Live Interactive Training for Video Segmentation

Xinyu Yang Haozheng Yu Yihong Sun Bharath Hariharan Jennifer J. Sun

Cornell University

Abstract

Interactive video segmentation often requires many user interventions for robust performance in challenging scenarios (e.g., occlusions, object separations, camouflage, etc.). Yet, even state-of-the-art models like SAM2 use corrections only for immediate fixes without learning from this feedback, leading to inefficient, repetitive user effort. To address this, we introduce Live Interactive Training (LIT), a novel framework for prompt-based visual systems where models also learn online from human corrections at inference time. Our primary instantiation, LIT-LoRA, implements this by continually updating a lightweight LoRA module on-the-fly. When a user provides a correction, this module is rapidly trained on that feedback, allowing the vision system to improve performance on subsequent frames of the same video. Leveraging the core principles of LIT, our LIT-LoRA implementation achieves an average 18-34% reduction in total corrections on challenging video segmentation benchmarks, with a negligible training overhead of ∼0.5s per correction. We further demonstrate its generality by successfully adapting it to other segmentation models and extending it to CLIP-based fine-grained image classification. Our work highlights the promise of live adaptation to transform interactive tools and significantly reduce redundant human effort in complex visual tasks. Project: https://youngxinyu1802.github.io/projects/LIT/.

1. Introduction

Modern vision foundation models [25, 37] have introduced powerful new capabilities for interacting with users, such as generating video segmentation masks from simple clicks or boxes. However, despite this transformative potential, achieving robust performance in complex real-world scenarios remains challenging. Persistent issues such as occlusions [13, 47], look-alike objects [66], non-rigid deformations [45, 59], and small instances [13, 58, 64] frequently lead to errors that demand substantial human intervention within a single video.

This human-model collaboration, however, faces a critical limitation: even state-of-the-art models like SAM2 primarily leverage user corrections for immediate prediction refinement. Such feedback is often applied temporarily or cached in memory without leading to fundamental model adaptation. Consequently, the system does not truly *learn* or generalize from these valuable interactions to subsequent frames, forcing users into an inefficient and frustrating cycle of repeatedly correcting the same types of errors within

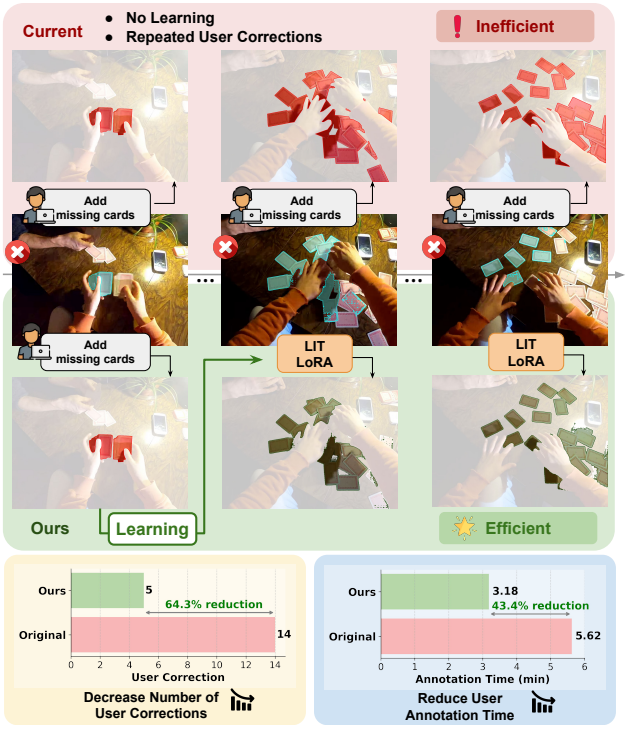


Figure 1. **Comparison between the current non-learning system and our LIT-LoRA approach.** The current system (top) does not learn from user feedback, leading to the same errors to reappear and requiring repeated corrections (e.g., 14 prompts to add the missing cards), which leads to substantial annotation time (e.g., 5.62 mins). In contrast, our LIT-LoRA method continuously adapts to user corrections and generalizes to similar future errors, reducing the number of required corrections (e.g., down to 4) and user annotation time (e.g., down to 3.18 mins).

the same video. This substantially limits overall workflow efficiency and long-term adaptability.

Figure 1 illustrates this: segmenting challenging subjects like the separating cards with SAM2 alone requires numerous manual interventions (e.g., 14 corrections taking more than 5 minutes) due to such recurring failures. Ideally, a system would learn from these initial corrections to autonomously handle subsequent, similar challenges, thereby significantly reducing the corrective burden. Our approach demonstrates this capability, requiring only 4 corrections in the same scenario and reducing the annotation time.

