Live Interactive Training for Video Segmentation

Supplementary Material

A. Implementation Details

Datasets We evaluate our method on four challenging VOS benchmarks:

VOST [45] evaluates robustness under object breaking and large transformations, featuring frequent splits, deformations, and small fragments that challenge mask propagation. Results are reported on the validation split.

LVOSv2 [19] is a long-term VOS benchmark emphasizing persistent tracking through occlusions and visually similar distractors, with many videos spanning hundreds of frames. We report results on the validation set.

MOSEv2 [11] extends *MOSEv1* [10] to more complex real-world scenes with frequent object disappearance, occlusion, crowding, and adverse conditions. Since MOSE lacks public validation masks and MOSEv1 was already used to train SAM2, we evaluate on a filtered subset of *MOSEv2* training samples that do not overlap with MOSEv1.

SA-V [37] is a large-scale dataset from SAM2, covering diverse real-world scenes and object types to test model generalization. We evaluate on its validation and test splits.

LIT-LoRA on VOS Throughout the experiments on VOS, we use the SAM2.1-Large checkpoint released on 2024-09-30. In our setup, the SAM2 backbone θ is kept fixed, and a lightweight LoRA module is inserted into the mask decoder. The decoder is composed of stacked two-way transformer blocks, and we augment each attention layer by modifying the query, key, and value projections (W_Q, W_K, W_V) with a low-rank residual update

$$W = W_0 + \Delta W, \qquad \Delta W = BA,$$

where $A \in \mathbb{R}^{r\times d}$ and $B \in \mathbb{R}^{d\times r}$ are the trainable LoRA matrices. The IoU prediction head and object-existence head remain frozen during training.

For inference, SAM2 produces several mask candidates, and we follow the default configuration and select the mask associated with the first mask token as the final prediction. Online updates are driven by user-provided corrections. Each correction minimizes a composite loss

$$\mathcal{L} = \lambda_{\text{focal}}\,\mathcal{L}_{\text{focal}} + \lambda_{\text{dice}}\,\mathcal{L}_{\text{dice}},$$

with a weighting ratio of $\lambda_{\text{focal}} : \lambda_{\text{dice}} = 20 : 1$ following SAM2's default parameters. This design enables the decoder-side LoRA parameters ΔW to adapt online while the remainder of SAM2 stays frozen.

The LoRA for VOS is configured with a rank of 4, $\alpha = 4$, dropout of 0.1, and a learning rate of 1×10^{-4}.

LIT-LoRA on image classification In the experiments on image classification, we use CLIP ViT-B/32. The full CLIP backbone θ is frozen, and a lightweight LoRA module is inserted in the feature space. We train the image encoder LoRA parameters while keeping the text encoder frozen.

At inference time, CLIP assigns each image a class based on similarity to all text prompts, where we use the text prompt "a photo of x". Whenever the ground-truth label does not appear in the top-k predictions (e.g. $k = 3$), a user correction is triggered and used to update the LoRA module. Each correction optimizes a cross-entropy loss over all class prompts, supplemented by a margin-based separation term and L_2 regularization on ΔW.

$$\mathcal{L} = \mathcal{L}_{\text{CE}} \;+\; \lambda_{\text{margin}}\,\mathcal{L}_{\text{margin}} \;+\; \lambda_2\,\|\Delta W\|_2^2.$$

The margin loss is defined as

$$\mathcal{L}_{\text{margin}} = \max\big(0,\; m - (s_y - s_{\max})\big),$$

where m specifies the required separation between the ground-truth logit s_y and the highest incorrect logit $s_{\max}$. We set $m = 10.0$, $\lambda_{\text{margin}} = 0.1$, $\lambda_2 = 1 \times 10^{-4}$.

The LoRA in image classification is configured with a rank of 8, $\alpha = 16$, dropout of 0.1, and a learning rate of 1×10^{-4}. We use a higher LoRA rank for image classification because the adapter operates only on a small feature-space projection in CLIP. This provides additional expressive capacity for fine-grained corrections at negligible computational cost, while preserving fast online updates.

