StreamRefine : Learning When to Stop by Occasionally Continuing in Medical Image Translation

Anonymous authors
Paper under double-blind review

Abstract

3D medical image translation synthesizes a target modality while preserving patient anatomy. Iterative refinement can improve predictions, but useful refinement varies across modality pairs and cases, motivating learned stopping. However, when stopping also truncates training trajectories, it hides the outcomes of later refinements and removes their direct supervision. Poorly trained later refinements can then reinforce premature stopping. In this work, we propose **StreamRefine**, which learns *when to stop without stopping refinement learning*. It models translation as an autoregressive stream of 3D refinement states conditioned on the source and preceding model predictions. Our ***Continue-to-Check*** training occasionally continues beyond proposed stops. These continuations train otherwise skipped refinements and teach a lightweight head whether further refinement is still worthwhile, balancing translation improvement, anatomical drift, and computation cost. At inference, the head decides when to stop without additional checks or ground truth targets. We evaluate **`StreamRefine`** across diverse 3D medical translation tasks for translation quality, anatomy-related performance, and computational efficiency. Further analyses show that occasional continuation strengthens later refinements, contributing to translation quality gains and better stopping decisions. Code is available at https://anonymous.4open.science/r/StreamRefine-36B4.

1 Introduction

Medical image translation should change modality-specific appearance while preserving patient anatomy (Zhan et al., 2024; Romoli et al., 2026). The amount of refinement needed to synthesize the target modality from a source volume varies across cases (Fig. 1(b)), with some translations requiring only a single pass (Rassmann et al., 2026), while subtle contrast changes or residual artifacts may benefit from repeated refinement (Wei et al., 2025; Song et al., 2026; Li et al., 2026). Iterative methods support such corrections (Bell et al., 2026; Zhao et al., 2026a), but a fixed number of steps can waste computation or miss useful refinements (Schusterbauer et al., 2026). A natural alternative lets the model decide, per case, when further refinement is no longer worthwhile. Importantly, whether further refinement remains worthwhile in medical translation depends not only on improving target appearance but also on avoiding unnecessary changes to patient anatomy (Zhou et al., 2026).

Learning such case-dependent stopping, however, creates a training challenge. Running every trajectory to the maximum horizon is costly, so a learned stop naturally truncates training as well. This saves computation, but later refinement states are then neither observed nor directly trained. They may therefore remain weak, causing the model to stop early partly because it has not learned to refine further, as illustrated in Fig. 1(c). Alternatively, training every case to the maximum horizon before learning to stop pays the full refinement cost. This creates a tension between efficient training and learning strong later refinements, leading to our central question:

Key Question

How can a model learn when to stop while still learning to make later refinements useful?

To answer this question, we let the model *occasionally continue* beyond its *proposed stops*, as shown in Fig. 1(a). Doing so reveals whether later refinements would improve the prediction or distort

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

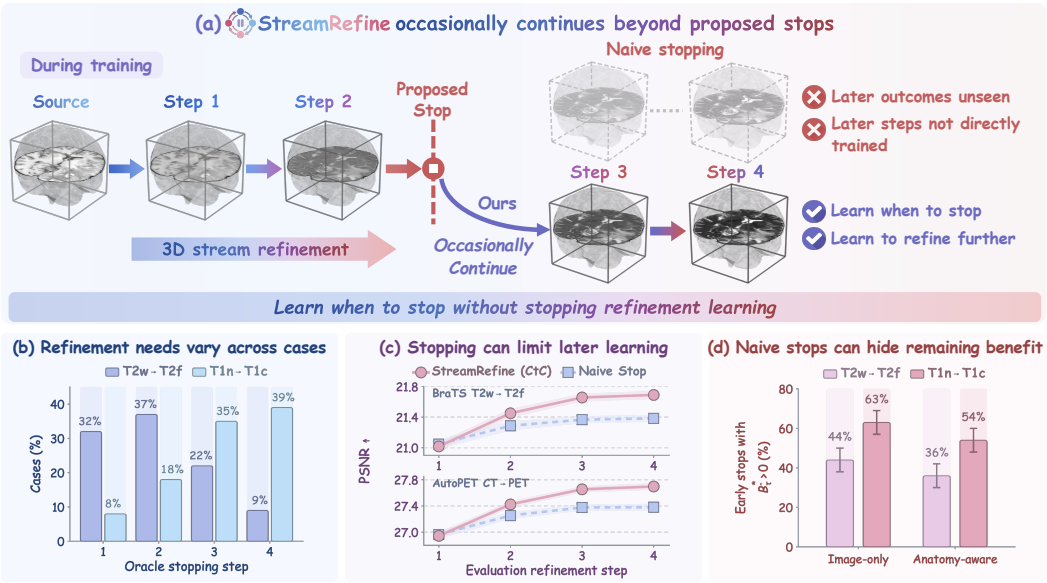


Figure 1: **(a)** During training, **`StreamRefine`** ***occasionally continues*** beyond *proposed stops* to learn when to *stop* while still training later refinements. **(b)** Full rollouts show that different cases prefer different stopping steps. **(c)** At the same evaluation steps, ***Continue-to-Check*** produces better later refinements than *Naive Stop*. **(d)** *Naive stopping* can terminate while useful refinement still remains (Experimental details in Appendix E.1).

anatomy (Fig. 1(d)), providing feedback on whether the model should have *stopped*. At the same time, the continued states receive supervision from the paired target, allowing the model to learn refinements that would otherwise be skipped. The same continuation therefore teaches both when to stop and how to refine, without requiring every training trajectory to run to the maximum horizon.

Key Insight

Occasionally *continuing* teaches both when to *stop* and how to refine further.

Guided by this insight, we propose **StreamRefine**, which models medical image translation as an autoregressive (AR) stream of successive 3D refinement states. Following streaming video generation (Huang et al., 2025; Yang et al., 2026), **`StreamRefine`** conditions each step on the source and generated history, but progresses through refinement states of a single 3D volume, with each step producing the next 3D refinement state. Our ***Continue-to-Check*** training uses the continued trajectories to train both later refinements and a lightweight head that estimates whether further refinement is still worthwhile. The stopping target balances translation improvement, anatomical drift, and computation cost. A simple reach correction reweights losses for later states that are observed less frequently during training. At inference, the head decides when to stop without continuation checks or ground truth (GT) targets. Our contributions are summarized as follows:

- **Conceptually**, we study learned stopping as a decision about both computation and learning. Truncation removes both feedback about later refinements and the supervision needed to learn them, and we distinguish these two effects.
- **Technically**, we introduce **`StreamRefine`** for medical image translation with *Continue-to-Check* training, which *occasionally continues* beyond *proposed stops* so that the same trajectories supervise both *continue benefit* prediction and otherwise skipped refinements.
- **Empirically**, across diverse 3D medical translation tasks, we evaluate **`StreamRefine`** under matched training budgets, demonstrating quality–efficiency gains and using mechanism analyses to explain how later-refinement learning and stopping decisions improve translation performance.

2 Related Work

Medical Image Translation. Medical image translation synthesizes a target modality from an available source while preserving patient-specific anatomy (Shin et al., 2025; Zhu et al., 2025; Wu

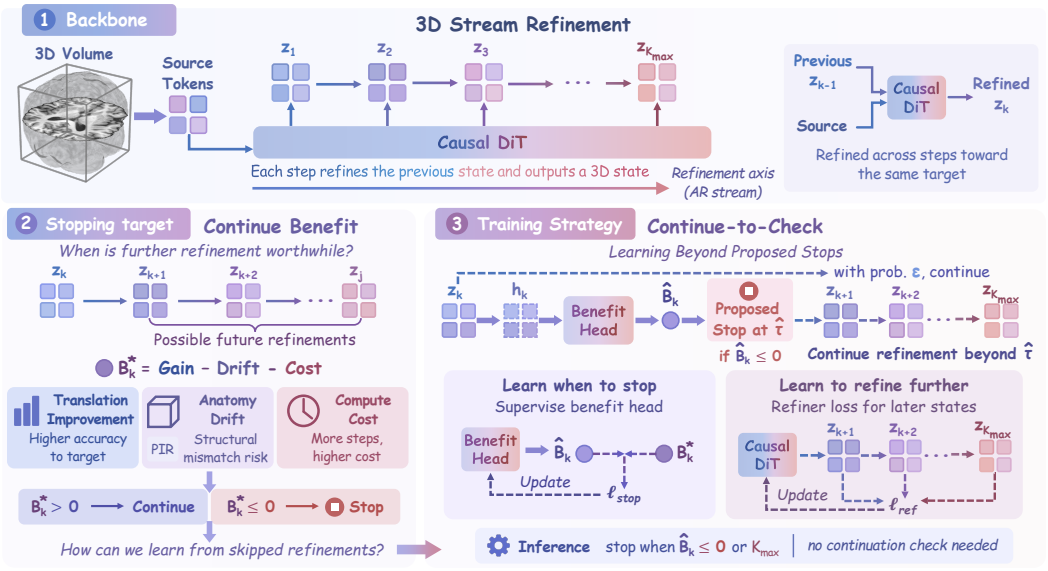


Figure 2: **Overview of `StreamRefine`.** A 3D Causal DiT produces a stream of 3D refinement states. ***Continue Benefit*** determines whether further refinement is worthwhile by balancing translation improvement, anatomical drift, and computation cost. ***Continue-to-Check*** *occasionally continues* beyond *proposed stops* during training to learn both when to *stop* and how to refine further.

et al., 2026; Chen et al., 2025; Du et al., 2026). Existing methods commonly use GANs or diffusion models for direct or iterative translation (Meng et al., 2024; Pang et al., 2026; Zhao et al., 2026b; Dorent et al., 2026), while recent AR formulations progressively refine predictions through evolving generated context (Tian et al., 2024; Ren et al., 2025; He et al., 2026). Nevertheless, these approaches typically follow predetermined generation schedules shared across cases, despite varying refinement needs across modalities and subjects (Arslan et al., 2025; Schusterbauer et al., 2026; Fabian et al., 2024). While this motivates case-dependent stopping, how such stopping affects later-refinement learning when it also truncates training trajectories remains largely underexplored.

Streaming Video Generation. Recent streaming video generators use causal or self-generated rollouts for progressive long-horizon generation (Huang et al., 2025; Yin et al., 2025; Lin et al., 2025; Yang et al., 2026; Zhu et al., 2026; Cai et al., 2026; Mao et al., 2026). **`StreamRefine`** adopts this self-generated rollout formulation, but progresses through refinement states of a single 3D volume rather than video frames, with each AR step producing the next 3D refinement state.

Learned Stopping and Iterative Generation. Learned stopping and adaptive generation allocate computation based on the input or current generation state (Graves, 2016; Chen et al., 2020; Banino et al., 2021; Qu et al., 2026; Ye et al., 2025; Le et al., 2026). Randomized truncation methods use stochastic continuation and reweighting to reduce bias from truncated computation (Tallec & Ollivier, 2017; Beatson & Adams, 2019). **`StreamRefine`** instead addresses the selective observation caused by learned stopping, using continued trajectories to train both later refinements and stopping decisions.

3 Methodology

Fig. 2 provides an overview of **`StreamRefine`** as an AR stream of 3D refinement states with occasional continuation beyond proposed stops to supervise both stopping and later refinement. Sec. 3.1 introduces the refinement stream, Sec. 3.2 defines *Continue Benefit*, Sec. 3.3 presents *Continue-to-Check*, and Sec. 3.4 summarizes training and inference.

3.1 Stream of 3D Refinement States

Given a source volume $\mathbf{x}$, we encode it into source tokens that serve as conditioning throughout the refinement stream, and the model generates

$$\mathbf{z}_k = G_\theta(\mathbf{x}, \mathbf{z}_{<k}), \qquad k = 1, \ldots, K_{\max}, \tag{1}$$

where $\mathbf{z}_k$ denotes the k-th 3D latent refinement state and $\mathbf{z}_{<k}$ denotes previous states. During training, each state corresponds to a latent crop, while whole-volume inference fuses synchronized window-level states into a full-volume state. We implement G_θ as a 3D Causal DiT with spatial modeling within each state and causal dependence across refinement steps. The first state starts from Gaussian noise (Liu et al., 2023b; Lipman et al., 2023), while later states start from the detached previous prediction and refine it conditioned on the source and history (Appendix B.1). The resulting stream is used to determine whether further refinement is worthwhile.

3.2 Continue Benefit: When Is Further Refinement Worthwhile?

Useful refinement can vary across cases, so a fixed horizon may stop too early or refine longer than needed. To determine when to stop, we quantify ***Continue Benefit*** as the remaining gain from further refinement after accounting for anatomical drift and computation cost.

Let L_k denote the translation error of state $\mathbf{z}_k$, measured as the mean absolute error to the paired target in latent space, and A_k a structural-error proxy. Since useful refinement should improve target appearance without unnecessarily altering patient anatomy, we also account for structural change. We compute A_k using Pair-Invariant Relations (PIR), a pair-calibrated latent structural proxy used only to define the stopping target (Appendix B.2).

We use normalized quantities $\bar{L}_k = L_k/s_L$ and $\bar{A}_k = A_k/s_A$, where $s_L, s_A > 0$ are fixed scales estimated from training trajectories. For a completed trajectory and $k < K_{\max}$, *Continue Benefit* is

$$B_k^* = \max_{k<j\le K_{\max}} \left[\underbrace{\bar{L}_k - \bar{L}_j}_{\text{Translation improvement}} \underbrace{-\lambda_a(\bar{A}_j - \bar{A}_k)}_{\text{Anatomical drift}} \underbrace{-\lambda_c(j-k)}_{\text{Compute cost}} \right], \tag{2}$$

where j indexes a later state and $\lambda_a, \lambda_c \ge 0$ control anatomy and computation. The maximum over future states captures delayed improvements. $B_k^* > 0$ favors *continuation*, while $B_k^* \le 0$ favors *stopping*; $B_{K_{\max}}^* = 0$ is used only as a boundary convention. We call B_k^* the *hindsight benefit*, computed from a completed trajectory for supervision, and $\widehat{B}_k$ the *predicted benefit*, estimated from the current state and used at inference.

A lightweight head b_ϕ predicts *Continue Benefit* from the current-state features $\mathbf{h}_k$, which encode the source and refinement history, as $\widehat{B}_k = b_\phi(\mathbf{h}_k)$. The model proposes *stopping* when $\widehat{B}_k \le 0$ and always stops at $K_{\max}$, without requiring future states or the paired target at inference. The challenge is then: *how can the head learn from later refinements that an early stop would otherwise hide?* We address this with *Continue-to-Check* in the next section.

3.3 Continue-to-Check: Learning Beyond Proposed Stops

Stopping early hides later outcomes and leaves later refinement states without direct supervision. We introduce ***Continue-to-Check***, which *occasionally continues* beyond *proposed stops* during training. The core idea is to use these continued trajectories to learn whether further refinement is worthwhile while also training the otherwise skipped refinements.

Let $\hat{\tau}$ denote the first proposed stopping step, with $\hat{\tau} = K_{\max}$ if no early stop is proposed. A single random draw determines the executed horizon $\tilde{\tau}$,

$$S \sim \text{Bernoulli}(\epsilon), \qquad \tilde{\tau} = \begin{cases} K_{\max}, & S = 1, \\ \hat{\tau}, & S = 0, \end{cases} \tag{3}$$

where $\epsilon \in (0, 1]$ denotes the continuation probability, meaning that an ϵ fraction of proposed stops are overridden to continue stream refinement until $K_{\max}$. Otherwise, the trajectory stops as proposed, while all preceding states remain unchanged.

Learning When to Stop. Completing the trajectory reveals otherwise unavailable benefit targets B_k^*, which supervise the benefit head through a per-state regression loss

$$\ell_k^{\text{stop}} = \text{SmoothL1}(\widehat{B}_k, B_k^*), \qquad k < K_{\max}. \tag{4}$$

where SmoothL1 measures the prediction error. By matching $\widehat{B}_k$ to B_k^*, the head learns whether further refinement is worthwhile at each state and thus when to stop. We detach $\mathbf{h}_k$ and B_k^*, so this

Table 1: **Quantitative results for brain MRI translation on BraTS 2024.**

Methods	T2w→T2f			T1c→T1n			T2f→T2w			T1n→T2w			T1n→T1c		
	PSNR↑	SSIM↑	FID↓	PSNR↑	SSIM↑	FID↓	PSNR↑	SSIM↑	FID↓	PSNR↑	SSIM↑	FID↓	PSNR↑	SSIM↑	FID↓
I^2SB (Liu et al., 2023a)	20.88	0.656	60.2	24.72	0.821	39.5	22.12	0.735	41.8	22.25	0.739	43.6	21.44	0.681	49.3
DBIM (Zheng et al., 2025)	21.08	0.664	55.4	24.97	0.827	36.8	22.31	0.742	42.6	22.46	0.746	40.7	21.76	0.688	47.9
selfRDB (Arslan et al., 2025)	21.42	0.674	49.9	25.16	0.834	34.0	22.48	0.748	38.7	22.68	0.761	38.0	22.03	0.695	44.8
DM-SAM (Pang et al., 2026)	21.35	0.671	48.6	25.23	0.832	35.2	22.94	0.771	35.8	22.83	0.766	38.1	22.08	0.697	46.5
CRSG (Song et al., 2026)	21.19	0.670	54.8	25.31	0.834	33.6	22.77	0.762	37.9	22.78	0.759	37.2	21.85	0.691	46.1
Fixed-K Refine	21.41	0.679	46.9	25.52	0.835	32.6	22.89	0.766	34.0	23.25	**0.779**	33.4	22.34	0.702	41.8
Post-hoc Stop	21.52	0.678	44.7	25.64	0.840	**31.2**	23.02	0.769	34.3	23.18	**0.779**	32.9	22.36	0.700	42.5
`StreamRefine`	**21.68**	**0.681**	**43.5**	**25.76**	**0.841**	31.7	**23.14**	**0.776**	**31.3**	**23.31**	0.778	**32.3**	**22.61**	**0.704**	**39.4**

Table 2: **Quantitative results for medical image translation on SynthRAD 2025 and AutoPET.**

Methods	SynthRAD 2025									AutoPET					
	MRI→CT			CBCT→CT			CT→MRI			CT→PET			PET→CT		
	PSNR↑	SSIM↑	FID↓	PSNR↑	SSIM↑	FID↓	PSNR↑	SSIM↑	FID↓	PSNR↑	SSIM↑	FID↓	PSNR↑	SSIM↑	FID↓
I^2SB (Liu et al., 2023a)	20.27	0.689	105.8	22.71	0.730	47.7	17.94	0.642	58.4	26.04	0.917	53.6	23.25	0.889	92.9
DBIM (Zheng et al., 2025)	20.53	0.697	99.5	22.87	0.731	45.1	18.19	0.654	55.0	25.97	0.920	54.4	23.53	0.895	88.3
selfRDB (Arslan et al., 2025)	20.81	0.708	93.6	23.19	0.740	39.2	18.48	0.670	51.3	26.08	0.923	51.1	23.67	0.901	83.7
DM-SAM (Pang et al., 2026)	21.09	0.712	91.8	23.31	0.741	39.8	18.87	0.679	**47.4**	26.72	0.938	49.6	24.16	0.908	**72.1**
CRSG (Song et al., 2026)	20.62	0.694	97.3	22.78	0.732	42.4	18.61	0.668	50.1	26.85	0.937	51.7	23.98	0.903	78.2
Fixed-K Refine	21.08	0.712	91.0	23.54	0.751	37.3	18.72	0.677	50.6	27.46	0.942	50.6	24.30	0.909	74.5
Post-hoc Stop	21.16	**0.718**	89.8	23.50	0.748	36.2	18.84	0.683	48.9	27.39	0.944	49.8	24.35	0.909	74.9
`StreamRefine`	**21.28**	0.716	**86.9**	**23.63**	**0.753**	**34.8**	**18.92**	**0.686**	47.8	**27.65**	**0.950**	**47.5**	**24.46**	**0.912**	72.7

loss updates only ϕ. Naturally completed trajectories provide supervision, while incomplete ones are skipped rather than assigned zero benefit.