To address the challenge of recurring corrections, there is a need for a framework for interactive vision models to learn and adapt live from user guidance. The rise of powerful vision foundation models and the development of parameter-efficient adaptation techniques make such online learning [31, 38] increasingly practical. We introduce Live Interactive Training (LIT) to realize this vision. LIT operationalizes principles of online learning specifically for the demands of prompt-based interactive visual systems (e.g., those guided by user clicks, boxes, or masks), enabling continuous adaptation from human corrections directly during inference. We initiate LIT's potential within video segmentation, a domain where user corrections are expensive and which demands accurate, adaptable, and efficient outcomes.

To realize the LIT framework, we introduce LIT-LoRA, a lightweight and modular implementation. LIT-LoRA trains compact LoRA modules online, leveraging user corrections in real-time. This approach allows the underlying foundation model to capture, generalize, and dynamically adapt to user-provided feedback. We extensively evaluate LIT-LoRA using SAM2 as a leading video segmentation exemplar. To rigorously benchmark the efficiency, we evaluate our method using a controlled, reproducible protocol with synthetic user corrections, a standard practice in interactive segmentation [6, 9, 25, 37]. To show generality, we further extend LIT to a CLIP-based image classification task, where user feedback is textual, and observe consistent improvements. We lay the foundation for interactive visual systems that are more efficient and adaptive, advancing human-AI collaboration in complex real-world settings.

To summarize, our contributions are:

- We present LIT, a novel framework for prompt-based visual systems that enables models to learn and adapt from human corrections during inference.
- We propose LIT-LoRA, a practical and lightweight implementation of LIT that demonstrates significant reductions in user effort (18-34% fewer corrections in challenging video segmentation) with negligible ($\sim$0.5s) overhead.
- We demonstrate the generality of the LIT framework, showing it is model-agnostic (improving multiple SAM2 variants) and task-agnostic (extending to CLIP-based image classification).

2. Related Work

Video Object Segmentation. The task of video object segmentation (VOS) is to segment an object throughout a video given its first-frame mask. One of the main approaches uses memory-based representations to propagate features across frames [5, 7, 64]. Building on this paradigm, SAM2 [37] extends SAM [25] to videos by incorporating a memory bank of object features. While achieving strong zero-shot generalization across benchmarks, its performance drops in challenging scenarios [3, 8, 19, 43, 45, 59]. To address the limitations, several recent studies have extended SAM2 with additional mechanisms. Many methods focus on enhancing memory design. For example, SAM2Long [13] introduces a tree-based memory retrieval strategy to reduce error propagation; SAMURAI [58] incorporates motion cues for more stable multi-object tracking; and DAM4SAM [47] adopts a distractor-aware memory update to improve robustness among similar objects. Other approaches modify the feature representation of SAM2 by introducing new learnable tokens (e.g. CAMSAM2 [66] and HQ-SAM2 [24]). There are also efforts to adapt SAM2 to specific domains through finetuning [4, 23, 29]. Despite these advances, they primarily focus on model or domain adaptation, and overlook a powerful, inherent capability of SAM2: user correction. Our work investigates a complementary approach: instead of modifying the core model, we leverage its interactive capability to enhance performance, specifically investigating how to integrate user feedback effectively to minimize repetitive user effort.

User-interactive Visual Systems. Human-in-the-loop collaboration enables models to achieve greater performance and usability through interactive support. Systems such as PromptCharm [54], DesignPrompt [34], and MagicQuill [30] explore how generative models can better interpret user sketches and strokes for interactive image editing. In image and video object segmentation, visual prompting similarly leverages user-provided cues (e.g., points, boxes, masks) to guide segmentation [6, 9, 25, 37, 50]. However, existing works focus on improving how models interpret user intent efficiently, rather than enabling models to learn from user interactions to correct future errors.

Parameter-Efficient Fine-Tuning. With the rise of large foundation models [2, 14, 36, 41, 46, 62], full fine-tuning has become prohibitively expensive in memory, storage, and latency. Parameter-Efficient Fine-Tuning (PEFT) mitigates this cost by freezing the backbone and updating only a small set of parameters or lightweight modules, enabling efficient adaptation to downstream tasks. This lightweight structure and minimal trainable footprint make PEFT models well-suited for our online interactive tasks where low

latency and small memory overhead are key requirements. Among the various PEFT methods [12, 16, 21, 22, 32, 33], we adopt LoRA [20] in our framework for its efficiency and ease of use, while our framework is inherently compatible with other PEFT techniques. LoRA injects trainable low-rank matrices into the Transformer layers while keeping the original weights frozen, achieving strong adaptation performance across diverse domains [60, 61, 63, 65]. However, most PEFT methods are trained offline on static, supervised datasets [28, 44, 53]. Even more recent work that explores online adaptation with LoRA focuses on continual learning from a stream of labeled data [18, 55]. Our framework employs LoRA for live, user-driven adaptation, enabling real-time model updates from interactive user feedback while keeping the training lightweight.