B. Additional Experiments and Analysis

Robustness under different adapter insertion locations In our main experiment on VOS, we apply LoRA to the mask decoder, although other components also contribute to the final mask prediction. To assess the effectiveness of alternative lightweight modules, we additionally evaluate the use of an adapter on the memory bank. We freeze the mask decoder and insert a lightweight residual adapter module into the memory features. The adapter consists of two 1×1 convolutions with a ReLU in between, reducing the feature dimension by half and then restoring it, followed by a residual connection that adds the transformed features back to the input.

We compare the reduction in user corrections achieved by inserting the adapter at different locations within the SAM2 model on the VOST dataset. As shown in Table 3, both insertion choices lead to consistent reductions in user corrections, with only minor differences in effectiveness. However, placing the adapter module in the memory

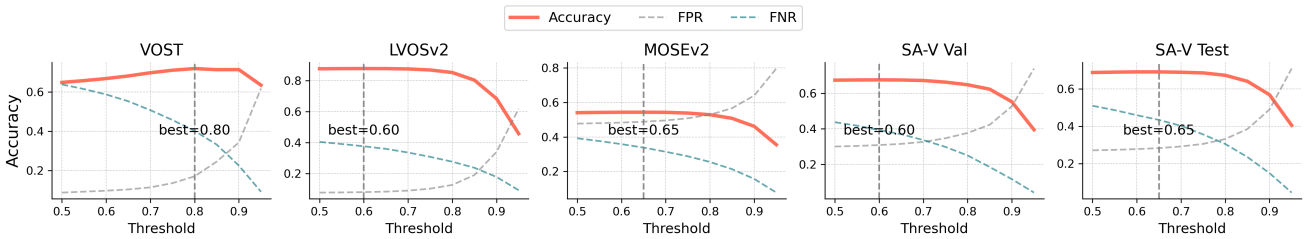

Figure 7. **Predicted IoU vs. Ground-Truth IoU (Accuracy, FPR, FNR).** Accuracy, false positive rate (FPR), and false negative rate (FNR) across different predicted IoU thresholds, measuring how well the predicted IoU aligns with the ground-truth IoU for each dataset. Vertical dashed lines mark the threshold that achieves the highest alignment accuracy.

features introduces more trainable parameters, resulting in higher computational overhead. Therefore, we adopt the mask-decoder insertion, which offers strong correction reduction while remaining lightweight.

Table 3. **User-correction reduction with different adapter insertion locations.**

	# Clicks ↓	# Params
Original	27.43	–
LIT$_{\text{mask decoder}}$	18.24	∼35k
LIT$_{\text{memory adapter}}$	18.84	∼66k (∼2×)

Train on more corrections. We compare training with more corrections under two batch size settings (joint updates with $bs = N$ and single-correction updates with $bs = 1$) against training with only one correction (Table 4). Joint updates ($bs = N$) degrade performance, as gradients from different corrections interfere under the limited capacity of LoRA adaptation. In contrast, sequential updates with $bs = 1$ allow the adapter to incorporate each correction signal more effectively, improving performance at the cost of longer training time. Given the favorable trade-off between performance and training time, we adopt training with a single correction in our method.

Table 4. **Train on more corrections.**

	# Clicks ↓	Reduction	Time (s)
Original	27.43	–	–
1/1 (ours)	18.24	33.56%	0.58
3/3	18.74	31.69%	0.61
5/5	19.03	30.62%	0.62
3/1	17.73	35.39%	1.61
5/1	17.64	35.71%	2.32

Exploration of using predicted IoU from SAM2 In our experiment, we assume a user is monitoring the video segmentation process and making decisions about when to apply corrections and whether the LIT-LoRA correction is acceptable. To move toward a more automated pipeline, we investigate whether model-internal signals can replace these human decisions for both error triggering and correction acceptance.