Learning to Refine Further. The same continuation also provides paired supervision to otherwise skipped refinements through the step-wise refiner loss ℓ_k^{ref} (Appendix B.3). The additional losses $\ell_{\hat{\tau}+1}^{\text{ref}}, \ldots, \ell_{K_{\max}}^{\text{ref}}$ update G_θ together with the preceding steps, allowing continuation to both assess and improve later refinements.

Reach Correction. States after the proposed stop are observed only when continuation is selected. We therefore reweight reached refiner losses by their inverse reach probability, using weight 1 before the stop and $1/\epsilon$ afterwards, and average them to form $\mathcal{L}_{\text{ref}}$. Stopping losses are similarly reweighted when complete benefit targets are available to form $\mathcal{L}_{\text{stop}}$ (details in Appendix B.4).

3.4 Training and Inference

Training. We jointly train G_θ and b_ϕ on *Continue-to-Check* trajectories using $\mathcal{L} = \mathcal{L}_{\text{ref}} + \mathcal{L}_{\text{stop}}$, where $\mathcal{L}_{\text{ref}}$ and $\mathcal{L}_{\text{stop}}$ denote the reweighted refinement and stopping losses, respectively.

Inference. At inference, the model generates successive states until $\widehat{B}_k \leq 0$ or $K_{\max}$ is reached, and returns the corresponding prediction. No continuation checks or PIR computations are required.

4 Experiments

In this section, we conduct experiments to answer the following four research questions (RQs):

RQ1 Does `StreamRefine` improve translation quality and the quality–efficiency trade-off? §4.2

We compare translation performance, anatomy-related performance, and inference efficiency against representative translation methods and matched internal baselines.

RQ2 Are the key design choices of `StreamRefine` necessary? §4.3

We use a *One-step* reference and a 2×2 ablation to isolate the effects of *Continue-to-Check* and *anatomy-aware stopping* on translation quality, downstream segmentation, and refinement efficiency.

RQ3 Does occasional continuation improve later refinements? §4.4

We compare different *continuation* variants at the same refinement depths and with matched prefixes to isolate improvements in later refinements.

RQ4 Does `StreamRefine` learn when further refinement is worthwhile? §4.5

We compare *predicted* and *hindsight benefit*, analyze stopping regret on shared trajectories, and investigate how anatomy-aware stopping affects downstream performance.

Table 3: **Tumor-ROI reconstruction and downstream tumor segmentation on BraTS 2024.**

Methods	T2w→T2f			T1c→T1n			T2f→T2w			T1n→T2w			T1n→T1c		
	PSNR↑	SSIM↑	Dice↑	PSNR↑	SSIM↑	Dice↑	PSNR↑	SSIM↑	Dice↑	PSNR↑	SSIM↑	Dice↑	PSNR↑	SSIM↑	Dice↑
I^2SB (Liu et al., 2023a)	17.92	0.651	0.737	20.56	0.778	0.802	18.84	0.697	0.758	18.67	0.695	0.767	19.23	0.671	0.751
DBIM (Zheng et al., 2025)	17.81	0.645	0.740	20.88	0.789	0.816	19.10	0.709	0.772	18.93	0.702	0.762	19.51	0.681	0.769
selfRDB (Arslan et al., 2025)	18.15	0.668	0.761	21.12	0.795	0.825	19.42	0.719	0.791	19.28	0.711	0.786	19.86	0.692	0.788
DM-SAM (Pang et al., 2026)	18.28	0.663	0.754	21.46	0.803	0.833	19.71	0.727	0.803	19.49	0.719	0.797	20.02	0.699	0.794
CRSG (Song et al., 2026)	18.03	0.665	0.749	21.57	0.801	0.837	19.58	0.724	0.798	19.56	0.716	0.801	20.07	0.692	0.782
Fixed-K Refine	18.21	0.668	0.763	21.38	0.801	0.833	19.76	0.725	0.801	19.59	0.722	0.801	20.18	0.696	0.802
Post-hoc Stop	18.25	0.664	0.760	21.49	0.800	0.838	19.73	0.730	0.807	19.52	0.719	0.808	20.10	0.696	0.797
StreamRefine	**18.37**	**0.674**	**0.768**	**21.68**	**0.805**	**0.845**	**19.89**	**0.735**	**0.815**	**19.68**	**0.726**	**0.812**	**20.21**	**0.701**	**0.807**

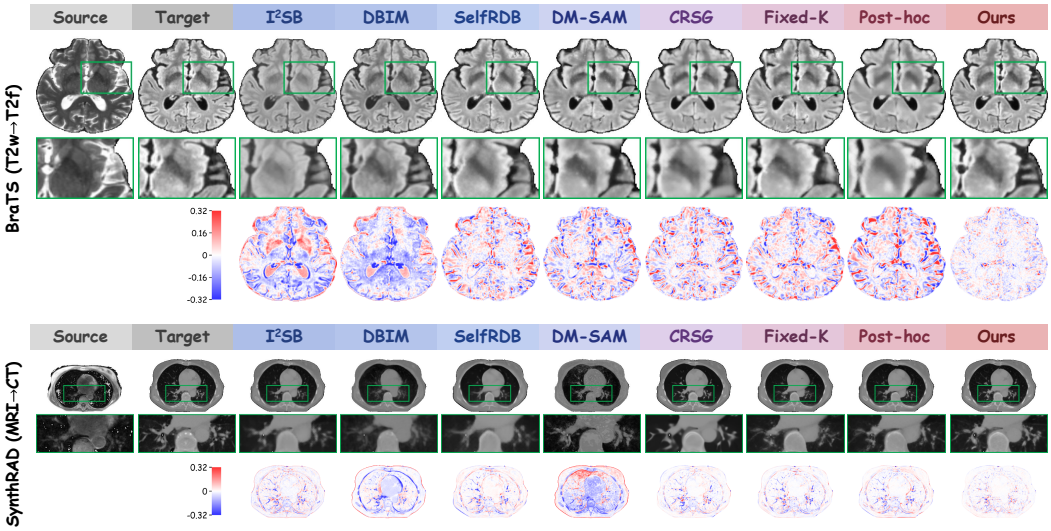


Figure 3: **Qualitative results** on BraTS T2w→T2f and SynthRAD MRI→CT cases, shown from top to bottom, with corresponding error maps below.

4.1 Experimental Setup

Datasets. We conduct experiments on three medical imaging datasets. BraTS 2024 (de Verdier et al., 2024) contains multi-modal brain MRI, including T1c, T1n, T2w, and T2f. SynthRAD 2025 (Thummerer et al., 2025) includes paired MRI and CT as well as CBCT and CT scans from head-and-neck and abdominal regions. AutoPET (Dexl et al., 2026) consists of paired PET and CT scans spanning the brain, pelvis, and abdomen. Additional information is provided in Appendix A.

Implementation Details. We build **StreamRefine** on Self-Forcing (Huang et al., 2025) and use a frozen VidTok-KL tokenizer (Tang et al., 2024) with an $8 \times 8 \times 4$ compression ratio for 3D volumes. The Causal DiT is adapted with 3D patchification and 3D RoPE, with 4 denoising steps per refinement state. We set $K_{\max} = 4$, $\lambda_a = 1.0$, $\lambda_c = 0.1$, and $\epsilon = 0.15$, and train on aligned $192 \times 192 \times 96$ crops. Full-volume inference uses sliding windows and KV caching. For sliding-window inference, stopping is determined jointly at the volume level. Further details are presented in Appendix C.

Comparison Methods. We compare **StreamRefine** with representative translation baselines and two matched internal baselines using the same backbone. Specifically, ***Fixed-K Refine*** executes the full fixed horizon $K = K_{\max} = 4$ for all cases., while ***Post-hoc Stop*** learns stopping only after a full-horizon refiner has been trained and frozen. This comparison distinguishes gains from adaptive stopping alone from those of jointly learning when to stop and how to refine further.

4.2 Main Results (RQ1)

Translation Performance. Across Tables 1 and 2, **StreamRefine** performs strongly across most tasks and metrics. Its modest but consistent gains over the two internal baselines support jointly learning when to stop and how to refine further. Fig. 3 compares representative cases with zoomed regions and corresponding error maps. **StreamRefine** more closely matches the target appearance with reduced errors across different translation tasks. Statistical significance and multi-seed results are provided in Appendix D.

Table 4: **Ablation study of the main design choices.** *CtC* denotes *Continue-to-Check*, *Anat. Stop* denotes *anatomy-aware stopping*, and Avg. Steps denotes the average refinement steps at inference.

Methods	CtC	Anat. Stop	BraTS T2w→T2f				AutoPET CT→PET			
			PSNR↑	FID↓	Dice↑	Avg. Steps↓	PSNR↑	FID↓	Dice↑	Avg. Steps↓
One-step ($K = 1$)	✗	✗	21.03	58.9	0.738	**1.00**	26.95	57.5	0.542	**1.00**
Naive Stop, image-only	✗	✗	21.37	47.7	0.751	1.79	27.20	53.2	0.556	1.95
Naive Stop	✗	✓	21.40	48.6	0.759	1.68	27.36	51.8	0.573	1.97
CtC, image-only	✓	✗	21.51	46.9	0.753	2.30	27.48	50.1	0.571	2.89
StreamRefine	✓	✓	**21.68**	**43.5**	**0.768**	2.13	**27.65**	**47.5**	**0.580**	2.57

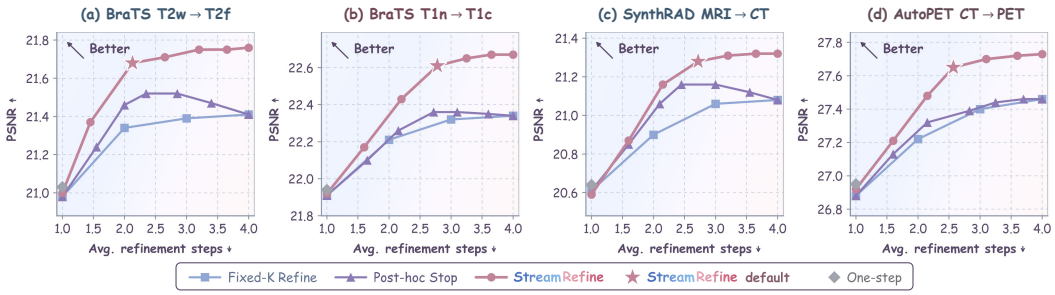


Figure 4: **Quality–efficiency trade-offs.** PSNR is plotted against the average number of refinement steps. Stars mark the default **StreamRefine** settings, and diamonds denote *One-step* references.

Anatomy-Related Performance. We assess anatomy-related performance using ROI reconstruction and downstream segmentation. For downstream evaluation, a multimodal UNet (Ronneberger et al., 2015; Isensee et al., 2021) uses the three available real modalities and the synthesized target modality, with Dice reported as the segmentation metric. **StreamRefine** consistently improves both measures across all tasks in Table 3. Please refer to Appendix D.3 for more results.

Quality–Efficiency Trade-off. Under matched total training budgets, we vary the inference horizon of *Fixed-K Refine* from one to four steps and sweep the stopping thresholds of *Post-hoc Stop* and **StreamRefine** without retraining. Across the four tasks in Fig. 4, **StreamRefine** generally achieves higher PSNR at similar average step counts. Its default settings capture most of the PSNR gains, with only small improvements from further refinement.

4.3 Ablation Studies (RQ2)

We evaluate *Continue-to-Check* and *anatomy-aware stopping* with a 2×2 ablation on BraTS T2w→T2f and AutoPET CT→PET, using *One-step* ($K = 1$) as a reference. Results in Table 4 show that *CtC* consistently improves translation quality over naive stopping across both datasets, while anatomy-aware stopping brings clearer gains in downstream Dice. Combining both gives the best overall performance while using fewer refinement steps than *CtC* with *image-only stopping*.

4.4 Learning Beyond Proposed Stops (RQ3)

We study later-refinement learning on BraTS T2w→T2f, asking whether *continuation* improves later refinements beyond *stopping* decisions. All variants share the same warm start, training setup, and number of optimization updates. ***Naive*** uses no continuation, ***Head-only*** uses it only for head supervision, and ***Refiner-only*** uses it only for later-refiner training. ***Joint*** (**StreamRefine**) uses both, with ***Full-horizon*** training as a reference.

Do later refinements improve? We disable stopping and evaluate all variants at the same refinement depths on the same cases (Fig. 5(a)). At $k = 4$, PSNR gains over *Naive* are 0.07, 0.28, and 0.35 for *Head-only*, *Refiner-only*, and *Joint*, respectively, compared with 0.39 for *full-horizon* training. *Joint* therefore recovers about 90% of the *full-horizon* improvement. The *Head-only* and *Refiner-only* gains are also numerically close to additive, suggesting that improved stopping feedback and direct later-refiner supervision provide largely complementary benefits.

Do gains come only from better prefixes? We treat the first two refinement states as the *prefix*, i.e., the preceding history for later refinement, and independently swap the prefix and suffix refiner

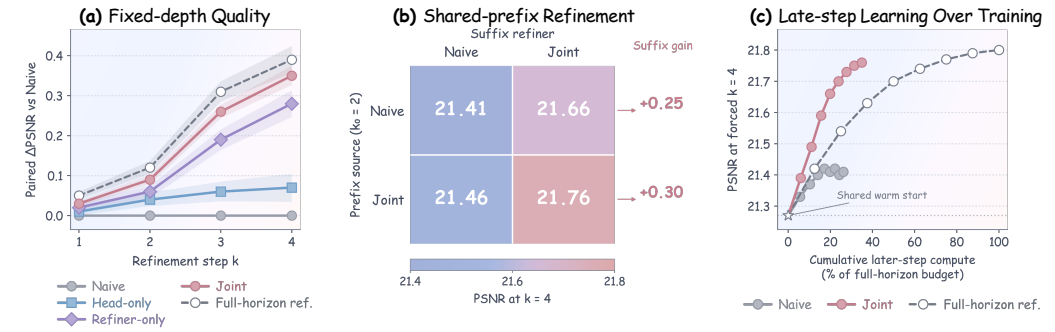


Figure 5: **Why does *continuation* help later refinement?** **(a)** At the same refinement depths, continuation improves later-step quality, especially with direct refiner supervision. **(b)** With the first two states fixed as the prefix, *Joint* still produces stronger later refinements. **(c)** *Joint* approaches *full-horizon* quality with substantially less later-step training compute.



Figure 6: **How does `StreamRefine` learn when to *stop*?** **(a)** *Predicted benefit* agrees with *hindsight benefit* across cases and refinement steps. **(b)** On the same completed held-out trajectories, the learned head has lower stopping regret than fixed, random, and image-only rules, and approaches a full-label head trained with complete trajectory supervision. **(c)** On cases where the two heads choose different stops, *anatomy-aware stopping* improves downstream Dice and also outperforms always stopping at $K = 2$.

(Fig. 5(b)). Changing the suffix refiner contributes about 0.25–0.30 in PSNR, while changing the prefix contributes only 0.05–0.10. This suggests that most of the gain comes from stronger later refinements rather than better preceding states.

Can we approach full-horizon quality efficiently? We next measure progress against cumulative later-step training compute. *Joint* reaches a PSNR of 21.76, within 0.04 of the 21.80 *full-horizon* reference, while using only 35% of its later-step training compute (Fig. 5(c)). *Naive* instead plateaus at 21.41 as later states receive limited training, whereas *Joint* continues improving beyond the shared warm start. This indicates that *occasional continuation* captures most of the *full-horizon* later-refinement gain without training every trajectory to the maximum depth.

Takeaway

Occasional continuation helps the model learn better later refinements, while using much less later-step training compute than full-horizon training.

4.5 Learning When to Stop (RQ4)

We examine whether the learned head chooses useful *stopping* points. For quantitative analysis on BraTS T2w→T2f, we freeze the refiner, complete held-out trajectories to $K_{\max} = 4$ offline, and apply all stopping rules to the same volume-level states.

Can the head tell when to continue? We compare predicted and hindsight *Continue Benefit* at $k = 1, 2, 3$ on the same completed trajectories (Fig. 6(a)). They show strong overall agreement ($\rho = 0.811$), which remains within each step ($\rho = 0.869/0.752/0.528$ for $k = 1/2/3$). Their signs also agree in 86.0% of cases, indicating consistent *continue*-or-*stop* decisions.