Model Adaptation Paradigm. Several model adaptation paradigms have emerged in recent years, such as Test-Time Training (TTT) [15, 17, 42], Continual Test-Time Adaptation (CTTA) [40, 52], and Online Learning [35, 39]. TTT adapts a pre-trained model using unlabeled test samples at inference time via self-supervised objectives, enabling limited domain adaptation but assuming a static test set. OSVOS [49] follows this paradigm by fine-tuning on the first video frame to adapt the VOS model. CTTA extends this idea to streaming inputs, continuously updating models to shifting distributions using pseudo-labels, as demonstrated in VOS by OnAVOS [48]. Online Learning [56] instead incrementally updates models using ground-truth labels as they arrive in the data stream. Our framework, LIT-LoRA, can be viewed as a *user-feedback-driven variant of online learning* that operates at inference time. While it shares CTTA's streaming and deployment-time nature, it differs critically in supervision: adaptation is guided by human corrections rather than pseudo-labels. This interactive signal turns the process into a lightweight, human-in-the-loop continual adaptation scheme, combining the immediacy of online learning with the practicality of inference-time adaptation. Furthermore, unlike continuous adaptation methods for single-image segmentation [1, 26], LIT-LoRA adapts at the level of the data stream (e.g., a video containing multiple frames), enabling the correction to benefit later samples that share similar challenges.

3. Method

We propose Live Interactive Training (LIT), a framework designed to enable lightweight, real-time adaptation for user corrections in interactive visual systems. While the framework is designed to be task- and model-agnostic, we initialize it on SAM2 (Section 3.1), as it serves as a strong, representative system for user-interactive segmentation. In the following, we first present the overall LIT framework (Section 3.2) and then detail its specific implementation, LIT-LoRA (Section 3.3).

3.1. SAM2 Preliminaries

SAM2 [37] is an exemplar model for promptable visual segmentation (PVS) in temporally dynamic settings. In the PVS framework, users can provide prompts, such as positive/negative clicks, bounding boxes, or masks, on the video frame to initialize object segmentation. Once a prompt is received, the model immediately returns a segmentation mask for that frame and then propagates the object representation throughout the video to generate a masklet (i.e. the predicted object mask on every frame).

To maintain interactivity, additional prompts can be provided at any time to refine the video-wide prediction further. Architecturally, SAM2 achieves this interaction by augmenting the standard SAM architecture with a memory bank that stores object features from previous propagation. While this promptable design offers flexibility, the model does not truly learn from these user interactions. Corrections are merely stored as conditional inputs or as recent-frame context in the memory bank of the model. However, since the model's parameters remain frozen, it cannot internalize or generalize from these corrections. As a result, similar errors can persist in future frames, limiting the system's ability to improve over time.

3.2. LIT Framework

We introduce the **LIT (Live Interactive Training)** framework for building interactive visual systems that adapt continuously from user feedback during inference. Unlike conventional approaches that rely on static, pre-trained models or batch fine-tuning, LIT is a *user-feedback-driven variant of online learning* that operates at inference time. It emphasizes real-time and user-driven adaptation, enabling models to improve performance dynamically in response to user feedback.

In the LIT setting, data arrive as a stream $\{x_t\}_{t=1}^{T}$, where each data sample x_t (e.g., an image or a video frame) is processed sequentially within the stream. For stable online adaptation, the stream is conceptually divided into coherent groups, such as individual videos or subsets of samples that share similar visual or semantic characteristics. As each sample is processed, the model produces an initial prediction $\hat{y}_t = f_{\theta,\phi_t}(x_t)$, where θ denotes frozen backbone parameters and ϕ_t denotes the lightweight, trainable adapter active at time t.

When the user identifies an error and provides a correction y_t^*, that correction is immediately treated as a supervision signal. This signal is used to train a lightweight, parameter-efficient module ϕ_t in real-time. The adapter is updated via

$$\phi_{t+1} \leftarrow \phi_t - \eta \nabla_{\phi_t} \mathcal{L}\big(f_{\theta,\phi_t}(x_t),\, y_t^*\big),$$