SAM2's mask decoder outputs include MLP heads that predict the IoU score and an occlusion score alongside the predicted masks, which could be a signal of the quality of the predicted mask. During training, SAM2 supervises the predicted IoU using ground truth IoU via an IoU output token. Motivated by this, we explore whether the predicted IoU can serve as a reliable signal for automatically detecting segmentation errors and determining whether a correction should be accepted.

To assess this, we evaluate how well the predicted IoU aligns with the ground truth IoU (set as 0.5) under different thresholds (Figure 7). Specifically, we define a prediction as correct when both the predicted and ground truth IoU values are simultaneously above or below a given threshold. This allows us to calculate an accuracy score for each threshold setting. We additionally report the false positive rate (FPR), where the predicted IoU overestimates mask quality, and the false negative rate (FNR), where it underestimates quality, to more precisely characterize this alignment behavior.

As shown in Figure 7, the predicted IoU score is unreliable in practice. Notably, the optimal threshold for predicted IoU to align with ground truth IoU could differ substantially from the predefined ground truth threshold of 0.5 used in our setup and varies across different datasets. For example, on VOST, the threshold that best aligns predicted IoU with the 0.5 ground-truth IoU is around 0.8, yet the accuracy peaks at only 0.74. Moreover, the accompanying FPR and FNR curves further highlight the instability of predicted IoU as a quality estimator: since predicted IoU scores are often underestimated, FNR is high at low thresholds, meaning many high-quality masks are incorrectly flagged as low quality; as the threshold increases, FPR rises quickly,

causing the model to misclassify many low-quality masks as high quality and miss frames that actually require correction. This imbalance shows that predicted IoU does not provide a stable or trustworthy signal of true segmentation performance and should therefore be used cautiously.

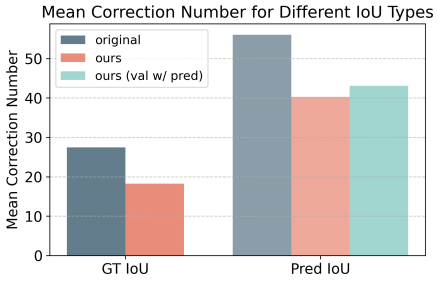

Figure 8. **Using predicted IoU for correction triggering and validation.**

Nevertheless, we explore using the predicted IoU as a signal for both correction triggering and correction validation on VOST dataset. Based on our earlier analysis (Fig. 7), we adopt a threshold of 0.8 for predicted IoU in place of the 0.5 threshold defined for ground truth IoU ($\tau_{\text{IoU}} = 0.5$). We evaluate two configurations: (1) using predicted IoU only to trigger corrections, while still relying on the user (i.e., ground-truth IoU) to validate LIT-LoRA outputs (gray and red bars in Figure 8); and (2) using predicted IoU for both triggering and validation (green bar on the right in Figure 8).

Our results (Figure 8) show that although LIT-LoRA still reduces user effort under the automated settings, correction efficiency declines compared to full user supervision. Notably, the semi-automatic variant, where only triggering is automated using predicted IoU while validation remains user-based, achieves greater user effort reduction, but still lags behind the full user monitoring baseline. These findings highlight that while SAM2's predicted IoU can partially assist interaction, its misalignment with true segmentation quality limits its reliability. Developing a more accurate segmentation quality estimator remains a key avenue for enabling more effective automation.

C. Limitations and Broader Impacts

While our method provides an effective framework for efficient user corrections in interactive visual systems, it has some limitations. First, our system requires user monitoring to detect errors, since SAM2 lacks a robust internal quality estimator. Incorporating automated validation mechanisms, such as learned IoU predictors or uncertainty estimation, could enable the system to automatically identify failure cases and further reduce the need for user intervention. Second, our experiments primarily rely on synthetic user interactions. Although this protocol is standard in interactive segmentation research, real users may exhibit different behaviors and correction strategies. While we conducted a small-scale human evaluation to validate our method, a more systematic and large-scale user study would provide a more comprehensive assessment of real-world practice. Third, our approach leverages LoRA-based live interactive training to achieve low-latency adaptation. This design assumes that the underlying base model already possesses strong generalization ability, enabling it to adapt quickly with lightweight updates. As a result, the effectiveness of our method may depend on large, well-pretrained models such as SAM2 or CLIP. In future work, we hope that advances in foundation models and more efficient adaptation mechanisms will further broaden the applicability of our framework to a wider range of visual systems.