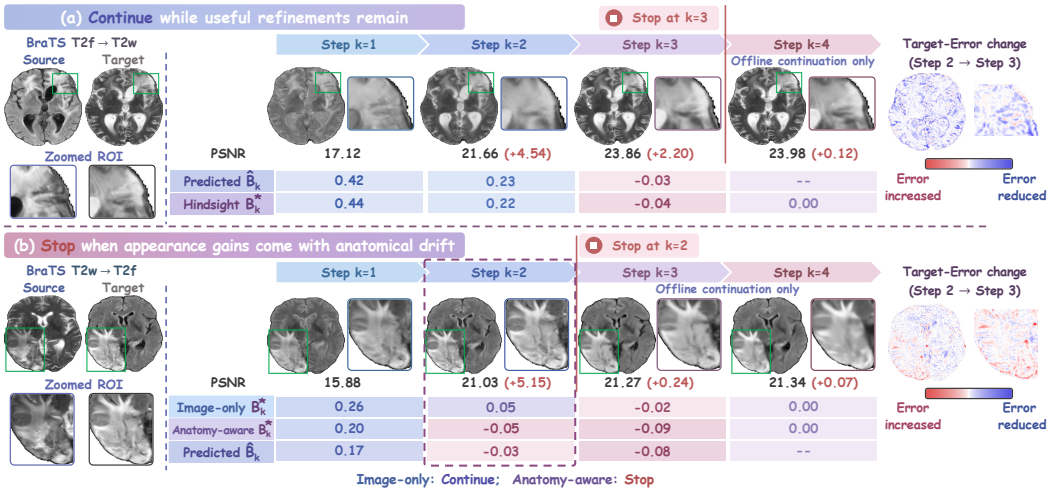


Figure 7: **Visualization of *stopping* decisions. In (a),** `StreamRefine` stops at $k = 3$ when both *predicted* and *hindsight benefits* become non-positive. **In (b),** Step 3 slightly improves PSNR, but the *anatomy-aware benefit* is already negative at $k = 2$, leading to an earlier stop.

Does learning improve stopping decisions? We measure stopping regret, the extra cost relative to the best stop found from the completed trajectory. The learned head achieves much lower regret than *matched-random stopping* (0.023 vs. 0.105) at the same average of 2.13 steps, and also improves over *fixed* $K = 2$ (0.079) (Fig. 6(b)). It closes about 78% of the random-to-oracle regret gap and approaches the full-label head (0.013), which uses complete trajectory labels during training. Consistently, stops that still leave positive *hindsight benefit* decrease from 43.6% to 23.9% for *image-only stopping* and from 36.4% to 18.9% for *anatomy-aware stopping* (Appendix E.4).

Why include anatomy in stopping? The *anatomy-aware* and *image-only* heads choose different stops for 56 of 254 cases (Fig. 6(c)). On these cases, *anatomy-aware stopping* improves Dice from 0.757 to 0.771 with fewer steps, and also exceeds *fixed* $K = 2$ (0.761) at similar PSNR. Across all cases, changing the stop alone gives about $+0.003$ Dice, smaller than the $+0.015$ overall gain in Table 4, indicating an additional training-time effect.

Qualitative examples in Fig. 7 illustrate how `StreamRefine` adapts its *stopping* decisions to the remaining refinement value. Case (a) *continues* while later corrections remain useful, whereas case (b) *stops* earlier when a small PSNR gain is accompanied by anatomical drift.

Overall, Secs. 4.4 and 4.5 help explain the gain in Table 1. At forced $k = 4$, the *CtC*-trained refiner reaches 21.76 PSNR, compared with 21.41 for the compute-matched *Fixed-K Refine*. Adaptive stopping then removes 47% of refinement steps with only a 0.08 decrease, retaining a 0.27 gain at 2.13 steps. In contrast, *Post-hoc Stop* only selects among states of a fixed refiner.

Takeaway

Learned *stopping* makes better use of refinement steps, while *anatomy-aware* decisions help avoid unnecessary anatomical changes.

5 Conclusion

We propose **StreamRefine**, a framework for learning when to *stop* iterative 3D medical image translation without sacrificing the learning of later refinements. Its *Continue-to-Check* training *occasionally continues* beyond *proposed stops*, allowing the same trajectories to improve later refinement and supervise when further refinement remains worthwhile. Across diverse translation tasks, `StreamRefine` improves the quality–efficiency trade-off, while our analyses show that continuation strengthens later refinements and supports more effective stopping decisions. We hope this work helps illuminate how adaptive stopping can be learned together with iterative refinement in medical image translation.

AI USE STATEMENT

Generative AI tools were used to aid and polish manuscript writing, support literature retrieval and discovery, including identifying relevant related work, and provide assistance during research ideation and experimental development. All AI-assisted suggestions and outputs were independently reviewed and verified by the authors. The research ideas, methodological design, experimental results, and scientific claims were developed and validated by the authors, who take full responsibility for the final content of the paper.

REFERENCES

Fuat Arslan, Bilal Kabas, Onat Dalmaz, Muzaffer Ozbey, and Tolga Çukur. Self-consistent recursive diffusion bridge for medical image translation. *Medical Image Analysis*, 106:103747, 2025.

Andrea Banino, Jan Balaguer, and Charles Blundell. Pondernet: Learning to ponder. *arXiv preprint arXiv:2107.05407*, 2021.

Alex Beatson and Ryan P Adams. Efficient optimization of loops and limits with randomized telescoping sums. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 534–543, 2019.

Evan Bell, Jiaming Liu, Yifan Chen, and Yu Sun. Generative translation priors: Bayesian imaging with cross-modality image translation. *arXiv preprint arXiv:2608.28872*, 2026.

Shengqu Cai, Ceyuan Yang, Lvmin Zhang, Yuwei Guo, Junfei Xiao, Ziyan Yang, Yinghao Xu, Zhenheng Yang, Alan Yuille, Leonidas Guibas, Maneesh Agrawala, Lu Jiang, and Gordon Wetzstein. Mixture of contexts for long video generation. In *International Conference on Learning Representations*, volume 2026, pp. 92075–92093, 2026.

Junxin Chen, Zhiheng Ye, Renlong Zhang, Hao Li, Bo Fang, Li bo Zhang, and Wei Wang. Medical image translation with deep learning: Advances, datasets and perspectives. *Medical Image Analysis*, 103:103605, 2025.

Xinshi Chen, Hanjun Dai, Yu Li, Xin Gao, and Le Song. Learning to stop while learning to predict. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 1520–1530, 2020.

Maria Correia de Verdier, Rachit Saluja, Louis Gagnon, Dominic LaBella, Ujjwall Baid, Nourel Hoda Tahon, Martha Foltyn-Dumitru, Jikai Zhang, Maram Alafif, Saif Baig, et al. The 2024 brain tumor segmentation (brats) challenge: Glioma segmentation on post-treatment mri. *arXiv preprint arXiv:2405.18368*, 2024.

Jakob Dexl, Katharina Jeblick, Andreas Mittermeier, Balthasar Schachtner, Anna Theresa Stüber, Johanna Topalis, Maximilian Rokuss, Fabian Isensee, Klaus H Maier-Hein, Hamza Kalisch, et al. The autopet3 challenge–automated lesion segmentation in whole-body pet/ct-multitracer multicenter generalization. *arXiv preprint arXiv:2605.05775*, 2026.

Reuben Dorent, Nazim Haouchine, Alexandra Golby, Sarah Frisken, Tina Kapur, and William Wells. Unified cross-modal medical image synthesis with hierarchical mixture of product-of-experts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(2):1641–1656, 2026.

Chenhe Du, Qing Wu, Xuanyu Tian, Jingyi Yu, Hongjiang Wei, and Yuyao Zhang. Improving 2d diffusion models for 3d medical imaging with inter-slice consistent stochasticity. In *The Fourteenth International Conference on Learning Representations*, 2026.

Zalan Fabian, Berk Tinaz, and Mahdi Soltanolkotabi. Adapt and diffuse: Sample-adaptive reconstruction via latent diffusion models. In *Forty-first International Conference on Machine Learning*, 2024.

Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.

Zhicheng He, Yunpeng Zhao, Junde Wu, Ziwei Niu, Zijun Li, Bohan Li, Lanfen Lin, and Yueming Jin. Medvar: Towards scalable and efficient medical image generation via next-scale autoregressive prediction. *arXiv preprint arXiv:2602.14512*, 2026.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 167283–167308, 2025.

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.

Thanh-Tung Le, Yunhan Zhao, Menglei Chai, Zhengyang Shen, Zhe Cao, Danhang Tang, Xiaohui Xie, and Deying Kong. Dsa: Dynamic step allocation for fast autoregressive video generation. *arXiv preprint arXiv:2606.04432*, 2026.

Zizheng Li, RenDong Xie, Huadeng Wang, Zhifen He, Bin Liu, Bo Li, and Xiaonan Luo. Proto-gaussian: Mri modality translation based on learnable structural prototypes and 2d gaussian splatting. In *European Conference on Computer Vision*, pp. 77–95, 2026.

Shanchuan Lin, Ceyuan Yang, Hao He, Jianwen Jiang, Yuxi Ren, Xin Xia, Yang Zhao, Xuefeng Xiao, and Lu Jiang. Autoregressive adversarial post-training for real-time interactive video generation. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 41061–41086, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos Theodorou, Weili Nie, and Anima Anandkumar. I^2SB: Image-to-image schrödinger bridge. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 22042–22062, 2023a.

Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023b.

Xiaofeng Mao, Shaohao Rui, Kaining Ying, Bo Zheng, Chuanhao Li, Mingmin Chi, and Kaipeng Zhang. Packforcing: Short video training suffices for long video sampling and long context inference. *arXiv preprint arXiv:2603.25730*, 2026.

Xiangxi Meng, Kaicong Sun, Jun Xu, Xuming He, and Dinggang Shen. Multi-modal modality-masked diffusion network for brain mri synthesis with random modality missing. *IEEE Transactions on Medical Imaging*, 43(7):2587–2598, 2024.

Haowen Pang, Xiaoming Hong, Peng Zhang, Pengli Zhu, Shun Yao, Fengping An, Guoyuan Yang, Tiantian Liu, Anqi Qiu, Chuyang Ye, and Tianyi Yan. Cascaded diffusion model and segment anything model for medical image synthesis. *Pattern Recognition*, 180:114148, 2026.

Xiangyan Qu, Zhenlong Yuan, Jing Tang, Rui Chen, Datao Tang, Meng Yu, Lei Sun, Yancheng Bai, Xiangxiang Chu, Gaopeng Gou, Gang Xiong, and Yujun Cai. From scale to speed: Adaptive test-time scaling for image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23272–23282, 2026.

Sebastian Rassmann, David Kügler, Christian Ewert, and Martin Reuter. Regression is all you need for medical image translation. *IEEE Transactions on Medical Imaging*, 45(5):2156–2172, 2026.

Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Beyond next-token: Next-x prediction for autoregressive visual generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15781–15791, 2025.

Giulia Romoli, Alessia Capoccia, Filippo Ruffini, Francesco Di Feola, Luca Boldrini, Arturo Chiti, Renato Cuocolo, Tugba Akinci D'Antonoli, Fatemeh Darvizeh, Marcello Di Pumpo, et al. Cross modality image translation in medical imaging using generative frameworks. *arXiv preprint arXiv:2605.13686*, 2026.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, 2015.

Johannes Schusterbauer, Ming Gui, Yusong Li, Pingchuan Ma, Felix Krause, and Björn Ommer. Denoising, fast and slow: Difficulty-aware adaptive sampling for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 43260–43270, 2026.

Yejee Shin, Yeeun Lee, Hanbyol Jang, Geonhui Son, Hyeongyu Kim, and Dosik Hwang. Anatomical consistency and adaptive prior-informed transformation for multi-contrast mr image synthesis via diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 30918–30927, 2025.

Zhiyun Song, Xin Wang, Honglin Xiong, Mengjun Liu, Manman Fei, Xiangyu Zhao, Zhenrong Shen, Sheng Wang, Dongdong Chen, Linlin Yao, Qian Wang, and Lichi Zhang. Enhancing generative models for modality imputation of 3-d mris via consistency-aware refinement and super-resolution guidance. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2026.

Corentin Tallec and Yann Ollivier. Unbiasing truncated backpropagation through time. *arXiv preprint arXiv:1705.08209*, 2017.

Anni Tang, Tianyu He, Junliang Guo, Xinle Cheng, Li Song, and Jiang Bian. Vidtok: A versatile and open-source video tokenizer. *arXiv preprint arXiv:2412.13061*, 2024.

Adrian Thummerer, Erik van der Bijl, Arthur Jr Galapon, Florian Kamp, Mark Savenije, Christina Muijs, Shafak Aluwini, Roel J. H. M. Steenbakkers, Stephanie Beuel, Martijn PW Intven, Johannes A. Langendijk, Stefan Both, Stefanie Corradini, Viktor Rogowski, Maarten Terpstra, Niklas Wahl, Christopher Kurz, Guillaume Landry, and Matteo Maspero. Synthrad2025 grand challenge dataset: Generating synthetic cts for radiotherapy from head to abdomen. *Medical Physics*, 52(7): e17981, 2025.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *Advances in Neural Information Processing Systems*, volume 37, pp. 84839–84865, 2024.

Jinbao Wei, Yuhang Chen, Zhijie Wang, Gang Yang, Shimin Tao, Jian Gao, Aiping Liu, and Xun Chen. Rethinking diffusion bridge model with dual alignments for medical image synthesis. In *Proceedings of the 33rd ACM International Conference on Multimedia*, MM '25, pp. 1052–1061, 2025.

Yicheng Wu, Tao Song, Zhonghua Wu, Jin Ye, Zongyuan Ge, Wenjia Bai, Zhaolin Chen, and Jianfei Cai. Virtual full-stack scanning of brain mri via imputing any quantised code. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21026–21035, 2026.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Ying-Cong Chen, Yao Lu, Song Han, and Yukang Chen. Longlive: Real-time interactive long video generation. In *The Fourteenth International Conference on Learning Representations*, 2026.

Zilyu Ye, Zhiyang Chen, Tiancheng Li, Zemin Huang, Weijian Luo, and Guo-Jun Qi. Schedule on the fly: Diffusion time prediction for faster and better image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23412–23422, 2025.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22963–22974, 2025.

Chenlu Zhan, Yu Lin, Gaoang Wang, Hongwei Wang, and Jian Wu. Medm2g: Unifying medical multi-modal generation via cross-guided diffusion with visual invariant. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11502–11512, 2024.

Jinze Zhao, Anqi Xiao, Jie Tian, Shimao Zhu, and Zhenhua Hu. Spid-dm: Structure-guided diffusion model with proportional-integral-derivative controller for unpaired medical image translation. *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2026a.

Weiren Zhao, Yi Dong, and Cheng Chen. Synermedgen: Synergizing medical multimodal understanding with generation via task alignment. *arXiv preprint arXiv:2605.08724*, 2026b.

Kaiwen Zheng, Guande He, Jianfei Chen, Fan Bao, and Jun Zhu. Diffusion bridge implicit models. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 81857–81884, 2025.

Langtao Zhou, Xiaoxia Qu, Tianyu Fu, Jiaoyang Wu, Hong Song, Jingfan Fan, Danni Ai, Deqiang Xiao, Junfang Xian, and Jian Yang. Anatomy-aware sketch-guided latent diffusion model for orbital tumor multi-parametric mri missing modalities synthesis. *IEEE Transactions on Medical Imaging*, 45(5):2140–2155, 2026.

Hongzhou Zhu, Min Zhao, Guande He, Hang Su, Chongxuan Li, and Jun Zhu. Causal forcing: Autoregressive diffusion distillation done right for high-quality real-time interactive video generation. In *Forty-third International Conference on Machine Learning*, 2026.

Xiyue Zhu, Dou Hoon Kwark, Ruike Zhu, Kaiwen Hong, Yiqi Tao, Shirui Luo, Yudu Li, Zhi-Pei Liang, and Volodymyr Kindratenko. Introducing 3d representation for dense volume-to-volume translation via score fusion. In *Forty-second International Conference on Machine Learning*, 2025.

Appendix Contents

A Datasets **15**

B Additional Method Details **15**

B.1 3D Stream Refinement Architecture 15

B.2 Pair-Invariant Relations as a Structural Proxy 16

B.3 Refiner Training Objective 17

B.4 Reach Correction 17

B.5 Benefit Scale Calibration 18

C More Implementation Details **18**

C.1 Volume Latent Generation 18

C.2 Training and Inference Details 18

C.3 Tokenizer Reconstruction Performance 19

C.4 Interpretation of Volume-Level Stopping 20

D Further Experimental Results **20**

D.1 Statistical Significance of Main Results 20

D.2 Multi-seed Robustness Analysis 21

D.3 Tumor- and Lesion-ROI Evaluation 21

D.4 Additional Qualitative Results 21

D.5 Additional Ablation Studies 22

E Additional Analysis **25**

E.1 Stopping and Refinement Analysis 25

E.2 PIR Agreement with Independent Structural Measures 27

E.3 Later-Refinement Analysis Protocol 27

E.4 Stopping Decisions and Hidden Continuation Benefit 28

E.5 Qualitative Analysis of Later-Refinement Learning 29

E.6 Computational Efficiency Analysis 29

F Discussion and Limitations **30**

APPENDIX

A DATASETS

BraTS 2024. Our BraTS 2024 (de Verdier et al., 2024) experiments combine three cohorts, namely post-treatment adult glioma (GLI), brain metastases (MET), and pediatric brain tumors (PED). For each subject, four co-registered MRI modalities are available, including T1n, T1c, T2w, and T2f, along with tumor annotations. These annotations are excluded from translation training and used only for ROI-based and downstream evaluation. We randomly divide the subjects into training and validation sets using a 9:1 ratio.

SynthRAD 2025. We use the head-and-neck and abdominal subsets of the multi-center SynthRAD 2025 dataset (Thummerer et al., 2025). The dataset contains paired MRI–CT and CBCT–CT scans prepared for radiotherapy applications. Following the released preprocessing, the paired scans are spatially aligned and form corresponding 3D source and target volumes for our translation tasks. We adopt the official training and validation partitions.