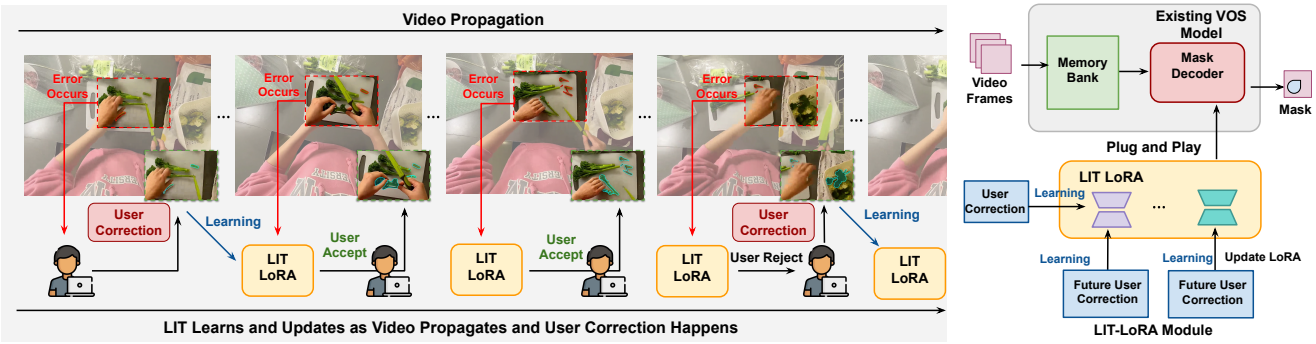


Figure 2. *Left:* **Overview of the LIT-LoRA framework on VOS.** As the video progresses, segmentation errors may arise. When the user provides a correction (which can be time-consuming), the correction is used to train a LoRA module on-the-fly. The LoRA module is then consulted for later errors: if its prediction meets the validation criterion, it is accepted to correct the error; otherwise, the adapter is further refined using the latest correction. *Right:* **LIT LoRA module illustration.**

where $\mathcal{L}$ is a task-specific loss and η is a small learning rate enabling stable real-time updates.

This adapted module is then used to improve predictions for subsequent data in the same stream:

$$y_{t'}^{\phi} = f_{\theta,\phi_{t'}}(x_{t'}), \quad t' > t.$$

The adapter is maintained and incrementally refined within each stream group, enabling it to accumulate group-specific knowledge and correct recurring failure patterns that stem from shared visual or semantic structures. When the system transitions to a new group, the adapter is reinitialized.

This update–predict cycle repeats whenever new corrections arrive, forming a closed-loop system in which the model continually incorporates user guidance. As a result, LIT progressively reduces repeated errors and improves system efficiency during the same inference session, all without retraining the full backbone.

The key features of the LIT framework include:

- **Live training:** The model supports low-latency updates during inference.
- **Interactive learning:** The system is able to respond immediately to user feedback and learn from this interactive result.
- **Online improvement:** The model continuously incorporates feedback to refine predictions over time, improving its ability as the system proceeds.

3.3. LIT-LoRA on VOS

While the LIT framework is designed to be task- and model-agnostic, we instantiate it on the video object segmentation (VOS) task using LoRA (denoted as $\mathcal{A}_t$) as the lightweight learning module (ϕ_t in the LIT framework), forming **LIT-LoRA**. In this setup, the LoRA adapters are attached to a frozen segmentation backbone (θ) and updated online in response to user corrections. This design enables real-time learning with minimal overhead and allows the model to transfer correction patterns to future errors, thereby reducing redundant user effort and improving segmentation quality over time.

As illustrated in Figure 2, LIT-LoRA operates as a live, user-driven adaptation loop. During inference, the model sequentially processes incoming video frames and may produce segmentation errors. When an error is identified, the user provides a correction for that frame, which serves as an immediate supervision signal. The LoRA adapter is then trained on-the-fly to integrate this correction. For future error cases, the updated adapter generates refined predictions; if the predicted mask meets the quality criterion, it is accepted, otherwise the user provides another correction and the adapter is updated again. This continual loop allows the model to adapt dynamically within the same inference session, effectively closing the gap between inference and training. We describe the detailed process below.

Interactive Error Detection and Correction In the LIT framework, the adaptation loop is initiated by an error trigger. In a real-world interactive scenario, this trigger is a manual user correction (e.g., clicks or masks) provided when the user identifies a segmentation failure. This correction, M_t^{corr}, serves as feedback for model adaptation.

Live Model Updating from User Corrections A lightweight LoRA module $\mathcal{A}_t$ is trained whenever a correction is received. It takes the visual embedding of the current frame F_t together with the corresponding memory bank information to obtain fused features x_t, and outputs a refined mask prediction $M_t^{\mathcal{A}} = f_{\theta,\mathcal{A}_t}(x_t)$. The adapter is optimized using a standard segmentation loss

$$\mathcal{L} = \mathcal{L}_{\text{seg}}(M_t^{\mathcal{A}}, M_t^{\text{corr}}).$$

We follow the SAM2 training setup, which combines focal and Dice losses with a weighting ratio of 20:1 to balance pixel-level accuracy and region-level consistency. Since only a small number of low-rank parameters in $\mathcal{A}_t$ are updated, the optimization converges typically under one second and incurs minimal computational or memory overhead. The updated adapter is immediately applied to subsequent errors, enabling the system to incorporate user feedback in real time and progressively refine segmentation quality across the video.

Propagation and Validation of Updates For a future frame $F_{t'}$ where an error occurs, the system employs the updated LoRA adapter $\mathcal{A}_{t'}$ to produce a refined mask prediction $M_{t'}^{\mathcal{A}} = f_{\theta,\mathcal{A}_{t'}}(x_{t'})$. This refined mask is presented to the user for validation:

- If the user accepts this prediction (i.e., by not providing another correction), the prediction is used as the final output for $F_{t'}$. It replaces the previous result and can be stored in the memory bank to enhance future propagation.
- If the user identifies a new failure and provides a new correction $M_{t'}^{\text{corr}}$, this is treated as a new error event.