D. Additional Qualitative Results

Successful cases We present additional qualitative examples to highlight both the successful corrections and the broader applicability of LIT-LoRA. Figure 9 shows successful cases on image classification: after learning from user-corrected labels, LIT-LoRA is able to correct similar mistakes that appear in subsequent images within the same group. This demonstrates that LIT-LoRA can effectively capture similar error patterns and correct them in future predictions, extending its usefulness beyond video object segmentation.

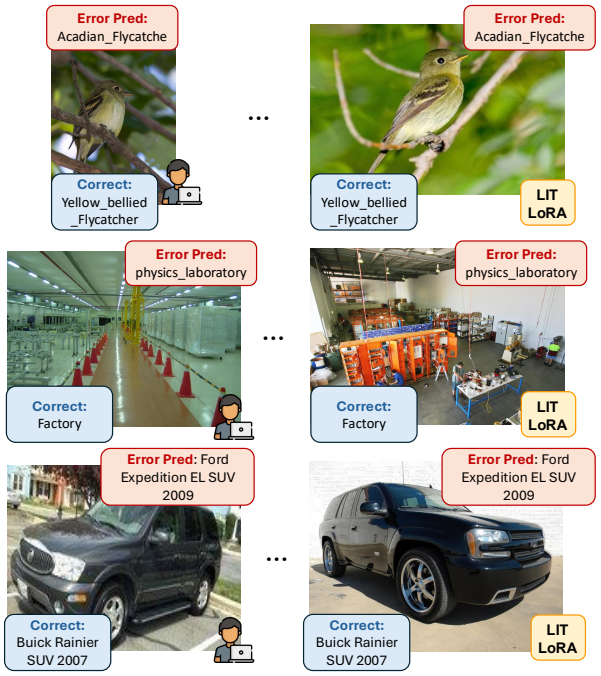

Figure 9. **Qualitative result of successful cases of LIT-LoRA on image classification.**