AutoPET. AutoPET (Dexl et al., 2026) contains paired whole-body PET and CT scans together with tumor-lesion annotations. We construct CT→PET and PET→CT translation tasks from the paired volumes, which span anatomical regions such as the brain, pelvis, and abdomen. The lesion annotations are not used during translation training and are retained only for ROI-based and downstream evaluation. We use the official training and validation splits.

Dataset splits and subject counts for all translation tasks are provided in Table 5. A subject-disjoint subset of the training data is held out for model selection and hyperparameter tuning, whereas the validation subjects reported in the table are reserved exclusively for final evaluation. All model and hyperparameter settings are finalized before evaluation on the validation set.

Table 5: **Summary of datasets and translation tasks.** We report the number of subjects used for training and validation for each task.

Dataset	Translation Direction	Training	Validation
BraTS 2024 (de Verdier et al., 2024)	T2w→T2f	2,280	254
	T1c→T1n	2,280	254
	T2f→T2w	2,280	254
	T1n→T2w	2,280	254
	T1n→T1c	2,280	254
SynthRAD 2025 (Thummerer et al., 2025)	MRI→CT	407	106
	CT→MRI	407	106
	CBCT→CT	622	152
AutoPET (Dexl et al., 2026)	CT→PET	1,279	323
	PET→CT	1,279	323

B ADDITIONAL METHOD DETAILS

B.1 3D STREAM REFINEMENT ARCHITECTURE

Our backbone follows the self-generated AR rollout of Self-Forcing (Huang et al., 2025), adapted from video generation to volumetric refinement. Instead of generating video frames, **`StreamRefine`** autoregressively generates a sequence of complete 3D latent states. During training and inference, each refinement state is conditioned on the source volume and the model's own previously generated states, matching the self-rollout setting used at inference.

We use a frozen VidTok-KL tokenizer (Tang et al., 2024) to represent each medical volume as a continuous 3D latent tensor. The tokenizer's temporal dimension is interpreted as anatomical depth only for volumetric encoding; autoregression is performed along the refinement axis rather than along anatomical slices. For our main setting, a $192 \times 192 \times 96$ image crop is compressed to a $24 \times 24 \times 24$ latent crop with 16 channels.

The refiner G_θ is implemented as a shared 3D Causal DiT. Each latent state is divided into $2 \times 2 \times 2$ 3D patches, and three-dimensional rotary positional embeddings encode their locations along the depth, height, and width axes. Spatial tokens within the current 3D state are modeled jointly, while causal dependence is maintained across refinement states so that state k only uses the source and history $z_{<k}$. The same model parameters are shared across all refinement steps.

For each refinement step, we define the flow starting point as

$$a_k = \begin{cases} \eta, & k = 1, \\ \mathrm{sg}(z_{k-1}), & k \geq 2, \end{cases} \tag{5}$$

where $\eta \sim \mathcal{N}(0, I)$ and $\mathrm{sg}(\cdot)$ denotes stop-gradient. The first state is therefore generated from noise, while each subsequent state starts from the previous prediction and directly refines it toward the target rather than regenerating it from scratch. Training and inference use the same starting point a_k, with inference following the same four-step Euler solver for rectified flow (Liu et al., 2023b; Lipman et al., 2023).

For state $\mathbf{z}_k$, the feature $\mathbf{h}_k$ is obtained from the final normalized hidden tokens of the current refinement state after the 3D Causal DiT. These tokens already incorporate the previous refinement states through causal self-attention and the source volume through source cross-attention. We apply validity-weighted mean pooling over the current-state tokens, where each token is weighted by the valid fraction of its 3D patch. The pooled feature is detached and passed through a lightweight LayerNorm–Linear head to predict $\widehat{B}_k$, so the stopping loss updates only the head parameters.

B.2 Pair-Invariant Relations as a Structural Proxy

PIR measures structural change using local relations that are shared by a paired source and target. Since the two volumes are spatially aligned, anatomical structure should remain similar across modalities even when image appearance changes. PIR therefore compares relations between nearby latent features rather than their absolute values.

Let $\mathbf{z}_s$ and $\mathbf{z}_t$ denote the frozen tokenizer latents of the paired source and target, and let $\mathbf{z}_k$ denote a predicted refinement state. For a latent position i and spatial offset δ, we define the local relation

$$r_{i,\delta}(z) = \cos(\mathbf{z}_i, \mathbf{z}_{i+\delta}), \tag{6}$$

where the cosine similarity measures how similar the feature vectors are at two nearby locations. We use a small fixed set of offsets $\Delta = \{(2,0,0), (0,1,0), (0,0,1)\}$ in (D, H, W) order, where the two-step depth offset compensates for the smaller compression ratio along depth.

Not all local relations are equally preserved across modalities. We therefore assign each relation a reliability weight based on its agreement between the paired source and target,

$$w_{i,\delta} = \left(1 - \frac{1}{2} \left| r_{i,\delta}(\mathbf{z}_s) - r_{i,\delta}(\mathbf{z}_t) \right| \right)^2. \tag{7}$$

Relations that are already similar between the source and target receive larger weights, while modality-dependent relations are downweighted.

The PIR structural error of refinement state $\mathbf{z}_k$ is then

$$A_k = \frac{1}{|\mathcal{E}|} \sum_{(i,\delta) \in \mathcal{E}} w_{i,\delta} \left| r_{i,\delta}(z_k) - r_{i,\delta}(\mathbf{z}_t) \right|, \tag{8}$$

where $\mathcal{E}$ contains valid neighboring pairs within the anatomical region. Because PIR emphasizes local relations that remain consistent across the paired modalities, a larger A_k indicates that the prediction has moved further away from these shared structural relations, providing a proxy for anatomical drift. By construction, $A(\mathbf{z}_t) = 0$.

PIR is computed without gradient and is used only to construct the stopping target in Sec. 3.2; it does not directly supervise the refiner.

B.3 Refiner Training Objective

Given the step-specific starting point a_k defined in Appendix B.1, the target latent y, and flow time $t \in [0, 1]$, we construct

$$y_t^{(k)} = (1-t)a_k + ty, \qquad v_k^* = y - a_k. \tag{9}$$

Conditioned on the source and previous refinement states, G_θ predicts $\hat{v}_k$ and is optimized with

$$\ell_k^{\text{flow}} = \|\hat{v}_k - v_k^*\|_{\mathcal{M}}^2, \tag{10}$$

where $\mathcal{M}$ denotes the valid latent region. We additionally form the corresponding target estimate

$$\hat{y}_k = y_t^{(k)} + (1-t)\hat{v}_k \tag{11}$$

and encourage it to match the paired target with

$$\ell_k^{\text{lat}} = \|\hat{y}_k - y\|_{\mathcal{M},1}. \tag{12}$$

The step-wise refiner objective used throughout the paper is

$$\ell_k^{\text{ref}} = \ell_k^{\text{flow}} + \ell_k^{\text{lat}}. \tag{13}$$

This loss trains the model to improve each refinement state. In contrast, PIR is used only to determine whether further refinement is worthwhile and is not included in ℓ_k^{ref}.

B.4 Reach Correction

The reach correction follows directly from the continuation rule. Conditioning on the current model and all rollout randomness except the continuation draw S, let $\mathcal{T}$ denote the corresponding potential *full-horizon* trajectory and $\hat{\tau}$ its first proposed stop. Let $R_k \in \{0, 1\}$ indicate whether refinement state k is executed. States up to $\hat{\tau}$ are always reached, while later states are reached only when continuation is selected. Therefore,

$$p_k = \Pr_S(R_k = 1 \mid \mathcal{T}) = \begin{cases} 1, & k \le \hat{\tau}, \\ \epsilon, & k > \hat{\tau}. \end{cases} \tag{14}$$

Weighting an observed loss by its inverse reach probability gives

$$\mathbb{E}_S\left[\frac{R_k}{p_k}\ell_k^{\text{ref}} \,\middle|\, \mathcal{T}\right] = \ell_k^{\text{ref}}. \tag{15}$$

Thus, later states receive weight $1/\epsilon$ when executed, while earlier states keep unit weight. This recovers each state's expected loss contribution for the current rollout under the same stop-gradient convention as *full-horizon* training. The proposed stop, reach probabilities, and sampling weights are treated as constants during backpropagation.

Benefit-head supervision is handled similarly, but its target B_k^* requires a complete trajectory. A trajectory that naturally reaches $K_{\max}$ receives unit weight, while an early-stopping trajectory completed through *Continue-to-Check* receives weight $1/\epsilon$. If an early stop is not overridden, the complete benefit target is unavailable and no head loss is applied. Accordingly,

$$w_{\text{stop}} = \begin{cases} 1, & \hat{\tau} = K_{\max}, \\ 1/\epsilon, & \hat{\tau} < K_{\max} \text{ and continuation is selected}, \\ 0, & \text{otherwise}. \end{cases} \tag{16}$$

The resulting head loss is

$$\mathcal{L}_{\text{stop}} = \frac{w_{\text{stop}}}{K_{\max} - 1} \sum_{k=1}^{K_{\max}-1} \ell_k^{\text{stop}}. \tag{17}$$

The refiner loss is

$$\mathcal{L}_{\text{ref}} = \frac{1}{K_{\max}} \sum_{k=1}^{K_{\max}} \frac{R_k}{p_k}\ell_k^{\text{ref}}. \tag{18}$$

This averages the reweighted losses over the fixed refinement horizon. For the default $\epsilon = 0.15$, continued states receive weight $1/\epsilon \approx 6.67$. This inverse weighting increases sampling variance for sparsely reached later states, while preserving their conditional expected contribution. In practice, we use the fixed $K_{\max}$ normalization above together with global gradient clipping at 1.0 as described in Appendix C.2.

Table 6: **Preprocessing settings across datasets and imaging modalities.** P_q refers to the qth percentile calculated separately for each volume and modality, while P_q^+ denotes the corresponding percentile over positive-valued voxels.

Dataset	**BraTS 2024**				**SynthRAD 2025**			**AutoPET**	
Modality	T1n	T1c	T2w	T2f	CT	CBCT	MRI	CT	PET
Spatial orientation	RAS				RAS			RAS	
Spatial spacing	Determined by volume resizing				Original spacing			Original spacing	
Foreground criterion	$x > 0$				$x > -950$	$x > -950$	$x > 0$	$x > -950$	$x > 0$
Clipping range	$[P_{0.5}, P_{99.5}]$				$[-1000, 1000]$	$[-1000, 1000]$	$[P_{0.5}^+, P_{99.5}^+]$	$[-1000, 1000]$	$[P_{0.5}^+, P_{99.5}^+]$
Intensity scaling	$[-1, 1]$				$[-1, 1]$			$[-1, 1]$	
Spatial resizing	$256 \times 256 \times 128$				No fixed resizing; variable size			No fixed resizing; variable size	
Data augmentation	Random rotations and flips				Random rotations and flips			Random rotations and flips	

B.5 Benefit Scale Calibration

We estimate s_L and s_A from *fixed-horizon* training trajectories. For each metric, we collect its values across cases and refinement states and use the 75th percentile of its absolute deviation from the median as the scale, with a minimum value of 10^{-6}. The resulting scales are computed once and kept fixed when constructing *Continue Benefit*.

C More Implementation Details

C.1 Volume Latent Generation

Before stream AR training, we use the frozen non-causal VidTok-KL-488-16chn tokenizer (Tang et al., 2024) to precompute full-volume continuous latent representations for all modalities. We extract the deterministic posterior mean, yielding 16-channel latents without stochastic sampling. Each modality is encoded and cached independently, while all modalities from the same case share spatial preprocessing, spatial grids, padding, and tile coordinates to preserve voxel-wise correspondence, with intensity normalization performed separately for each modality. Full-volume latents are constructed through overlapping tiled encoding with 25% overlap, followed by weighted blending of the tile-wise posterior means. Each cache stores the resulting continuous latent tensor, together with latent-space validity and anatomy masks and spatial metadata. During autoregressive training, spatially aligned, fixed-size source–target latent windows are sampled from these cached representations, avoiding repeated execution of the VidTok encoder.

C.2 Training and Inference Details

Data Preprocessing. We adopt dataset-specific preprocessing to account for differences in image intensity distributions and spatial resolution across modalities, with the detailed settings summarized in Table 6. For BraTS 2024 (de Verdier et al., 2024), each volume is resampled to a common size of $256 \times 256 \times 128$. In contrast, SynthRAD 2025 (Thummerer et al., 2025) and AutoPET (Dexl et al., 2026) exhibit considerable variation in native volume size across subjects, so we retain their original full-volume dimensions rather than enforcing a fixed spatial shape. For every paired sample, the source and target images are kept on an identical spatial grid during preprocessing to maintain voxel-wise correspondence.

Tokenization and Stream Generation. We use the frozen non-causal VidTok-KL (Tang et al., 2024) tokenizer and cached posterior-mean latents described in Appendix C.1. Its compression ratio of $(8, 8, 4)$ in (H, W, D) maps each $192 \times 192 \times 96$ image crop to a $24 \times 24 \times 24$ latent grid with 16 channels. The 3D Causal DiT operates on $2 \times 2 \times 2$ latent patches, conditioned on the source and previously generated refinement states. We use $K_{\max} = 4$ with four Euler integration steps per state. The first state starts from Gaussian noise (Liu et al., 2023b; Lipman et al., 2023), while subsequent states refine the detached previous prediction.

Optimization. We jointly train the refiner and benefit head for 100k optimization steps using AdamW with $(\beta_1, \beta_2) = (0.9, 0.95)$. The learning rate increases to 5×10^{-5} during warmup and then follows a cosine decay schedule. We use bfloat16 mixed precision and clip the global gradient norm to

Table 7: **Dataset-specific configurations** of the **`StreamRefine`** architecture, training procedure, and inference protocol.

Setting	BraTS 2024	SynthRAD 2025	AutoPET
Model Configurations			
Layers	12	6	8
Hidden dim	512	384	512
Attention heads	8	8	8
Patch size	$2 \times 2 \times 2$	$2 \times 2 \times 2$	$2 \times 2 \times 2$
$K_{\max}$	4	4	4
Diffusion steps	4	4	4
Parameters	51.19M	19.39M	34.37M
FLOPs	1734.8G	711.2G	1158.5G
Training Settings			
Optimizer	AdamW	AdamW	AdamW
Learning rate	5e−5	5e−5	5e−5
(β_1, β_2)	$(0.9, 0.95)$	$(0.9, 0.95)$	$(0.9, 0.95)$
Learning rate schedule	Warmup cosine	Warmup cosine	Warmup cosine
Batch size	6	8	6
Training steps	100k	100k	100k
Gradient clipping	1.0	1.0	1.0
Precision	bfloat16	bfloat16	bfloat16
Training crop size	$192 \times 192 \times 96$	$192 \times 192 \times 96$	$192 \times 192 \times 96$
Latent window size	$24 \times 24 \times 24$	$24 \times 24 \times 24$	$24 \times 24 \times 24$
Inference Settings			
Inference protocol	Sliding window inference	Sliding window inference	Sliding window inference
Inference window size	$192 \times 192 \times 96$	$192 \times 192 \times 96$	$192 \times 192 \times 96$
Inference overlap	0.25	0.25	0.25

1.0. Batch sizes are 6, 8, and 6 for BraTS 2024, SynthRAD 2025, and AutoPET, respectively. All experiments are conducted on a single NVIDIA GeForce RTX 5090 GPU. Dataset-specific model configurations are summarized in Table 7.

Inference. For all datasets, whole-volume synthesis uses sliding windows of $192 \times 192 \times 96$ voxels, corresponding to $24 \times 24 \times 24$ latent windows, with 25% overlap. Within each window, KV caching reuses source and refinement-history features during flow integration. At each refinement step, overlapping predictions are fused using Gaussian-weighted blending to form a complete latent volume. The fused state supplies the history for subsequent refinement, with the stopping decision shared across the volume as detailed below. The final latent volume is decoded by the frozen VidTok decoder. Inference requires neither paired targets, PIR computation, nor *Continue-to-Check* exploration.

Volume-level Stopping. For volumes that require sliding-window inference, we use a single stopping decision for the entire volume rather than stopping each window independently. At refinement step k, all windows are advanced synchronously and their latent predictions are blended into a global volume state. Let $\widehat{B}_{k,w}$ denote the benefit predicted for window w. We aggregate the window scores as

$$\widehat{B}_k^{\text{vol}} = \frac{\sum_w \omega_w \widehat{B}_{k,w}}{\sum_w \omega_w}, \tag{19}$$

where ω_w is the validity- and overlap-aware weight of window w. The whole volume stops when $\widehat{B}_k^{\text{vol}} \leq 0$ or $k = K_{\max}$; otherwise, the fused global state is used for the next refinement step. This synchronized decision keeps all windows at the same refinement depth and avoids inconsistencies caused by independently stopping overlapping windows. This aggregation serves as a practical surrogate for synchronized volume-level stopping rather than an exact evaluation of the fused-volume *hindsight benefit*; its interpretation and limitations are discussed in Appendix C.4.

C.3 Tokenizer Reconstruction Performance

We use the publicly released VidTok-KL checkpoint `vidtok_kl_noncausal_488_16chn` (Tang et al., 2024) as the frozen visual tokenizer throughout all experiments. As shown in Table 8, VidTok

Table 8: **Reconstruction performance** of the VidTok tokenizer across different datasets and modalities. Averages are computed over all evaluated volumes before rounding.

Dataset	Modality	PSNR↑	SSIM↑	MAE↓
BraTS 2024	T1n	37.05	0.974	0.013
	T1c	37.67	0.981	0.013
	T2w	38.65	0.986	0.010
	T2f	37.45	0.978	0.013
	Avg	37.71	0.979	0.012
SynthRAD 2025	MRI	38.07	0.973	0.013
	CT	40.68	0.982	0.009
	CBCT	40.43	0.969	0.010
	Avg	40.07	0.976	0.010
AutoPET	CT	40.49	0.981	0.006
	PET	38.99	0.986	0.007
	Avg	39.74	0.983	0.007

achieves consistently high reconstruction fidelity across all datasets and modalities, with PSNR ranging from 37.05 to 40.68 dB and SSIM from 0.969 to 0.986. The corresponding MAE remains low (0.006–0.013), indicating that the tokenizer preserves both image appearance and structural content well across heterogeneous medical imaging modalities. These results support the use of the frozen VidTok latent space for subsequent medical image translation and iterative refinement.