In the second case, the system flags an error, and the LoRA module $\mathcal{A}_{t'}$ is then incrementally updated using this new correction signal, enabling the model to refine its parameters in real time. This continual loop of user validation and model adaptation allows the system to dynamically adjust to new frames, maintain robust segmentation performance, and minimize the number of user interventions required for high-quality results.

4. Experiments

4.1. Experiment Setup

We evaluate LIT-LoRA under an interactive online evaluation protocol following the default settings of SAM2 [37]. The model processes video frames sequentially in a single forward pass, where each frame is visited once and never revisited, reflecting the real-time nature of interactive usage. For each frame F_t, the model predicts a segmentation mask M_t. The prediction quality is assessed by $\text{IoU}(M_t, M_t^{\text{gt}})$. When this score falls below a predefined threshold τ_{IoU}, an error event is triggered and the system enters a correction phase. Unless otherwise specified, we use $\tau_{\text{IoU}} = 0.5$ as the default error-trigger threshold, corresponding to visibly noticeable failures and aligning with the standard AP@50 evaluation threshold.

Following standard practice in interactive segmentation [6, 9, 25, 37], we employ synthetic user corrections for control and reproducibility. Corrections are simulated as follows: (1) the user first provides up to three clicks at error centers to locally refine the mask; (2) if the IoU remains below τ_{IoU}, a full ground-truth mask is supplied instead. This hybrid strategy balances efficiency and effectiveness, since clicks enable rapid local adjustments, while full masks recover complex errors that cannot be easily fixed by clicks.

During correction, the simulated user mask M_t^{corr} supervises online updates of the LoRA adapter $\mathcal{A}$ while the backbone remains frozen. For subsequent frames $F_{t'}$, the updated adapter produces refined predictions $M_{t'}^{\mathcal{A}}$. Masks with $\text{IoU} \geq \tau_{\text{IoU}}$ are accepted and stored in the memory bank for future propagation; otherwise, further corrections are applied.

In our LoRA training, the LoRA is configured with rank of 4, $\alpha = 4$, dropout of 0.1, and a learning rate of 1×10^{-4}. The LoRA is injected to each layer of the mask decoder, and each LoRA is trained for 40 epochs with early stop.

Datasets and Evaluation We evaluate our method on four challenging VOS benchmarks: *VOST* [45], *LVOSv2* [19], *MOSEv2* [11] and *SA-V* [37] test and val set. These benchmarks include challenging scenarios such as object separation, heavy occlusion, and long video sequences, which require frequent user corrections. Detailed dataset descriptions are provided in Appendix A.

We measure the number of corrections required to satisfy the target quality threshold. For segmentation performance, we follow standard VOS metrics: $\mathcal{J}\&\mathcal{F}$, which combines region similarity ($\mathcal{J}$, IoU) and contour accuracy ($\mathcal{F}$). For VOST, we follow the official evaluation protocol and report $\mathcal{J}$ only. To estimate user interaction time, we adopt annotation time statistics from SAM2 [37] and EVA-VOS [9]: localization takes $T_{\text{loc}} = 1$ sec and each click requires $T_{\text{click}} = 1.5$ sec, while a full-mask annotation takes 80 sec.

4.2. Empirical Analysis of User Interaction

We begin by examining how user corrections occur in promptable video segmentation systems. Figure 3 analyzes user correction patterns across different VOS benchmarks. We observe a long-tailed distribution of corrections (Figure 3 (a)): while many videos require few interventions, a small subset of challenging sequences accounts for most of the total user effort. Correspondingly, the average number of corrections is modest, but the high-interaction subset (≥ 10 corrections) requires up to 3 times more effort (Figure 3 (b)). These are also the cases where user corrections yield the largest performance improvements (Figure 3 (c)), indicating feedback is most valuable precisely where it is most costly. Since these high-interaction cases dominate both user burden and observable benefit, our LIT-LoRA experiments primarily target this challenging subset. Additionally, corrections are temporally concentrated at initialization and again later in the sequence due to temporal drift (Figure 3 (d)), showing that errors reappear over time. This motivates converting user feedback into online adaptation to reduce repeated user interaction effort.

Table 1. **Comparison of user corrections and annotation time across datasets under different IoU thresholds.** LIT consistently reduces both user corrections and annotation time at $\tau_{\text{IoU}} = 0.5$ and 0.75.

