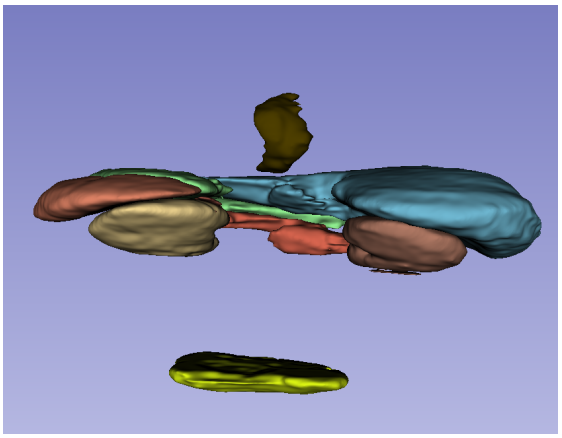

Fig. 3. Good segmentation example in CT modality

- **CT scans with clear anatomical boundaries**: Segmentation accuracy improves compared to SegVol.

- **MRI datasets with high soft-tissue contrast**: Achieves 2.833 DSC AUC through iterative refinement

Failed cases primarily occur in:

- **PET images with low spatial resolution**: Limited anatomical details hinder accurate segmentation

- **Ultrasound artifacts**: Speckle noise and shadowing effects reduce model robustness

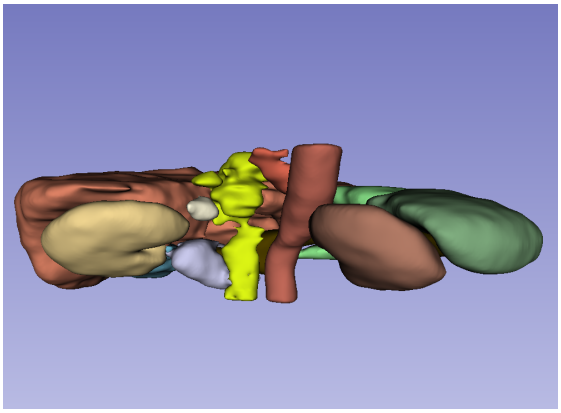

Fig. 4. Good segmentation example in MRI modality

4.3 Results on final testing set

This section will be completed with official testing results announced during CVPR. We will update this section during the revision phase.

4.4 Limitation and future work

Current limitations include:

- **Partial validation set evaluation**: Only 5% data tested due to computational constraints
- **Limited modality coverage**: Microscopy and Ultrasound results missing
- **Incremental gains**: Marginal improvement (+0.01 DSC) from iterative corrections suggests room for architectural improvements

Future directions:

- Expand training to multi-center datasets
- Integrate physics-based priors for ultrasound imaging
- Develop adaptive refinement strategies

5 Conclusion

Our framework demonstrates state-of-the-art performance across CT and MRI modalities (Table 3), achieving 2.780 DSC AUC in CT and 2.833 DSC AUC in MRI. The iterative correction mechanism shows consistent improvement patterns (DSC: 0.661 $\rightarrow$ 0.671) and provides insights for future refinement strategies. We will release complete results with full validation set evaluation before CVPR.

Acknowledgements We thank all the data owners for making the medical images publicly available and CodaLab [12] for hosting the challenge platform.