C.4 Interpretation of Volume-Level Stopping

Volume-level aggregation. Eq. 19 aggregates crop-level benefit predictions into a single stopping decision for the whole volume. This keeps overlapping windows at the same refinement depth and allows the fused state to be used consistently in the next step. Because the crop-level benefit includes a maximum over future refinement depths, this aggregation is a practical synchronized stopping rule rather than an exact fused-volume utility.

To see the distinction, let $\alpha_w = \omega_w / \sum_u \omega_u$ be the normalized window weight, and let $g_{k,w,j}$ denote the net benefit of continuing window w from step k to a common future depth j:

$$g_{k,w,j} = (\bar{L}_{k,w} - \bar{L}_{j,w}) - \lambda_a(\bar{A}_{j,w} - \bar{A}_{k,w}) - \lambda_c(j-k). \tag{20}$$

Then

$$\sum_w \alpha_w \max_{k<j\le K_{\max}} g_{k,w,j} \;\ge\; \max_{k<j\le K_{\max}} \sum_w \alpha_w g_{k,w,j}. \tag{21}$$

The left side allows each window to favor a different future depth, whereas the right side uses one shared depth for the volume. The two coincide when the relevant windows favor the same future depth, and the discrepancy grows when their refinement preferences differ. Since the weights are normalized, it is not determined by the number of windows alone, although more heterogeneous regions can increase the disagreement.

Relation to stopping evaluation. This comparison only clarifies the effect of aggregating crop-level benefits; losses measured after overlapping windows are fused need not be exactly additive across crops. In practice, Sec. 4.5 evaluates stopping on shared volume-level trajectories. The *predicted* and *hindsight benefits* obtain a correlation of 0.811 with 86.0% sign agreement, while the learned rule reduces stopping regret from 0.105 for matched-random stopping to 0.023. These results support the use of Eq. 19 as a practical volume-level stopping rule, while the aggregation gap itself is not separately measured.

D Further Experimental Results

D.1 Statistical Significance of Main Results

Tables 9 and 10 report subject-level variability together with paired statistical comparisons for the main results. Across the three datasets, the paired tests broadly agree with the performance trends

Table 9: **Subject-level statistical analysis on BraTS 2024**. We report the mean performance with subject-level standard deviations for five translation tasks. Asterisks indicate statistically significant differences between each competing method and **`StreamRefine`** based on two-sided paired Wilcoxon signed-rank tests: $^{*}p < 0.05$, $^{**}p < 0.01$, and $^{***}p < 0.001$. Subscripts denote subject-level standard deviations.

Methods	T2w→T2f		T1c→T1n		T2f→T2w	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
I^2SB (Liu et al., 2023a)	$20.88^{***}_{\pm 2.18}$	$0.656^{***}_{\pm 0.05}$	$24.72^{***}_{\pm 2.03}$	$0.821^{***}_{\pm 0.04}$	$22.12^{***}_{\pm 1.85}$	$0.735^{***}_{\pm 0.03}$
DBIM (Zheng et al., 2025)	$21.08^{***}_{\pm 2.06}$	$0.664^{***}_{\pm 0.05}$	$24.97^{***}_{\pm 1.97}$	$0.827^{***}_{\pm 0.04}$	$22.31^{***}_{\pm 2.21}$	$0.742^{***}_{\pm 0.04}$
selfRDB (Arslan et al., 2025)	$21.42^{***}_{\pm 1.77}$	$0.674^{***}_{\pm 0.04}$	$25.16^{***}_{\pm 1.98}$	$0.834^{***}_{\pm 0.05}$	$22.48^{***}_{\pm 1.93}$	$0.748^{***}_{\pm 0.04}$
DM-SAM (Pang et al., 2026)	$21.35^{***}_{\pm 1.93}$	$0.671^{***}_{\pm 0.04}$	$25.23^{**}_{\pm 1.72}$	$0.832^{**}_{\pm 0.03}$	$22.94^{**}_{\pm 2.07}$	$0.771^{***}_{\pm 0.05}$
CRSG (Song et al., 2026)	$21.19^{**}_{\pm 2.13}$	$0.670^{*}_{\pm 0.06}$	$25.31^{**}_{\pm 2.16}$	$0.834^{**}_{\pm 0.04}$	$22.77^{***}_{\pm 1.89}$	$0.762^{***}_{\pm 0.03}$
Fixed-K Refine	$21.41^{***}_{\pm 2.02}$	$0.679^{**}_{\pm 0.04}$	$25.52^{***}_{\pm 1.96}$	$0.835^{***}_{\pm 0.05}$	$22.89^{**}_{\pm 2.18}$	$0.766^{***}_{\pm 0.04}$
Post-hoc Stop	$21.52^{**}_{\pm 2.25}$	$0.678^{**}_{\pm 0.04}$	$25.64^{***}_{\pm 2.23}$	$0.840^{*}_{\pm 0.05}$	$23.02^{***}_{\pm 2.11}$	$0.769^{***}_{\pm 0.05}$
`StreamRefine`	**21.68** $_{\pm 2.19}$	**0.681** $_{\pm 0.05}$	**25.76** $_{\pm 2.36}$	**0.841** $_{\pm 0.06}$	**23.14** $_{\pm 2.04}$	**0.776** $_{\pm 0.06}$

Methods	T1n→T2w		T1n→T1c	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑
I^2SB (Liu et al., 2023a)	$22.25^{***}_{\pm 2.29}$	$0.739^{***}_{\pm 0.04}$	$21.44^{***}_{\pm 2.44}$	$0.681^{***}_{\pm 0.06}$
DBIM (Zheng et al., 2025)	$22.46^{***}_{\pm 1.97}$	$0.746^{***}_{\pm 0.05}$	$21.76^{***}_{\pm 2.12}$	$0.688^{***}_{\pm 0.05}$
selfRDB (Arslan et al., 2025)	$22.68^{**}_{\pm 2.19}$	$0.761^{**}_{\pm 0.05}$	$22.03^{***}_{\pm 2.30}$	$0.695^{***}_{\pm 0.06}$
DM-SAM (Pang et al., 2026)	$22.83^{***}_{\pm 2.16}$	$0.766^{***}_{\pm 0.04}$	$22.08^{*}_{\pm 2.43}$	$0.697^{**}_{\pm 0.04}$
CRSG (Song et al., 2026)	$22.78^{***}_{\pm 2.28}$	$0.759^{***}_{\pm 0.04}$	$21.85^{**}_{\pm 2.55}$	$0.691^{***}_{\pm 0.05}$
Fixed-K Refine	$23.25^{*}_{\pm 2.42}$	$0.779^{***}_{\pm 0.03}$	$22.34^{***}_{\pm 1.95}$	$0.702^{***}_{\pm 0.04}$
Post-hoc Stop	$23.18^{**}_{\pm 1.86}$	$0.779^{***}_{\pm 0.04}$	$22.36^{***}_{\pm 2.12}$	$0.700^{***}_{\pm 0.03}$
`StreamRefine`	**23.31** $_{\pm 1.90}$	**0.778** $_{\pm 0.04}$	**22.61** $_{\pm 2.38}$	**0.704** $_{\pm 0.04}$

in Tables 1 and 2, with **`StreamRefine`** showing consistent PSNR improvements and comparable or improved SSIM across most tasks. This provides additional subject-level support for the main quantitative results.

D.2 Multi-seed Robustness Analysis

We repeat training with three independent seeds for all five BraTS 2024 (de Verdier et al., 2024) translation tasks, with results summarized in Table 11. Across seeds, **`StreamRefine`** maintains stable performance with only modest variation in PSNR, SSIM, and FID. The three-seed averages also closely match the main results, indicating that the observed performance is not sensitive to training randomness.

D.3 Tumor- and Lesion-ROI Evaluation

Tables 12 and 13 provide subject-level analyses of tumor- and lesion-ROI reconstruction together with downstream segmentation on BraTS 2024 (de Verdier et al., 2024) and AutoPET (Dexl et al., 2026). Across the evaluated tasks, **`StreamRefine`** generally achieves stronger ROI PSNR/SSIM and Dice, with many of the paired differences reaching statistical significance. The qualitative comparisons in Fig. 8 further illustrate the downstream tumor segmentation results across all five BraTS translation tasks. These findings provide additional evidence that the observed translation gains extend to anatomy-related regions and downstream evaluation.

D.4 Additional Qualitative Results

Figs. 10–13 extend the qualitative comparison to the remaining translation tasks across BraTS 2024 (de Verdier et al., 2024), SynthRAD 2025 (Thummerer et al., 2025), and AutoPET (Dexl et al., 2026). For each task, we provide axial, coronal, and sagittal views together with corresponding error maps. Across the selected cases, **`StreamRefine`** generally produces predictions closer to the target

Table 10: **Subject-level statistical analysis on SynthRAD 2025 and AutoPET**. We report the mean performance with subject-level standard deviations for five translation tasks. Asterisks indicate statistically significant differences between each competing method and **`StreamRefine`** based on two-sided paired Wilcoxon signed-rank tests: $^{*}p < 0.05$, $^{**}p < 0.01$, and $^{***}p < 0.001$. Subscripts denote subject-level standard deviations.

Methods	SynthRAD 2025					
	MRI→CT		CBCT→CT		CT→MRI	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
I^2SB (Liu et al., 2023a)	$20.27^{***}_{\pm 1.82}$	$0.689^{***}_{\pm 0.07}$	$22.71^{***}_{\pm 2.32}$	$0.730^{***}_{\pm 0.06}$	$17.94^{***}_{\pm 2.11}$	$0.642^{***}_{\pm 0.06}$
DBIM (Zheng et al., 2025)	$20.53^{***}_{\pm 2.29}$	$0.697^{***}_{\pm 0.06}$	$22.87^{***}_{\pm 2.16}$	$0.731^{**}_{\pm 0.04}$	$18.19^{***}_{\pm 1.98}$	$0.654^{***}_{\pm 0.03}$
selfRDB (Arslan et al., 2025)	$20.81^{***}_{\pm 1.95}$	$0.708^{**}_{\pm 0.08}$	$23.19^{*}_{\pm 1.98}$	$0.740^{**}_{\pm 0.08}$	$18.48^{***}_{\pm 1.83}$	$0.670^{**}_{\pm 0.05}$
DM-SAM (Pang et al., 2026)	$21.09^{***}_{\pm 2.20}$	$0.712^{***}_{\pm 0.06}$	$23.31^{**}_{\pm 1.87}$	$0.741^{***}_{\pm 0.07}$	$18.87^{*}_{\pm 2.10}$	$0.679^{***}_{\pm 0.04}$
CRSG (Song et al., 2026)	$20.62^{**}_{\pm 2.06}$	$0.694^{***}_{\pm 0.07}$	$22.78^{***}_{\pm 1.74}$	$0.732^{***}_{\pm 0.05}$	$18.61^{***}_{\pm 2.09}$	$0.668^{***}_{\pm 0.05}$
Fixed-K Refine	$21.08^{***}_{\pm 2.40}$	$0.712^{**}_{\pm 0.05}$	$23.54^{***}_{\pm 2.19}$	$0.751^{***}_{\pm 0.04}$	$18.72^{**}_{\pm 1.91}$	$0.677^{**}_{\pm 0.05}$
Post-hoc Stop	$21.16^{**}_{\pm 2.31}$	$0.718^{**}_{\pm 0.06}$	$23.50^{***}_{\pm 2.24}$	$0.748^{***}_{\pm 0.04}$	$18.84^{***}_{\pm 2.20}$	$0.683^{**}_{\pm 0.05}$
`StreamRefine`	**21.28** $_{\pm 2.19}$	**0.716** $_{\pm 0.06}$	**23.63** $_{\pm 2.34}$	**0.753** $_{\pm 0.05}$	**18.92** $_{\pm 2.04}$	**0.686** $_{\pm 0.07}$

Methods	AutoPET			
	CT→PET		PET→CT	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑
I^2SB (Liu et al., 2023a)	$26.04^{***}_{\pm 1.25}$	$0.917^{**}_{\pm 0.02}$	$23.25^{***}_{\pm 0.92}$	$0.889^{***}_{\pm 0.02}$
DBIM (Zheng et al., 2025)	$25.97^{***}_{\pm 1.39}$	$0.920^{***}_{\pm 0.03}$	$23.53^{***}_{\pm 1.07}$	$0.895^{***}_{\pm 0.03}$
selfRDB (Arslan et al., 2025)	$26.08^{***}_{\pm 1.51}$	$0.923^{**}_{\pm 0.03}$	$23.67^{**}_{\pm 1.01}$	$0.901^{***}_{\pm 0.02}$
DM-SAM (Pang et al., 2026)	$26.72^{**}_{\pm 1.42}$	$0.938^{***}_{\pm 0.03}$	$24.16^{**}_{\pm 1.19}$	$0.908^{*}_{\pm 0.03}$
CRSG (Song et al., 2026)	$26.85^{***}_{\pm 1.27}$	$0.937^{**}_{\pm 0.02}$	$23.98^{**}_{\pm 1.12}$	$0.903^{**}_{\pm 0.02}$
Fixed-K Refine	$27.46^{**}_{\pm 1.29}$	$0.942^{**}_{\pm 0.02}$	$24.30^{***}_{\pm 1.03}$	$0.909^{***}_{\pm 0.02}$
Post-hoc Stop	$27.39^{***}_{\pm 1.46}$	$0.944^{***}_{\pm 0.03}$	$24.35^{***}_{\pm 1.10}$	$0.909^{***}_{\pm 0.02}$
`StreamRefine`	**27.65** $_{\pm 1.38}$	**0.950** $_{\pm 0.02}$	**24.46** $_{\pm 0.95}$	**0.912** $_{\pm 0.02}$

Table 11: **Multi-seed robustness analysis on BraTS 2024**. We report the mean performance over three independent training runs, with subscripts denoting standard deviations across seeds.

Methods	T2w→T2f			T1c→T1n			T2f→T2w		
	PSNR↑	SSIM↑	FID↓	PSNR↑	SSIM↑	FID↓	PSNR↑	SSIM↑	FID↓
Fixed-K Refine	21.39 $_{\pm 0.17}$	0.678 $_{\pm 0.005}$	47.5 $_{\pm 1.1}$	25.49 $_{\pm 0.21}$	0.835 $_{\pm 0.004}$	33.2 $_{\pm 0.8}$	22.93 $_{\pm 0.21}$	0.765 $_{\pm 0.005}$	34.4 $_{\pm 1.0}$
Post-hoc Stop	21.55 $_{\pm 0.18}$	0.679 $_{\pm 0.003}$	44.6 $_{\pm 0.8}$	25.66 $_{\pm 0.19}$	0.840 $_{\pm 0.005}$	30.8 $_{\pm 1.3}$	23.06 $_{\pm 0.15}$	0.771 $_{\pm 0.003}$	33.9 $_{\pm 0.9}$
`StreamRefine`	**21.71** $_{\pm 0.20}$	**0.682** $_{\pm 0.004}$	**43.1** $_{\pm 1.1}$	**25.75** $_{\pm 0.17}$	**0.841** $_{\pm 0.003}$	**31.5** $_{\pm 0.9}$	**23.17** $_{\pm 0.16}$	**0.778** $_{\pm 0.004}$	**31.6** $_{\pm 1.3}$

Methods	T1n→T2w			T1n→T1c		
	PSNR↑	SSIM↑	FID↓	PSNR↑	SSIM↑	FID↓
Fixed-K Refine	23.22 $_{\pm 0.22}$	0.777 $_{\pm 0.004}$	34.0 $_{\pm 0.8}$	22.36 $_{\pm 0.17}$	0.703 $_{\pm 0.004}$	41.4 $_{\pm 1.0}$
Post-hoc Stop	23.20 $_{\pm 0.16}$	0.778 $_{\pm 0.004}$	32.5 $_{\pm 0.8}$	22.36 $_{\pm 0.19}$	0.702 $_{\pm 0.005}$	42.1 $_{\pm 1.3}$
`StreamRefine`	**23.34** $_{\pm 0.15}$	**0.779** $_{\pm 0.003}$	**31.7** $_{\pm 1.2}$	**22.61** $_{\pm 0.21}$	**0.705** $_{\pm 0.004}$	**39.0** $_{\pm 1.1}$

appearance with reduced localized errors. These examples complement the quantitative results and illustrate the cross-dataset behavior of **`StreamRefine`** across different translation directions.

D.5 Additional Ablation Studies

Effect of additional design choices. We further examine several design choices of **`StreamRefine`**, including reach correction, continuation frequency, the definition of *Continue Benefit*, computation cost, the use of the anatomy proxy, and the refinement horizon. All variants otherwise follow the same training and evaluation settings as the main ablation study.

Table 12: **Subject-level statistical analysis of tumor-ROI reconstruction and downstream tumor segmentation on BraTS 2024**. We report the mean performance with subject-level standard deviations for five translation tasks. Asterisks indicate statistically significant differences between each competing method and **`StreamRefine`** based on two-sided paired Wilcoxon signed-rank tests: $^{*}p < 0.05$, $^{**}p < 0.01$, and $^{***}p < 0.001$. Subscripts denote subject-level standard deviations.