		(a) Average user corrections per video					(b) Average annotation time per video (min)				
τ_{IoU}	**Method**	**VOST**	**LVOSv2**	**MOSEv2**	**SA-V Val**	**SA-V Test**	**VOST**	**LVOSv2**	**MOSEv2**	**SA-V Val**	**SA-V Test**
0.5	Original	27.43	33.59	31.48	20.66	20.90	18.42	14.83	22.49	13.07	13.26
	LIT	18.24	14.83	22.49	12.90	13.09	12.91	11.86	18.00	10.01	10.73
	Reduced	**↓33.51%**	**↓23.35%**	**↓18.22%**	**↓18.16%**	**↓22.35%**	**↓29.94%**	**↓20.03%**	**↓19.98%**	**↓22.44%**	**↓18.01%**
0.75	Original	40.40	56.32	41.47	27.48	29.83	37.70	44.09	40.08	25.01	26.90
	LIT	39.47	46.15	41.96	26.18	28.16	31.01	36.65	33.15	21.13	23.00
	Reduced	**↓23.65%**	**↓20.10%**	**↓16.98%**	**↓17.94%**	**↓17.89%**	**↓21.43%**	**↓20.58%**	**↓21.00%**	**↓19.32%**	**↓18.35%**

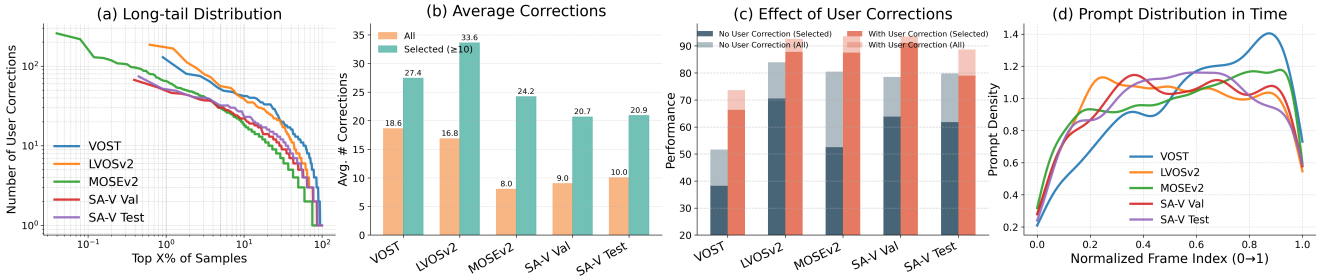


Figure 3. **User interaction patterns and the impact across datasets.** (a) The number of user corrections follows a clear long-tailed distribution: a small fraction of challenging videos accounts for the majority of interactions. (b) The challenging cases (≥ 10 corrections) require substantially more user inputs than the dataset average. (c) User feedback consistently improves segmentation performance, especially for the challenging subset. (d) Corrections are not uniformly distributed in time; most prompts occur in the early to late portions of each sequence, indicating the recurrence of errors.

4.3. Main Results of LIT-LoRA

Reducing user corrections and annotation time Table 1 (a) (top) shows the average number of user corrections required for each dataset, where each frame is required to achieve an IoU greater than $\tau_{\text{IoU}} = 0.5$. Across all four datasets, our method reduces the user corrections by 18% to 34%. The largest gains appear on VOST, which differs from SAM2's typical behavior because objects often split into multiple parts while SAM2 usually segments only one.

We further evaluate annotation cost in terms of total simulated annotation time as defined in Section 4.1, including the training and inference latency introduced by the LIT-LoRA module. As reported in Table 1 (b), LIT-LoRA reduces total annotation time by an average of 22.1% across datasets. With a 3–5 minute reduction per video, it can save hours of annotation time for the entire dataset.

We additionally evaluate LIT-LoRA under a stricter quality requirement ($\tau_{\text{IoU}} = 0.75$), where higher mask precision is demanded at every frame. As shown in Table 1 (bottom), LIT-LoRA consistently reduces user corrections and annotation time by 17–24% across datasets. Although the relative reduction is smaller than under $\tau_{\text{IoU}} = 0.5$, this is expected: stricter thresholds reject corrections that would otherwise be accepted, thereby increasing the overall user effort. In contrast, the computational overhead of LIT-LoRA remains constant at ∼0.5s per update. As a result, the absolute amount of user effort saved by LIT-LoRA remains substantial under stricter requirements, highlighting a favorable trade-off between lightweight computation and human annotation cost.

Improving accuracy when fixing user corrections To assess how effectively each correction improves segmentation, we compare our method with the baseline by allowing up to a fixed number of user corrections per video and evaluating the resulting propagated performance. This setup simulates a realistic interactive scenario in which the user's correction budget is limited. As shown in Figure 4, our method consistently achieves higher performance than the baseline under the same number of corrections. This validates the advantage of our approach: rather than applying corrections in a static manner, we adapt the model itself using lightweight online training, which allows each correction to improve not only the current frame but also generalize to future errors.