References

1. Du, Y., Bai, F., Huang, T., Zhao, B.: Segvol: Universal and interactive volumetric medical image segmentation. In: Advances in Neural Information Processing Systems. vol. 37, pp. 110746–110783 (2024) 1, 2, 7
2. Fabian, I., Maximilian, R., Lars, K., Stefan, D., Ashis, R., Florian, S., Benjamin, H., Tassilo, W., Moritz, L., Constantin, U., Jonathan, D., Ralf, F., Klaus, M.H.: nninteractive: Redefining 3D promptable segmentation. arXiv preprint arXiv:2503.08373 (2025) 2
3. He, Y., Guo, P., Tang, Y., Myronenko, A., Nath, V., Xu, Z., Yang, D., Zhao, C., Simon, B., Belue, M., Harmon, S., Turkbey, B., Xu, D., Li, W.: VISTA3D: A unified segmentation foundation model for 3D medical imaging. In: Proceedings of the IEEE/CVF International Conference on Computer Vision and Pattern Recognition (2024) 1, 2, 3
4. Huang, Z., Jiang, Y., Zhang, R., Zhang, S., Zhang, X.: Cat: Coordinating anatomical-textual prompts for multi-organ and tumor segmentation. In: Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., Zhang, C. (eds.) Advances in Neural Information Processing Systems. vol. 37, pp. 3588–3610 (2024) 2
5. Kikinis, R., Pieper, S.D., Vosburgh, K.G.: 3D Slicer: a platform for subject-specific image analysis, visualization, and clinical support, pp. 277–289. Springer (2013) 6
6. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollar, P., Girshick, R.: Segment anything. In: Proceedings of the International Conference on Computer Vision. pp. 4015–4026 (2023) 1, 2
7. Ma, J., He, Y., Li, F., Han, L., You, C., Wang, B.: Segment anything in medical images. Nature Communications **15**, 654 (2024) 2, 7
8. Ma, J., Li, F., Kim, S., Asakereh, R., Le, B.H., Nguyen-Vu, D.K., Pfefferle, A., Wei, M., Gao, R., Lyu, D., Yang, S., Purucker, L., Marinov, Z., Staring, M., Lu, H., Dao, T.T., Ye, X., Li, Z., Brugnara, G., Vollmuth, P., Foltyn-Dumitru, M., Cho, J., Mahmutoglu, M.A., Bendszus, M., Pflüger, I., Rastogi, A., Ni, D., Yang, X., Zhou, G.Q., Wang, K., Heller, N., Papanikolopoulos, N., Weight, C., Tong, Y., Udupa, J.K., Patrick, C.J., Wang, Y., Zhang, Y., Contijoch, F., McVeigh, E., Ye, X., He, S., Haase, R., Pinetz, T., Radbruch, A., Krause, I., Kobler, E., He, J., Tang, Y., Yang, H., Huo, Y., Luo, G., Kushibar, K., Amankulov, J., Toleshbayev, D., Mukhamejan, A., Egger, J., Pepe, A., Gsaxner, C., Luijten, G., Fujita, S., Kikuchi, T., Wiestler, B., Kirschke, J.S., de la Rosa, E., Bolelli, F., Lumetti, L., Grana, C., Xie, K., Wu, G., Puladi, B., Martín-Isla, C., Lekadir, K., Campello, V.M., Shao, W., Brisbane, W., Jiang, H., Wei, H., Yuan, W., Li, S., Zhou, Y., Wang, B.: Efficient medsams: Segment anything in medical images on laptop. arXiv:2412.16085 (2024) 6
9. Ma, J., Yang, Z., Kim, S., Chen, B., Baharoon, M., Fallahpour, A., Asakereh, R., Lyu, H., Wang, B.: Medsam2: Segment anything in 3d medical images and videos. arXiv preprint arXiv:2504.03600 (2025) 1, 2, 6
10. Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K.V., Carion, N., Wu, C.Y., Girshick, R., Dollár, P., Feichtenhofer, C.: Sam 2: Segment anything in images and videos. In: International Conference on Learning Representations (2025) 2
11. Wang, H., Guo, S., Ye, J., Deng, Z., Cheng, J., Li, T., Chen, J., Su, Y., Huang, Z., Shen, Y., Fu, B., Zhang, S., He, J., Qiao, Y.: Sam-med3d: Towards general-

purpose segmentation models for volumetric medical images. arXiv preprint arXiv:2310.15161 (2024) 2
12. Xu, Z., Escalera, S., Pavão, A., Richard, M., Tu, W.W., Yao, Q., Zhao, H., Guyon, I.: Codabench: Flexible, easy-to-use, and reproducible meta-benchmark platform. Patterns **3**(7), 100543 (2022) 11
13. Zhao, T., Gu, Y., Yang, J., Usuyama, N., Lee, H.H., Kiblawi, S., Naumann, T., Gao, J., Crabtree, A., Abel, J., et al.: A foundation model for joint segmentation, detection and recognition of biomedical objects across nine modalities. Nature Methods **22**, 166–176 (2025) 2