Methods	T2w→T2f			T1c→T1n			T2f→T2w		
	PSNR↑	SSIM↑	Dice↑	PSNR↑	SSIM↑	Dice↑	PSNR↑	SSIM↑	Dice↑
I^{2}SB (Liu et al., 2023a)	17.92 $^{***}_{\pm 3.05}$	0.651 $^{***}_{\pm 0.10}$	0.737 $^{***}_{\pm 0.10}$	20.56 $^{***}_{\pm 2.84}$	0.778 $^{***}_{\pm 0.09}$	0.802 $^{***}_{\pm 0.11}$	18.84 $^{***}_{\pm 3.14}$	0.697 $^{***}_{\pm 0.11}$	0.758 $^{***}_{\pm 0.12}$
DBIM (Zheng et al., 2025)	17.81 $^{***}_{\pm 3.39}$	0.645 $^{***}_{\pm 0.15}$	0.740 $^{***}_{\pm 0.13}$	20.88 $^{***}_{\pm 3.44}$	0.789 $^{***}_{\pm 0.12}$	0.816 $^{***}_{\pm 0.11}$	19.10 $^{***}_{\pm 3.16}$	0.709 $^{***}_{\pm 0.13}$	0.772 $^{***}_{\pm 0.09}$
selfRDB (Arslan et al., 2025)	18.15 $^{**}_{\pm 2.98}$	0.668 $^{*}_{\pm 0.09}$	0.761 $^{**}_{\pm 0.12}$	21.12 $^{***}_{\pm 3.11}$	0.795 $^{***}_{\pm 0.07}$	0.825 $^{***}_{\pm 0.12}$	19.42 $^{***}_{\pm 2.97}$	0.719 $^{***}_{\pm 0.08}$	0.791 $^{***}_{\pm 0.12}$
DM-SAM (Pang et al., 2026)	18.28 $^{**}_{\pm 3.09}$	0.663 $^{***}_{\pm 0.12}$	0.754 $^{***}_{\pm 0.13}$	21.46 $^{**}_{\pm 3.31}$	0.803 $^{*}_{\pm 0.11}$	0.833 $^{**}_{\pm 0.08}$	19.71 $^{***}_{\pm 3.35}$	0.727 $^{***}_{\pm 0.12}$	0.803 $^{***}_{\pm 0.09}$
CRSG (Song et al., 2026)	18.03 $^{***}_{\pm 2.72}$	0.665 $^{**}_{\pm 0.09}$	0.749 $^{***}_{\pm 0.12}$	21.57 $^{*}_{\pm 2.69}$	0.801 $^{*}_{\pm 0.09}$	0.837 $^{**}_{\pm 0.09}$	19.58 $^{***}_{\pm 3.19}$	0.724 $^{***}_{\pm 0.10}$	0.798 $^{***}_{\pm 0.10}$
Fixed-K Refine	18.21 $^{***}_{\pm 3.33}$	0.668 $^{*}_{\pm 0.13}$	0.763 $^{**}_{\pm 0.14}$	21.38 $^{***}_{\pm 3.57}$	0.801 $^{**}_{\pm 0.12}$	0.833 $^{***}_{\pm 0.12}$	19.76 $^{***}_{\pm 3.52}$	0.725 $^{***}_{\pm 0.11}$	0.801 $^{***}_{\pm 0.13}$
Post-hoc Stop	18.25 $^{***}_{\pm 3.07}$	0.664 $^{***}_{\pm 0.11}$	0.760 $^{***}_{\pm 0.13}$	21.49 $^{***}_{\pm 3.24}$	0.800 $^{***}_{\pm 0.10}$	0.838 $^{***}_{\pm 0.10}$	19.73 $^{***}_{\pm 3.30}$	0.730 $^{**}_{\pm 0.08}$	0.807 $^{***}_{\pm 0.10}$
`StreamRefine`	**18.37** $_{\pm 3.29}$	**0.674** $_{\pm 0.09}$	**0.768** $_{\pm 0.12}$	**21.68** $_{\pm 3.45}$	**0.805** $_{\pm 0.13}$	**0.845** $_{\pm 0.10}$	**19.89** $_{\pm 3.60}$	**0.735** $_{\pm 0.11}$	**0.815** $_{\pm 0.11}$

Methods	T1n→T2w			T1n→T1c		
	PSNR↑	SSIM↑	Dice↑	PSNR↑	SSIM↑	Dice↑
I^{2}SB (Liu et al., 2023a)	18.67 $^{***}_{\pm 2.97}$	0.695 $^{***}_{\pm 0.08}$	0.767 $^{***}_{\pm 0.13}$	19.23 $^{***}_{\pm 2.74}$	0.671 $^{***}_{\pm 0.10}$	0.751 $^{***}_{\pm 0.12}$
DBIM (Zheng et al., 2025)	18.93 $^{***}_{\pm 3.37}$	0.702 $^{***}_{\pm 0.13}$	0.762 $^{***}_{\pm 0.15}$	19.51 $^{***}_{\pm 3.11}$	0.681 $^{***}_{\pm 0.12}$	0.769 $^{***}_{\pm 0.13}$
selfRDB (Arslan et al., 2025)	19.28 $^{***}_{\pm 3.08}$	0.711 $^{***}_{\pm 0.11}$	0.786 $^{***}_{\pm 0.12}$	19.86 $^{***}_{\pm 3.23}$	0.692 $^{***}_{\pm 0.10}$	0.788 $^{***}_{\pm 0.11}$
DM-SAM (Pang et al., 2026)	19.49 $^{***}_{\pm 3.17}$	0.719 $^{*}_{\pm 0.10}$	0.797 $^{***}_{\pm 0.13}$	20.02 $^{***}_{\pm 3.00}$	0.699 $^{*}_{\pm 0.09}$	0.794 $^{***}_{\pm 0.13}$
CRSG (Song et al., 2026)	19.56 $^{**}_{\pm 3.02}$	0.716 $^{***}_{\pm 0.09}$	0.801 $^{***}_{\pm 0.10}$	20.07 $^{***}_{\pm 2.89}$	0.692 $^{**}_{\pm 0.08}$	0.782 $^{***}_{\pm 0.12}$
Fixed-K Refine	19.59 $^{**}_{\pm 3.48}$	0.722 $^{*}_{\pm 0.13}$	0.801 $^{***}_{\pm 0.12}$	20.18 $^{*}_{\pm 2.79}$	0.696 $^{**}_{\pm 0.10}$	0.802 $^{**}_{\pm 0.14}$
Post-hoc Stop	19.52 $^{***}_{\pm 2.90}$	0.719 $^{**}_{\pm 0.10}$	0.808 $^{*}_{\pm 0.11}$	20.10 $^{***}_{\pm 3.06}$	0.696 $^{**}_{\pm 0.11}$	0.797 $^{***}_{\pm 0.15}$
`StreamRefine`	**19.68** $_{\pm 3.24}$	**0.726** $_{\pm 0.12}$	**0.812** $_{\pm 0.13}$	**20.21** $_{\pm 3.10}$	**0.701** $_{\pm 0.10}$	**0.807** $_{\pm 0.12}$

Table 13: **Subject-level statistical analysis of lesion-ROI reconstruction and downstream lesion segmentation on AutoPET**. We report the mean performance with subject-level standard deviations for two translation tasks. Asterisks indicate statistically significant differences between each competing method and **`StreamRefine`** based on two-sided paired Wilcoxon signed-rank tests: $^{*}p < 0.05$, $^{**}p < 0.01$, and $^{***}p < 0.001$. Subscripts denote subject-level standard deviations.

Methods	CT→PET			PET→CT		
	PSNR↑	SSIM↑	Dice↑	PSNR↑	SSIM↑	Dice↑
I^{2}SB (Liu et al., 2023a)	17.05 $^{***}_{\pm 1.43}$	0.892 $^{***}_{\pm 0.03}$	0.551 $^{***}_{\pm 0.13}$	17.26 $^{***}_{\pm 1.46}$	0.773 $^{***}_{\pm 0.02}$	0.572 $^{***}_{\pm 0.14}$
DBIM (Zheng et al., 2025)	16.89 $^{***}_{\pm 1.50}$	0.894 $^{***}_{\pm 0.02}$	0.542 $^{***}_{\pm 0.17}$	17.37 $^{***}_{\pm 1.38}$	0.782 $^{***}_{\pm 0.03}$	0.569 $^{***}_{\pm 0.15}$
selfRDB (Arslan et al., 2025)	17.29 $^{***}_{\pm 1.36}$	0.907 $^{***}_{\pm 0.03}$	0.567 $^{*}_{\pm 0.14}$	17.52 $^{***}_{\pm 1.60}$	0.782 $^{**}_{\pm 0.04}$	0.582 $^{**}_{\pm 0.13}$
DM-SAM (Pang et al., 2026)	17.36 $^{***}_{\pm 1.54}$	0.905 $^{***}_{\pm 0.03}$	0.560 $^{***}_{\pm 0.15}$	17.84 $^{***}_{\pm 1.47}$	0.785 $^{***}_{\pm 0.03}$	0.592 $^{**}_{\pm 0.15}$
CRSG (Song et al., 2026)	17.51 $^{***}_{\pm 1.34}$	0.913 $^{**}_{\pm 0.02}$	0.565 $^{***}_{\pm 0.16}$	17.79 $^{***}_{\pm 1.27}$	0.788 $^{***}_{\pm 0.02}$	0.586 $^{**}_{\pm 0.13}$
Fixed-K Refine	17.68 $^{*}_{\pm 1.53}$	0.912 $_{\pm 0.03}$	0.571 $^{*}_{\pm 0.14}$	18.00 $^{***}_{\pm 1.49}$	0.787 $^{*}_{\pm 0.02}$	0.593 $^{**}_{\pm 0.14}$
Post-hoc Stop	17.62 $^{**}_{\pm 1.49}$	0.911 $^{*}_{\pm 0.02}$	0.574 $_{\pm 0.14}$	18.03 $^{*}_{\pm 1.35}$	0.789 $_{\pm 0.03}$	0.596 $^{*}_{\pm 0.14}$
`StreamRefine`	**17.82** $_{\pm 1.39}$	**0.915** $_{\pm 0.03}$	**0.580** $_{\pm 0.15}$	**18.14** $_{\pm 1.54}$	**0.794** $_{\pm 0.03}$	**0.601** $_{\pm 0.13}$

- **w/o Refiner Reach Correction.** We remove the inverse-reach weighting for refinement losses after the proposed stop, while keeping *Continue-to-Check* and the stopping-head weighting unchanged. This tests whether correcting the lower observation frequency of later refinements is useful.
- **Full Continuation ($\epsilon = 1$).** All training trajectories continue to $K_{\max}$, so later refinements and the benefit head receive *full-horizon* supervision. Inference remains adaptive and uses the learned stopping head as in **`StreamRefine`**.
- **Next-step Benefit.** We replace the maximum benefit over all future states with the net benefit of only the next refinement state. This evaluates whether considering delayed improvements across multiple steps is necessary.
- **w/o Compute Cost ($\lambda_c = 0$).** We remove the computation-cost term from *Continue Benefit* while keeping the translation and anatomy terms unchanged. This isolates the role of explicitly penalizing additional refinement steps.
- **Anatomy in Refiner Loss.** We remove the anatomy term from the stopping target and instead use the same structural proxy as an additional refiner regularizer. This tests whether anatomy is more useful for deciding when to stop or for directly training refinement.

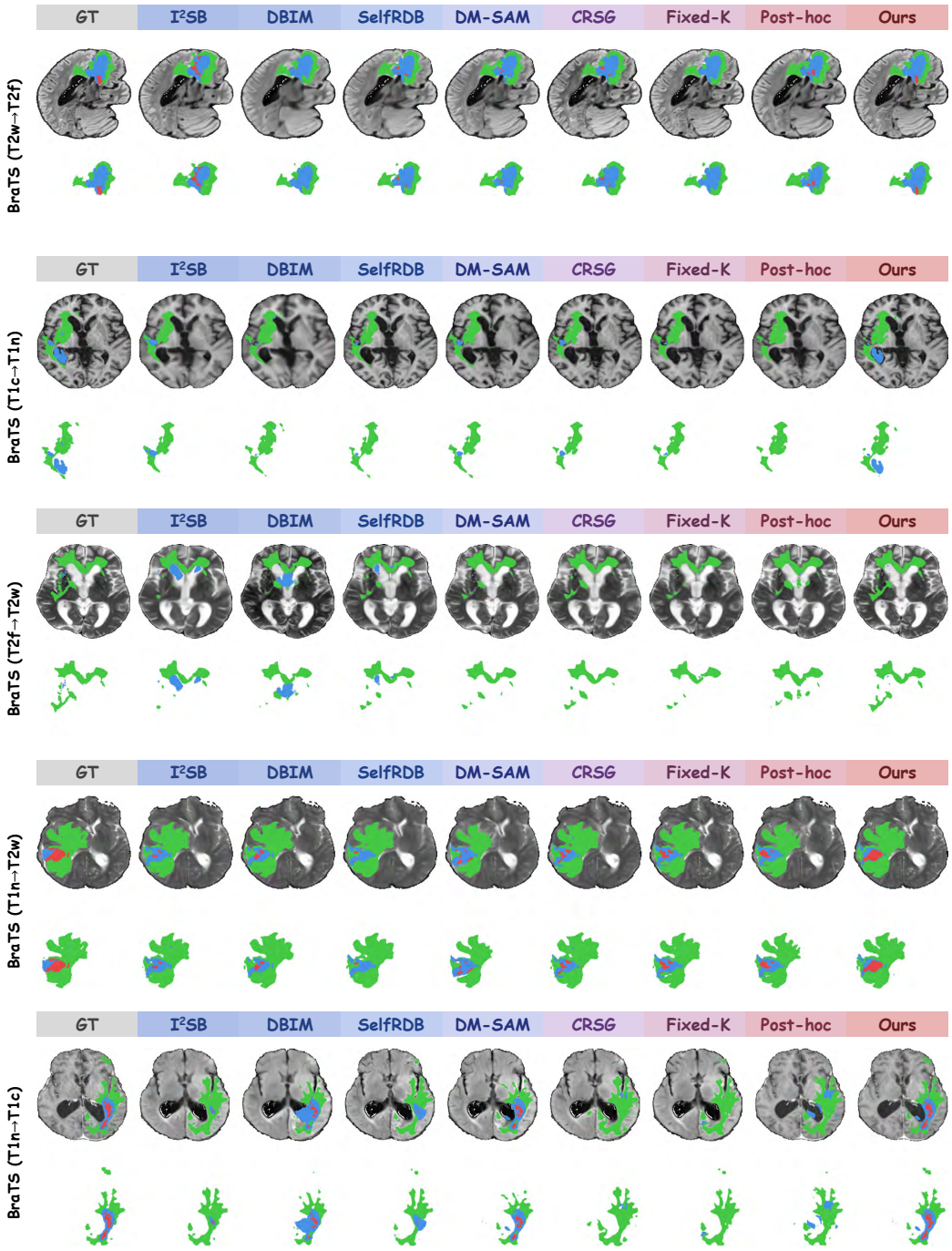


Figure 8: **Qualitative comparison of downstream tumor segmentation on BraTS 2024.** From top to bottom: T2w→T2f, T1c→T1n, T2f→T2w, T1n→T2w, and T1n→T1c. Segmentation overlays and corresponding masks are shown for each method.

- **Fixed Horizon ($K = 2$).** We disable learned stopping and execute exactly two refinement states for every case. This provides an intermediate *fixed-horizon* reference between *One-step* translation and the full $K_{\max} = 4$ refinement stream.

Table 14 further examines several additional design choices. Removing reach correction, using only next-step benefit, or moving anatomy supervision into the refiner generally reduces translation

Table 14: **Additional ablation studies.** We further examine design choices in training, stopping, and refinement. Avg. Steps denotes the average refinement steps at inference.

Variants	BraTS T2w→T2f				AutoPET CT→PET			
	PSNR↑	FID↓	Dice↑	Avg. Steps↓	PSNR↑	FID↓	Dice↑	Avg. Steps↓
Training Strategy								
w/o Refiner Reach Correction	21.51	45.9	0.761	2.05	27.45	49.7	0.575	2.39
Full Continuation ($\epsilon = 1$)	**21.70**	44.1	**0.770**	2.02	27.58	48.9	**0.580**	2.46
Benefit Design								
Next-step Benefit	21.41	48.0	0.755	**1.90**	27.55	47.8	0.570	2.28
w/o Compute Cost ($\lambda_c = 0$)	21.65	43.9	0.768	2.81	27.62	**47.0**	0.578	3.06
Alternative Design								
Anatomy in Refiner Loss	21.53	47.1	0.752	2.40	27.49	50.6	0.573	2.82
Fixed Horizon ($K = 2$)	21.34	49.0	0.757	2.00	27.23	53.2	0.568	**2.00**
`StreamRefine`	21.68	**43.5**	0.768	2.13	**27.65**	47.5	**0.580**	2.57

Table 15: **Sensitivity to the anatomy weight** λ_a on BraTS T2w→T2f.

λ_a	BraTS T2w→T2f		
	PSNR↑	Dice↑	Avg. Steps↓
0.5	21.59	0.761	2.26
1.0	**21.68**	**0.768**	2.13
2.0	21.62	**0.768**	2.01
4.0	21.50	0.763	**1.89**

Table 16: **Sensitivity to the compute-cost weight** λ_c on BraTS T2w→T2f.

λ_c	BraTS T2w→T2f		
	PSNR↑	Dice↑	Avg. Steps↓
0.05	21.63	0.765	2.40
0.1	**21.68**	**0.768**	2.13
0.2	21.48	0.765	1.90
0.4	21.51	0.761	**1.71**

or downstream performance, supporting their current roles in **`StreamRefine`**. Full continuation remains competitive with the default setting, suggesting that occasional continuation captures most of the benefit without requiring every trajectory to reach the maximum horizon. Removing the compute cost substantially increases the average refinement steps with little consistent quality gain, while the *fixed-horizon* variant performs worse than adaptive stopping.

Sensitivity to λ_a and λ_c. Tables 15 and 16 examine the sensitivity to the anatomy and computation weights. For λ_a, moderate values preserve better translation and downstream performance, while larger values lead to earlier stopping and reduced quality. Increasing λ_c consistently reduces the average refinement steps, with overly large values causing a clear performance drop. Overall, $\lambda_a = 1.0$ and $\lambda_c = 0.1$ provide a favorable balance between quality, anatomy-related performance, and refinement efficiency.

Sensitivity to ϵ. Table 17 studies the continuation probability ϵ on BraTS T2w→T2f. Increasing ϵ from 0.05 to 0.15 improves both PSNR and Dice, after which performance largely saturates. Higher continuation rates provide little consistent quality gain while increasing training cost. We therefore use $\epsilon = 0.15$ as a practical balance between performance and training efficiency.

Sensitivity to $K_{\max}$. Table 18 compares maximum refinement horizons of 4, 6, and 8 on two BraTS translation tasks. Longer horizons yield modest quality gains, but increasing $K_{\max}$ from 6 to 8 provides no consistent improvement across PSNR and Dice. At $K_{\max} = 8$, 82.7% of T2w→T2f cases and 74.0% of T1n→T1c cases still stop within four steps, with overall average depths of 2.67 and 3.46, respectively. These results suggest that longer horizons are used selectively, while $K_{\max} = 4$ remains a practical default that balances quality and refinement cost.