Lightweight overhead We evaluate the runtime of LIT-LoRA on an RTX Ada 6000 GPU. The LoRA module introduces only 35K trainable parameters, ($\sim 0.01\%$ of full-model fine-tuning), and each correction requires approximately 0.5 ± 0.2 s of online training. By comparison, the dominant cost in interactive segmentation remains the hu-

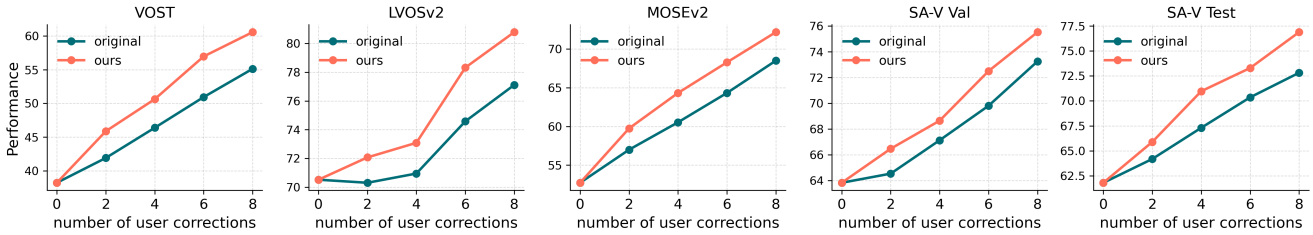


Figure 4. **Performance under different numbers of user corrections.**

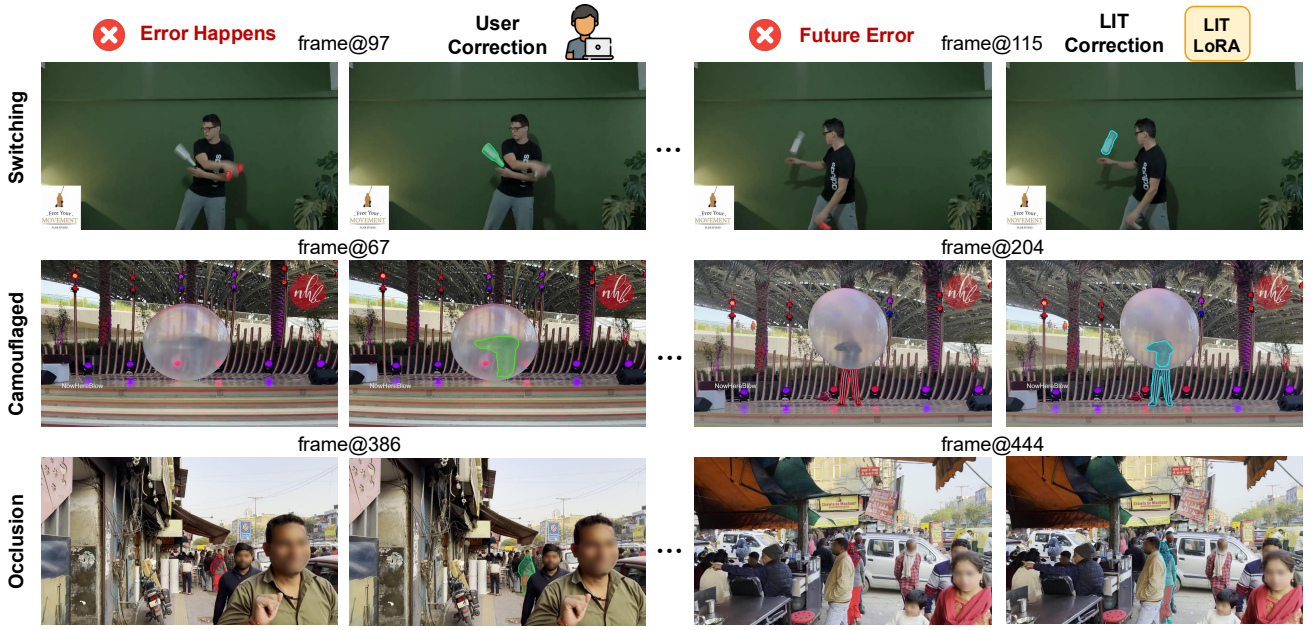


Figure 5. **Qualitative results.**

man input: annotating a single object typically takes about 1 sec to locate and 1.5 sec per click according to SAM2, and annotating a difficult object with a full mask can take up to 80 sec [9]. Under our hybrid correction strategy, the average per-correction annotation time is 30.91 sec at $\tau_{\text{IoU}} = 0.5$ and 45.62 sec at $\tau_{\text{IoU}} = 0.75$. These results show that the training overhead of LIT-LoRA is negligible relative to human annotation. Consequently, the system maintains its lightweight, real-time responsiveness and is well-suited for interactive workflows.

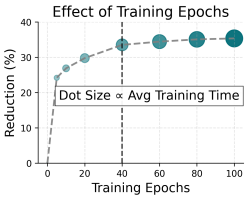
Qualitative results Figure 5 shows qualitative examples of our LIT-LoRA corrections when errors occur. Using prior correction signals, LIT-LoRA resolves recurring errors in later frames, including object switching, camouflaged objects, and occlusions.