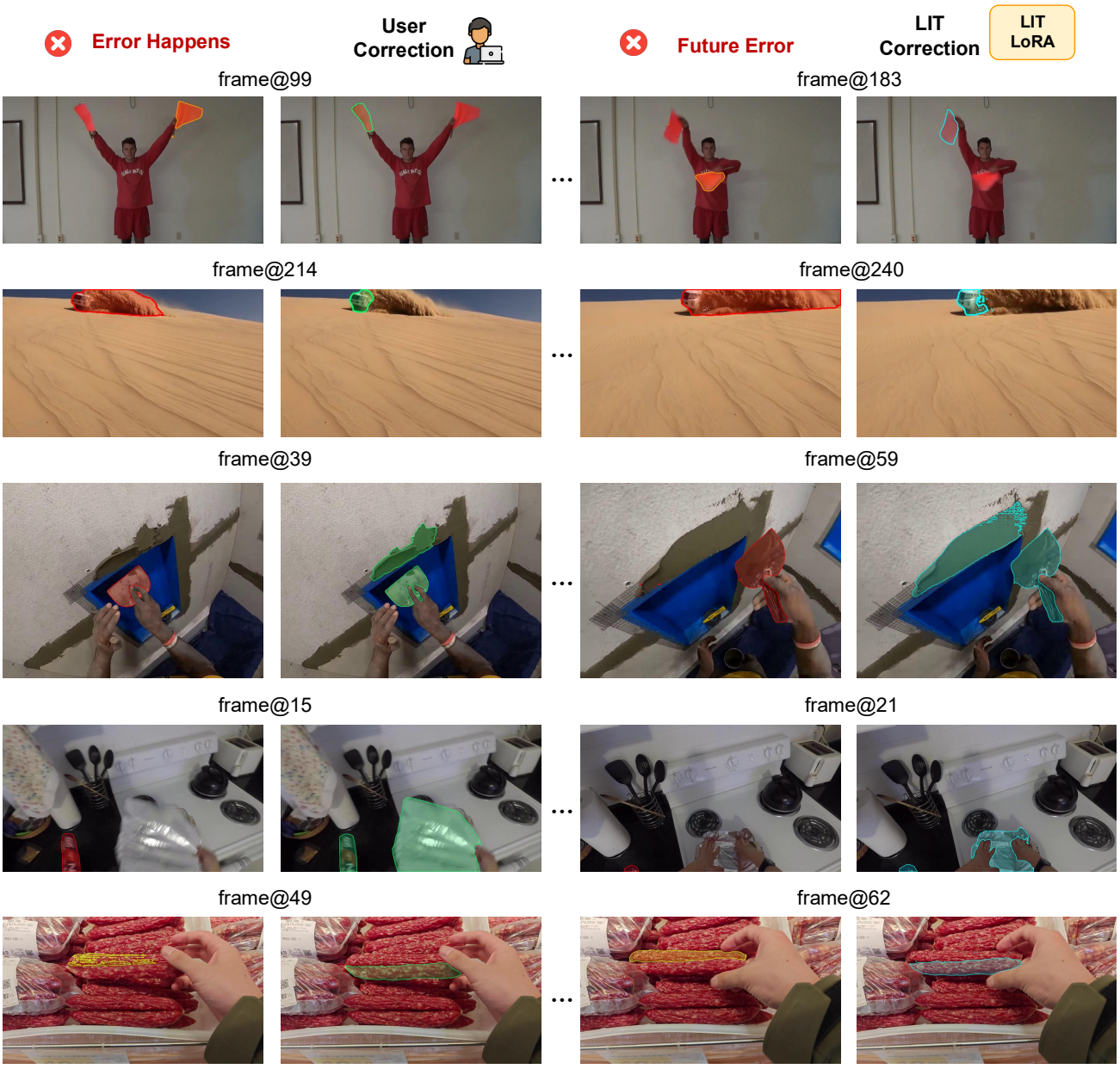

Figure 10. **Additional qualitative result of successful cases of LIT-LoRA on VOS.**

We show additional successful LIT-LoRA adaptations on video object segmentation in Figure 10. These examples illustrate that LIT-LoRA can handle a broad range of challenging failure modes, including object switching, ambiguous boundaries, and complex object separations etc. Notably, it achieves these corrections from learning from a few prior user interactions, demonstrating strong responsiveness and the ability to refine segmentation quality under difficult visual and structural conditions.

Failure cases We also present failure cases in Figure 11, illustrating scenarios where LIT-LoRA is unable to correct the errors. In VOS tasks, the challenges include situations where the object is heavily camouflaged within a cluttered background (case 1), where the target object is extremely small (case 2), and where the object undergoes large or abrupt transformations that invalidate previously learned correction patterns (case 3). Such failures reveal limitations in the mask decoder's generalization capacity and discrepancies between predicted and actual mask quality. Addressing these challenges may require richer training

Figure 11. **Failure cases of LIT-LoRA.**

data or more robust adaptation mechanisms.

In the image classification task, LIT-LoRA can also face challenges. It may fail to correct mistakes when an image appears visually similar to past errors but actually belongs to a different class (case 1), or when the new error differs substantially from what it has previously learned (case 2). These difficulties are inherent to the online streaming setting, where new samples may diverge from earlier corrections. As more user-corrected examples accumulate and the model continues to adapt, LIT-LoRA can gradually become more stable and reliable.

Despite these limitations, LIT-LoRA remains highly effective in practice. It reduces user corrections by 18%–34% on VOS and 35%–43% on image classification, substantially reducing the user's annotation burden. Moreover, since users validate the corrections, occasional LIT-LoRA mistakes do not degrade final performance: incorrect updates can be rejected and corrected by the user, and these user-provided corrections further serve as supervision signals for continuous online learning. Future work can explore strategies to further enhance the effectiveness and reliability of LIT-LoRA.