E Additional Analysis

E.1 Stopping and Refinement Analysis

Refinement-depth heterogeneity. For Fig. 1(b), we use the trained *Fixed-K Refine* model and force every held-out case to complete all $K_{\max} = 4$ refinement states. For each case i, we define its oracle refinement depth as the earliest state whose PSNR is within $\delta = 0.05$ of the best PSNR along the

Table 17: **Sensitivity to the continuation probability** ϵ on BraTS T2w→T2f. Avg. Steps denotes the average refinement steps at inference.

ϵ	BraTS T2w→T2f		
	PSNR↑	Dice↑	Avg. Steps↓
0.05	21.48	0.757	**1.99**
0.10	21.60	0.764	2.08
0.15	**21.68**	**0.768**	2.13
0.20	21.66	**0.768**	2.18
0.30	21.61	0.765	2.15

Table 18: **Sensitivity to the maximum refinement horizon on BraTS 2024.** We compare **`StreamRefine`** with different maximum horizons. Avg. Steps denotes the average inference refinement depth. Stop ≤ 4 and Stop < 6 report the numbers and percentages of cases stopping within four steps and before the sixth step, respectively, among 254 validation subjects.

$K_{\max}$	BraTS T2w→T2f					BraTS T1n→T1c				
	PSNR↑	Dice↑	Avg. Steps↓	Stop ≤ 4 n (%)	Stop < 6 n (%)	PSNR↑	Dice↑	Avg. Steps↓	Stop ≤ 4 n (%)	Stop < 6 n (%)
4	21.68	0.768	**2.13**	254 (100.0)	254 (100.0)	22.61	0.807	**2.76**	254 (100.0)	254 (100.0)
6	**21.72**	**0.770**	2.51	217 (85.4)	235 (92.5)	22.67	**0.810**	3.29	197 (77.6)	230 (90.6)
8	21.70	**0.770**	2.67	210 (82.7)	229 (90.2)	**22.69**	0.808	3.46	188 (74.0)	223 (87.8)

completed trajectory,

$$k_i^* = \min\left\{k : \mathrm{PSNR}_{i,k} \geq \max_{1\leq j\leq K_{\max}} \mathrm{PSNR}_{i,j} - \delta\right\}. \tag{22}$$

This tolerance avoids assigning a later step because of negligible PSNR fluctuations. We report the fraction of cases assigned to each refinement depth separately for every translation task.

Later-refinement learning. For Fig. 1(c), we compare *Naive Stop* (Sec. 4.3) and **`StreamRefine`** under matched total training FLOPs, including refinement, continuation, benefit-target construction, and stopping-head computation. Within each seed, both methods share the same initialization, backbone, and optimizer settings. At evaluation, stopping is disabled, and both models complete all four refinement states on the same held-out subjects. We report the mean and standard deviation across three training seeds. We additionally report the empirical execution frequencies and direct-supervision counts for Steps 3 and 4. An equal-data-exposure comparison is provided in Sec. 4.4.

Training reach under matched compute. To examine whether the later-step gains in Fig. 1(c) arise from additional total computation, we compare *Naive Stop* and **`StreamRefine`** under matched training FLOPs.

For this analysis, we additionally log the empirical execution frequency of each refinement state,

$$r_k^{\mathrm{exec}} = \frac{1}{N}\sum_{i=1}^{N} R_{i,k}, \tag{23}$$

where $R_{i,k} = 1$ if trajectory i executes state k. We also report the number of executed states that receive direct refiner supervision during optimization. Both statistics are measured before the inverse-reach reweighting described in Appendix B.4.

Table 19 shows that *Continue-to-Check* increases the exposure and direct supervision of later refinement states under the same total training budget, particularly for Steps 3 and 4. This provides a training-side explanation for the stronger later refinements observed in Fig. 1(c). *Head-only* increases the exposure of later states but provides only limited additional direct refiner supervision, whereas *Refiner-only* substantially increases later-step supervision. *Joint* combines both effects, providing a training-side explanation for the complementary gains observed in Sec. 4.4.

Hidden continuation benefit. For Fig. 1(d), we use the final *Naive Stop* checkpoints from Sec. 4.3. For *image-only stopping*, both the proposed stop and *hindsight benefit* are computed using the *image-only* utility; for *anatomy-aware stopping*, we use the corresponding *anatomy-aware* setting. For

Table 19: **Training reach under matched total compute.** We compare *Naive Stop*, *Head-only*, *Refiner-only*, and **`StreamRefine`** under the same total training FLOPs. r_k^{exec} denotes the fraction of training trajectories that actually execute Step k. Direct Sup. counts the corresponding states receiving direct refiner supervision before inverse-reach weighting. Values are averaged over three training seeds.

Method	Train FLOPs	Updates	Executed Reach (%) $\mathbf{r_3^{exec}}$	Executed Reach (%) $\mathbf{r_4^{exec}}$	Direct Sup. ($\times 10^3$) Step 3	Direct Sup. ($\times 10^3$) Step 4
BraTS T2w→T2f						
Naive Stop	1.00×	112k	31.2 ± 1.0	15.6 ± 0.7	210 ± 7	105 ± 5
Head-only	1.00×	106k	43.4 ± 1.2	30.0 ± 0.9	213 ± 7	112 ± 5
Refiner-only	1.00×	101k	40.8 ± 1.1	27.1 ± 0.8	247 ± 7	164 ± 5
`StreamRefine`	1.00×	100k	**41.6** ± 1.1	**28.2** ± 0.9	**250** ± 7	**169** ± 5
AutoPET CT→PET						
Naive Stop	1.00×	110k	35.8 ± 1.2	19.9 ± 0.9	236 ± 8	131 ± 6
Head-only	1.00×	104k	47.5 ± 1.3	34.1 ± 1.0	239 ± 8	140 ± 6
Refiner-only	1.00×	101k	44.6 ± 1.2	30.8 ± 0.9	270 ± 8	187 ± 6
`StreamRefine`	1.00×	100k	**45.4** ± 1.3	**31.9** ± 1.0	**272** ± 8	**191** ± 6

Table 20: **Independent validation of the PIR structural proxy** on BraTS T2w→T2f. We report Spearman correlations between PIR error and independent segmentation-based structural errors, both across refinement states and for adjacent-step changes. Positive correlations indicate that larger PIR error is associated with greater structural degradation.

Independent structural measure	State-level ρ	Step-change ρ
Tumor Dice error $(1 - \text{Dice})$	0.63	0.51
Boundary error (HD95)	0.68	0.54

each held-out case, we record the first proposed early stop $\hat{\tau} < K_{\max}$ and then complete the same trajectory offline to $K_{\max}$ without updating the model. We report the fraction of proposed early stops for which $B^*_{\hat{\tau}} > 0$.

E.2 PIR Agreement with Independent Structural Measures

We further examine whether PIR reflects structural changes independently of its use in the stopping target. On BraTS T2w→T2f, we complete all refinement trajectories to $K_{\max}$ and compare PIR error with tumor Dice error and boundary error (HD95). As reported in Table 20, PIR shows consistent positive correlations with both measures at the state level ($\rho = 0.63/0.68$) and for adjacent-step changes ($\rho = 0.51/0.54$). The consistent agreement with both measures supports PIR as a proxy for anatomy-related structural change during refinement.

E.3 Later-Refinement Analysis Protocol

We provide additional details of the later-refinement analysis in Sec. 4.4, including the compared variants and evaluation protocol.

Training variants. We use a shared *full-horizon* warm start and keep the same backbone, optimization settings, and stopping formulation across all variants.

- ***Naive*** follows the proposed stop without continuation.
- ***Head-only*** uses checked trajectories only for additional benefit-head supervision, whereas ***Refiner-only*** uses them only to train later refinements.
- ***Joint*** (**`StreamRefine`**) uses both signals.
- ***Full-horizon*** runs every training trajectory to $K_{\max}$ and serves as a reference.

Fixed-depth evaluation. All analyses are conducted on BraTS T2w→T2f. At evaluation, stopping is disabled and all variants are forced to the same refinement depths on the same held-out cases. For the shared-prefix analysis, we fix the first two refinement states from *Naive* or *Joint* and compare both refiners on the remaining steps, with final quality evaluated at $k = 4$.

Table 21: **Hidden-positive stop rate on BraTS T2w→T2f.** We report the fraction of proposed early stops with positive hindsight benefit after offline continuation. Lower is better.

Method	Image-only ↓	Anatomy-aware ↓
Naive Stop	43.6%	36.4%
StreamRefine	23.9%	18.9%

Training-progress analysis. Starting from the shared warm start, we periodically evaluate forced $K_{\max}$-step quality throughout training. For Fig. 5(c), cumulative later-step refiner compute counts the actually executed Step-3 and Step-4 states up to each checkpoint and normalizes them by the corresponding *full-horizon* budget. The execution frequencies in Table 19 are averaged over the complete training run. Unless otherwise stated, all variants use the same evaluation subjects, preprocessing, refinement horizon, and generation settings.

E.4 Stopping Decisions and Hidden Continuation Benefit

We provide additional details for the stopping analysis in Sec. 4.5. All experiments are conducted on BraTS T2w→T2f using the same held-out subjects as the main evaluation.

Completed trajectories. We freeze the trained refiner and complete every held-out trajectory to $K_{\max} = 4$ offline. All stopping rules are then applied to the same volume-level states, so differences reflect stopping decisions rather than different refinement trajectories. Offline completion is used only for analysis; normal inference still stops when $\widehat{B}_k^{\mathrm{vol}} \le 0$.

Hidden continuation benefit. We further quantify how often a proposed early stop still leaves useful refinement unrealized. For each method, we record its first proposed stop $\hat{\tau} < K_{\max}$ and complete the same trajectory offline to $K_{\max}$ without updating the model. We define the hidden-positive stop rate as

$$r_{\mathrm{hp}} = \Pr(B^*_{\hat{\tau}} > 0 \mid \hat{\tau} < K_{\max}), \tag{24}$$

i.e., the fraction of proposed early stops for which further refinement still has positive *hindsight benefit*. For Fig. 1(d), we report this quantity for the *Naive Stop* checkpoints. We additionally compare *Naive Stop* and **StreamRefine** to measure how much learned stopping reduces such premature decisions. As reported in Table 21, compared with *Naive Stop*, **StreamRefine** substantially reduces the hidden-positive stop rate under both utilities, from 43.6% to 23.9% for *image-only stopping* and from 36.4% to 18.9% for *anatomy-aware stopping*. This indicates that learning from continued trajectories reduces premature stops where useful refinement still remains.

Benefit prediction. For Fig. 6(a), we compare the predicted volume-level benefit $\widehat{B}_k^{\mathrm{vol}}$ with the *hindsight benefit* $B_k^{*,\mathrm{vol}}$ for all non-terminal states $k = 1, 2, 3$. Each point corresponds to one volume–step pair from a completed trajectory.

Stopping regret. For Fig. 6(b), all stopping rules are evaluated on the same completed trajectories using the same image, anatomy, and computation objective. Let j^* denote the earliest hindsight-optimal state and $\hat{\tau}$ the selected stopping step. We compute stopping regret as the extra objective cost at $\hat{\tau}$ relative to j^*, and separate it into early stopping ($\hat{\tau} < j^*$) and late continuation ($\hat{\tau} > j^*$). We compare a fixed stop at $K = 2$, matched-random stopping, the *image-only head*, the *anatomy-aware head*, a *full-label head*, and the *hindsight oracle*. The *full-label head* uses the same frozen refiner and head architecture, but is trained with complete benefit labels obtained by running every training trajectory to $K_{\max}$. It is evaluated on the same completed held-out trajectories and serves as a reference for complete stopping supervision. *Matched-random stopping* randomly permutes the learned stopping steps across subjects, preserving the stopping-step distribution and thus the same average computation.

Anatomy-aware stopping. For Fig. 6(c), we first compare the stopping steps selected by the *image-only* and *anatomy-aware* heads over all 254 subjects. They differ for 56 cases. We then evaluate downstream Dice, PSNR, and average refinement steps only on this different-stop subset, using the same downstream evaluation protocol as in Sec. 4.2. For the qualitative examples in Fig. 7, states beyond the selected stop are generated offline only to inspect the consequence of further refinement.

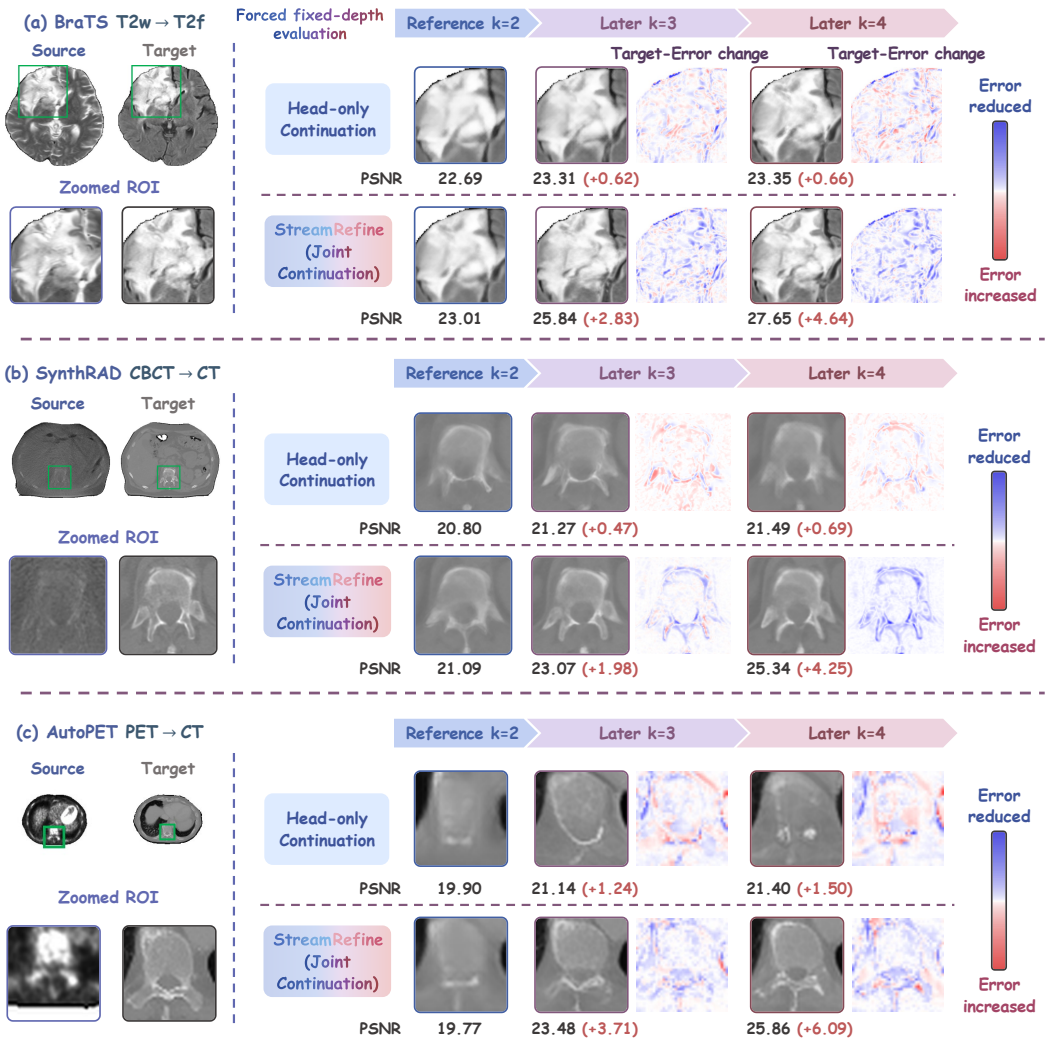


Figure 9: **Qualitative analysis of later-refinement learning across datasets.** With stopping disabled, *Head-only continuation* yields limited later-step corrections, whereas **`StreamRefine`** produces larger PSNR gains and stronger target-error reduction. Target-error change is measured relative to the $k = 2$ reference state, with blue indicating reduced error and red indicating increased error.

E.5 Qualitative Analysis of Later-Refinement Learning

Across examples from all three datasets in Fig. 9, **`StreamRefine`** makes more useful later-step corrections than *Head-only* continuation, with larger PSNR gains and stronger target-error reduction. This qualitatively supports the role of continuation in training otherwise skipped later refinements.

E.6 Computational Efficiency Analysis

Measurement Setup. We evaluate computational efficiency on BraTS 2024 using a single NVIDIA RTX 5090 GPU with the same precision and implementation settings as the main experiments. Training time is averaged over 200 updates after 20 warm-up iterations, and GPU hours are estimated over 100k updates. Forward FLOPs measure the computation of a fixed model forward and are therefore unchanged by ϵ; differences in executed continuation states are reflected by wall-clock time.

Wall-clock Cost of Training. Table 22 shows that occasional continuation adds only modest overhead over training without continuation, increasing time per update from 1.04 to 1.15 seconds. In contrast, full continuation requires 1.76 seconds per update. **`StreamRefine`** therefore uses only $0.65\times$ the

Table 22: **Training efficiency on BraTS 2024.** We report forward FLOPs, per-update training time, and estimated GPU hours. Time/Update is averaged over 200 updates after 20 warm-up iterations, with Relative Time normalized to full continuation and GPU hours estimated over 100k updates.

Training Setting	Per-forward FLOPs (G)↓	Time/Update (s)↓	Relative Time↓	GPU h ↓
No continuation ($\epsilon = 0$)	1734.8	**1.04**	**0.59×**	**28.89**
Full continuation ($\epsilon = 1$)	1734.8	1.76	1.00×	48.89
StreamRefine ($\epsilon = 0.15$)	1734.8	1.15	0.65×	31.94

Table 23: **Whole-volume inference efficiency on BraTS 2024.** We report inference cost and performance on T2w→T2f.

	Inference Cost			BraTS T2w→T2f	
Method	Avg. Refine Steps↓	Latency (s/vol)↓	Peak VRAM (GB)↓	PSNR↑	Dice↑
Fixed-K Refine ($K = 4$)	4.00	31.2	12.7	21.41	0.763
Post-hoc Stop	2.50	24.1	12.7	21.52	0.760
StreamRefine	**2.13**	**20.8**	12.7	**21.68**	**0.768**

wall-clock cost of full continuation, reducing the estimated 100k-update training cost from 48.89 to 31.94 GPU hours.

Inference Cost. Table 23 further demonstrates the practical inference savings from adaptive stopping. **StreamRefine** reduces the average refinement depth from 4.00 to 2.13 steps and latency from 31.2 to 20.8 seconds compared with *Fixed-K Refine*, while improving PSNR and downstream Dice. Peak GPU memory remains unchanged at 12.7 GB across the three settings.

F Discussion and Limitations

StreamRefine learns case-dependent stopping for iterative 3D medical image translation by occasionally continuing beyond proposed stops, allowing the same trajectories to improve later refinements and supervise stopping decisions. While our experiments demonstrate consistent quality–efficiency gains, several limitations remain.

Experimental scale. All experiments are conducted on a single GPU with moderate-sized models. Although this setting allows us to evaluate **StreamRefine** across multiple datasets, tasks, and analysis protocols under practical compute budgets, we have not yet studied its behavior with substantially larger backbones, longer training schedules, or multi-GPU training. Evaluating whether the benefits of adaptive stopping and *Continue-to-Check* persist at larger scales is an important direction for future work.

Maximum refinement horizon. We use $K_{\max} = 4$ in the main experiments. Extending the horizon to 6 and 8 on two BraTS translation tasks yields only modest and task-dependent performance changes. At $K_{\max} = 8$, 82.7% of T2w→T2f cases and 74.0% of T1n→T1c cases still stop within four steps, suggesting that $K_{\max} = 4$ is a practical default but remains restrictive for a subset of cases (Appendix D.5). Longer or dynamically expandable horizons may therefore be useful for tasks with greater refinement needs.

Generalization of the PIR structural proxy. PIR is a lightweight proxy used only to construct the *anatomy-aware stopping* target. On BraTS T2w→T2f, it shows consistent agreement with independent segmentation-based structural measures, but this validation is limited to one dataset and translation direction. The current formulation also requires paired source–target data to compute L_k and PIR-based stopping targets, limiting direct use in unpaired settings. Future work will evaluate PIR across broader datasets, modalities, anatomical regions, and structural measures.

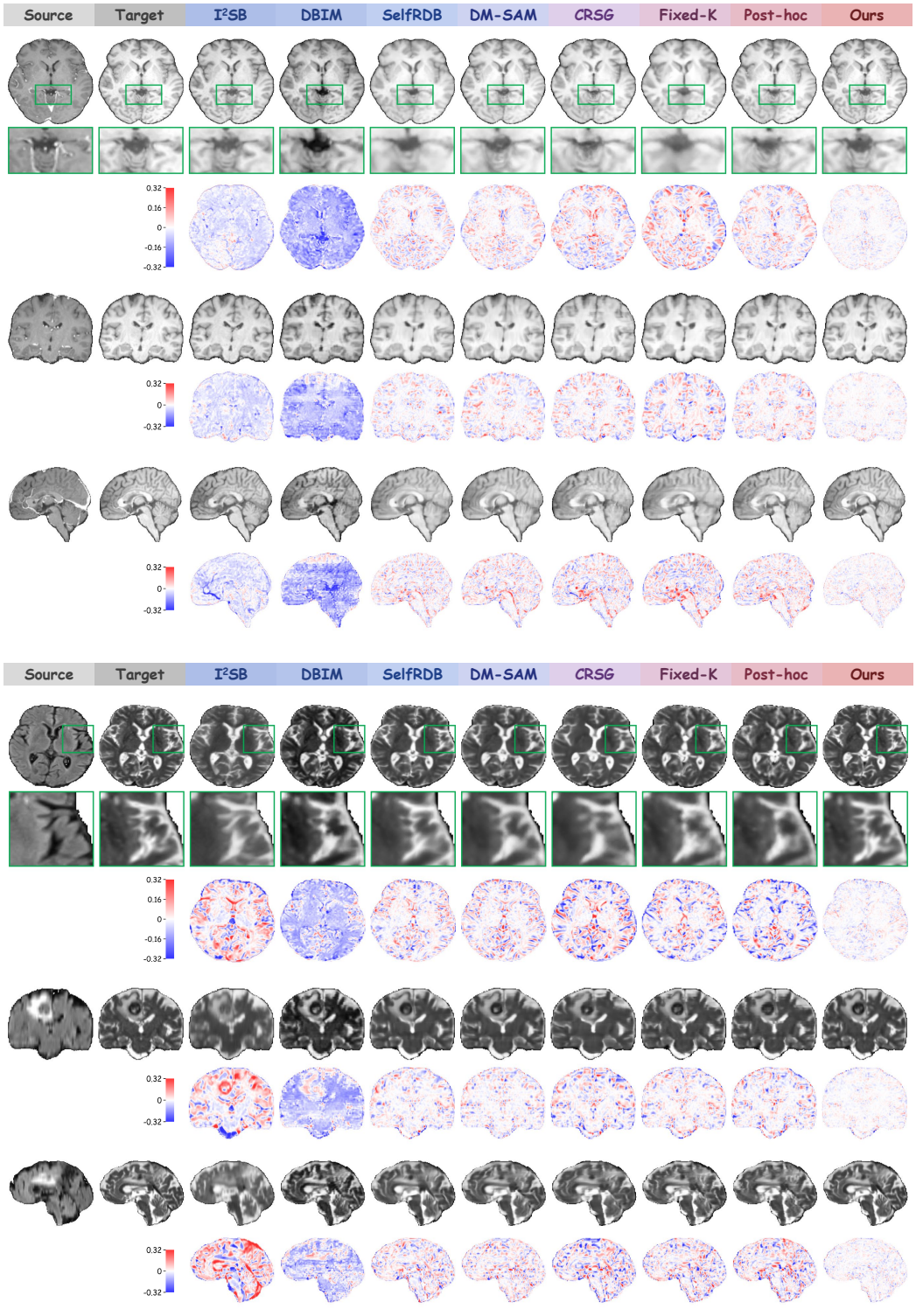


Figure 10: **Additional qualitative comparisons** on BraTS 2024 T1c→T1n (upper) and T2f→T2w (lower), showing axial, coronal, and sagittal views from top to bottom with corresponding error maps.

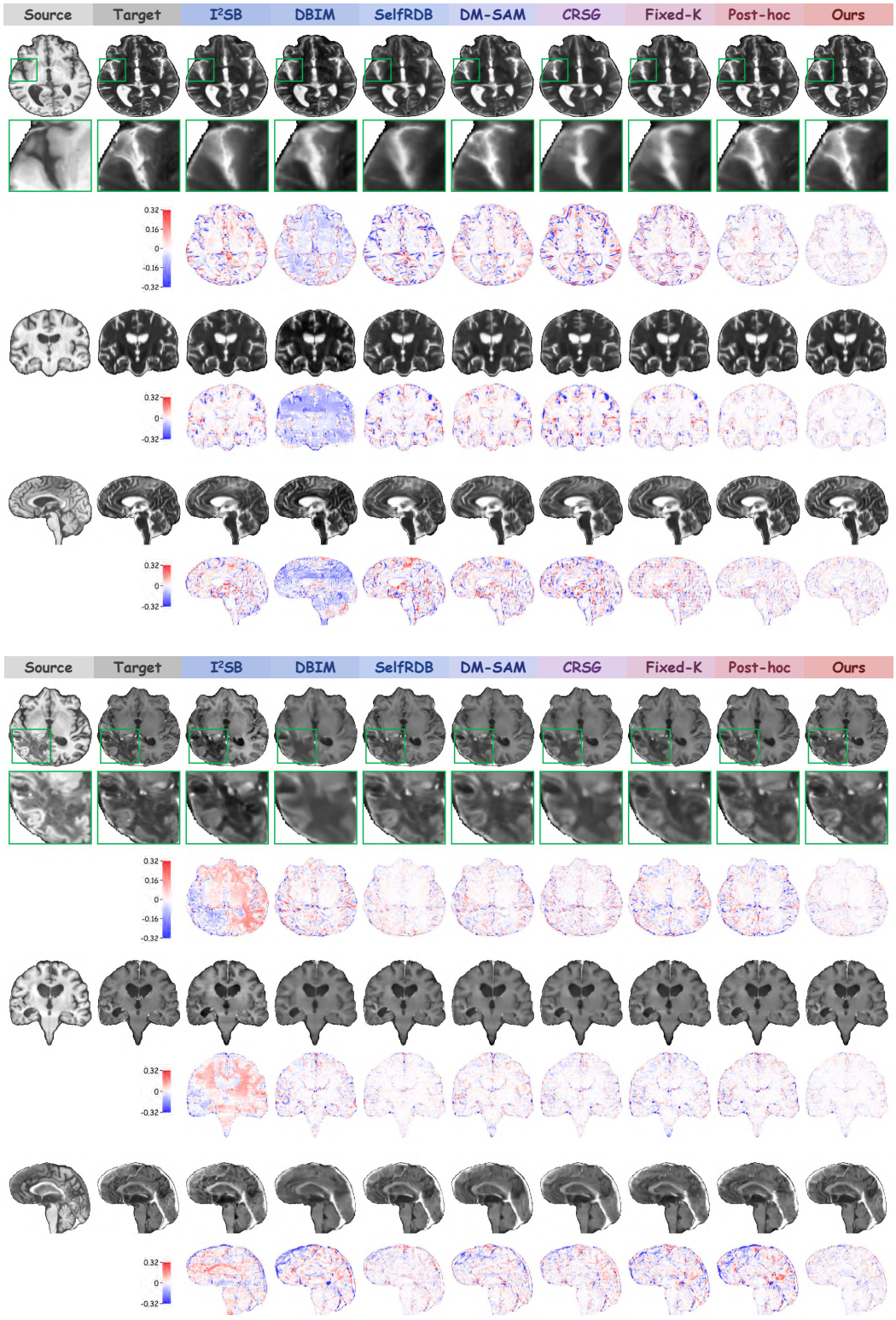


Figure 11: **Additional qualitative comparisons** on BraTS 2024 T1n→T2w (upper) and T1n→T1c (lower), showing axial, coronal, and sagittal views from top to bottom with corresponding error maps.

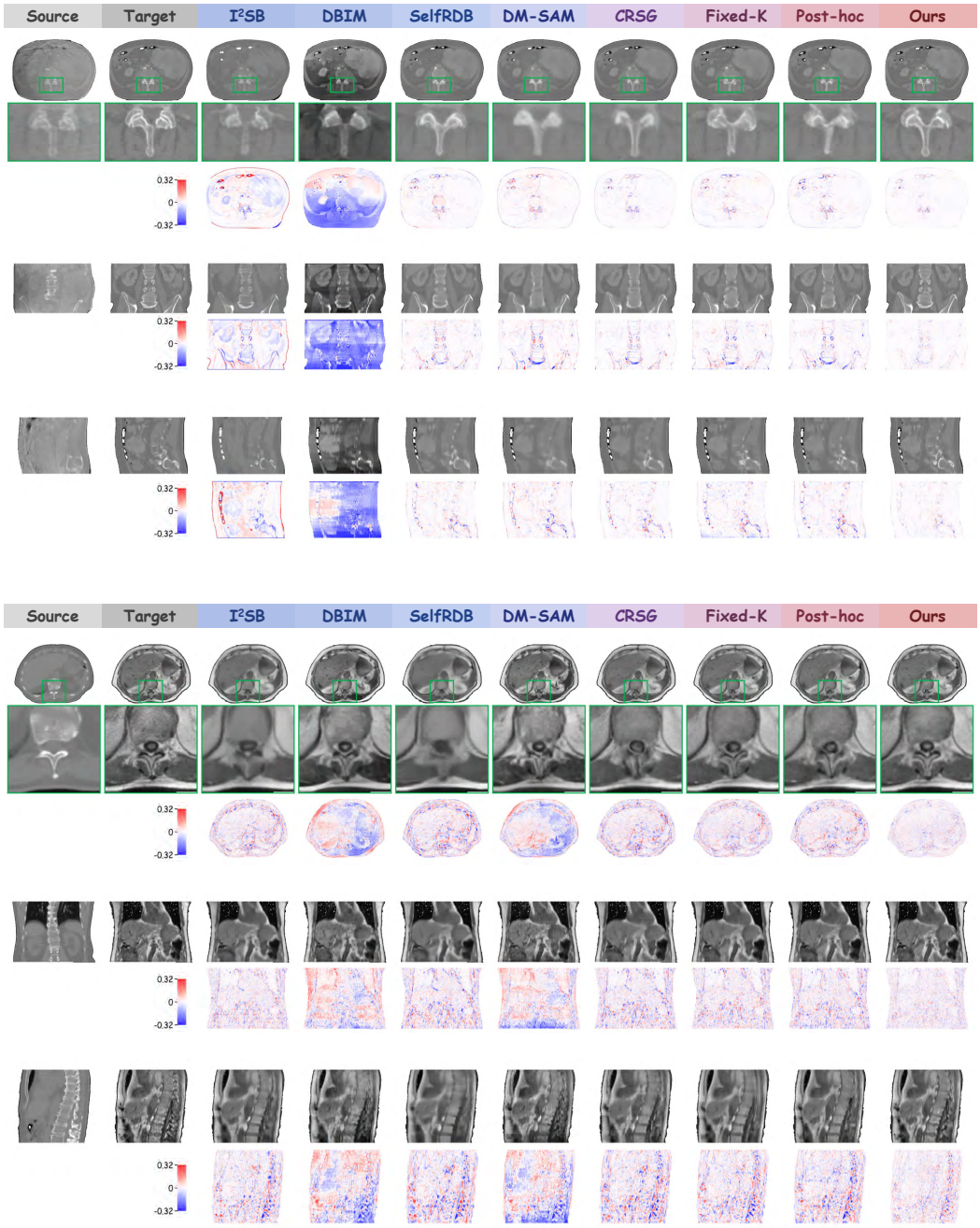


Figure 12: **Additional qualitative comparisons** on SynthRAD 2025 CBCT→CT (upper) and CT→MRI (lower), showing axial, coronal, and sagittal views from top to bottom with corresponding error maps.

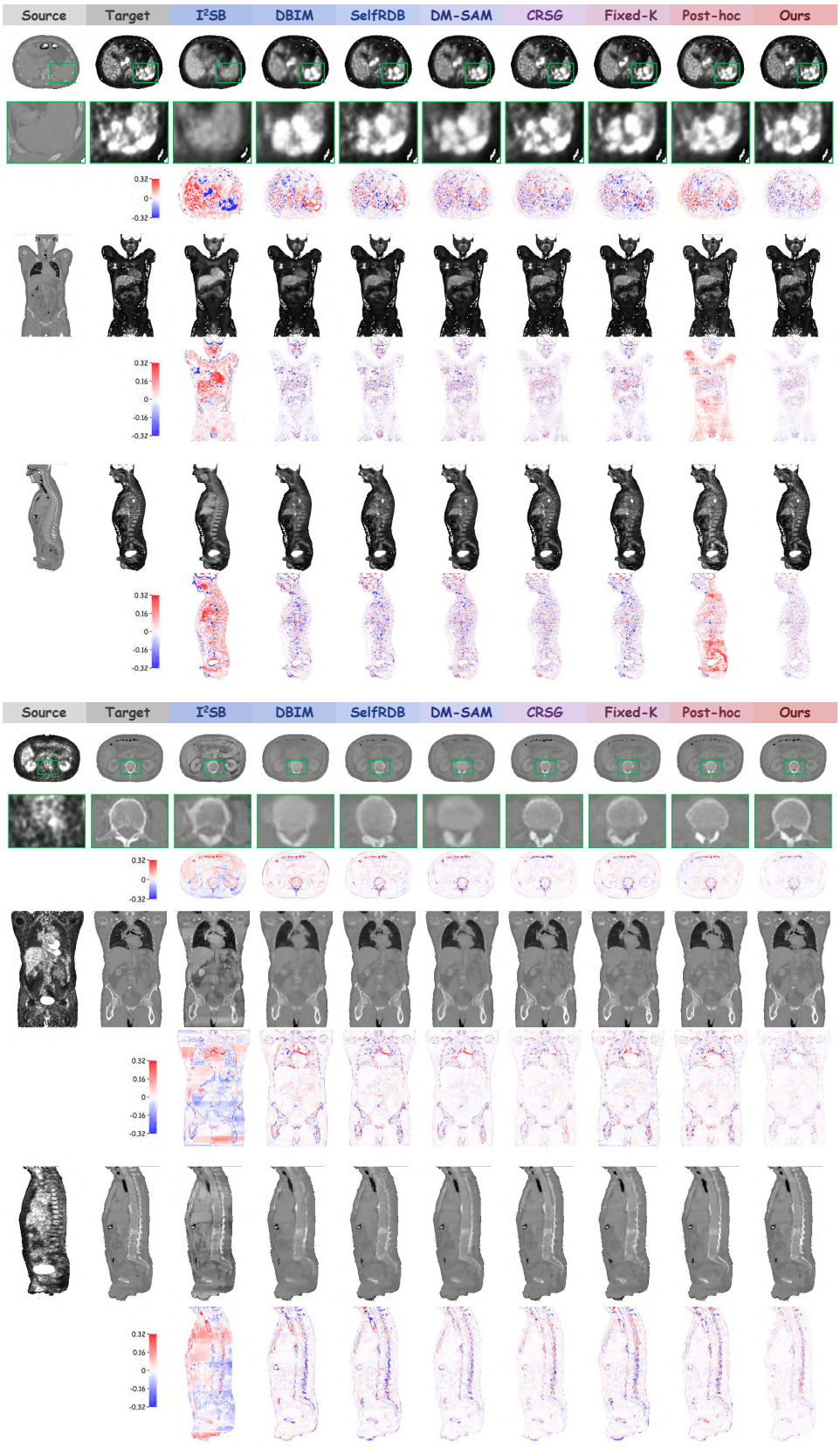


Figure 13: **Additional qualitative comparisons** on AutoPET CT→PET (upper) and PET→CT (lower), showing axial, coronal, and sagittal views from top to bottom with corresponding error maps.