User study To validate our method with real human annotators, we conducted a small-scale user study. We built an annotation interface on top of the SAM2 demo with online correction and LIT-LoRA, based on point corrections. We recruited 6 volunteers, who were first given time to familiarize themselves with the GUI. Each participant annotated 8 randomly selected videos from the VOST dataset (one used for familiarization) using both the baseline and our method in randomized order. The study follows the same online protocol: users pause and correct immediately once they see an error. To avoid familiarity bias, users reviewed the ground-truth masks before each session. We measure both correction number and correction time, which includes pausing, locating errors, and making corrections. We observe consistent reductions across users in correction number (**41.92%**) and correction time (**23.04%**). Note that the study used point corrections rather than masks, the relative time savings will be greater with more intensive corrections.

Adapt to other models and tasks A key strength of our method is its adaptation: it functions as a plug-and-play module that integrates into user interactive systems without modifying the architecture or retraining the backbone.

Table 2. **User correction reduction across models and tasks.**

(a) VOS			(b) Fine-grained image classification		
Dataset	**VOST**		**CUB-200-2011**	**Stanford Cars**	**SUN397**
Model	**DAM4SAM**	**SAMURAI**	**CLIP**		
Original	34.60	26.96	13.04	13.38	13.92
LIT	22.46	21.23	8.53	7.57	8.95
Reduction	↓35.09%	↓21.25%	↓34.55%	↓43.40%	↓35.70%



(a) Effect of training epochs

Method	**# Clicks ↓**	**#Params**
Original (no adaptation)	27.43	—
Replace original	32.47	35K
LIT-LoRA (No CL)	21.43	35K
LIT-FT	17.46	3.3M (∼100×)
LIT-LoRA (3 LoRAs)	17.87	105K (3×)
LIT-LoRA (ours)	**18.24**	**35K**

(b) Comparison of adaptation strategies

Figure 6. **Ablation studies on VOST dataset**

We first validate the adaptation on different VOS models. We apply our method to two recent SAM2-based models: DAM4SAM [47] and SAMURAI [58], and evaluate them on the VOST dataset. As shown in Table 2 (a), our approach consistently reduces the number of user corrections.

We further evaluate our method beyond video segmentation to fine-grained image classification. We use CLIP ViT-B/32 in a zero-shot configuration, converting classification into an online streaming annotation task. Images are grouped by the model's initial predicted label and processed sequentially. We adopt top-3 accuracy to decide when user correction is required: if the ground-truth label does not appear in the top-3 predictions, a misclassification is flagged and the user provides the correct label. Each correction supervises an update to a lightweight LoRA module, enabling the model to quickly adapt to the observed error. As new images in the group stream in and errors occur, we first query the LoRA module; if the correct label appears within its top-3 predictions, we accept the prediction. Otherwise, the user corrects the label and the LoRA is updated again. This simulates a realistic annotation scenario where visually and semantically similar mistakes recur and the model continuously improves through user feedback.

We conduct experiments on CUB-200-2011 [51], Stanford Cars [27], and SUN397 [57], measuring the average number of user corrections required per class for classes that require at least five corrections. As shown in Table 2 (b), our method consistently reduces the annotation burden by 35% - 43%. The results highlight that our LIT-LoRA not only transfers across model architectures, but also extends to tasks involving image–text alignment.

4.4. Ablation Studies

Number of Epochs We vary the number of training epochs used for LoRA training and report the reduction in user corrections along with the corresponding training time (Figure 6 (a)) on VOST. Increasing epochs improves correction reduction, but the gain quickly saturates: performance rises substantially from 5 to 40 epochs, while further increasing to 60–100 epochs yields only marginal improvement (< 2 pp). In contrast, training latency grows roughly proportionally to the number of epochs. Considering this trade-off, we use 40 epochs in our experiments, which achieves near-optimal correction reduction while keeping the per-correction training overhead low.

Different Design Choices We compare other adaptation strategies: (1) *Replace Original*: directly fine-tune the mask decoder at each correction, instead of maintaining a separate LIT-LoRA module; (2) *LoRA (No CL)*: fine-tune the LoRA only at the first correction, without continual learning for later corrections; (3) *LIT-FT*: fine-tune the entire mask decoder within the LIT framework instead of using LoRA modules; (4) *LIT-LoRA (3-LoRAs)*: maintain up to three LoRA modules, where a new LoRA is initialized whenever the current one fails. For each new correction, the user selects the best prediction among the three, and only the selected LoRA is updated.

Figure 6 (b) summarizes correction efficiency and parameter cost. Simply finetuning the original decoder leads to worse performance, suggesting that naïve fine-tuning can overfit to errors and disrupt stable representations. Finetuning a single LoRA only once is also insufficient, confirming the importance of continual learning during interaction. While full fine-tuning and the 3-LoRA setup reduce clicks, they are substantially heavier, and the 3-LoRA approach additionally introduces cognitive overhead by requiring users to choose among multiple predictions. Our design of LIT-LoRA achieves an average of 18.24 clicks with only 35K parameters, providing the best balance between efficiency and usability.

5. Conclusion

We introduce Live Interactive Training (LIT), a framework for live model adaptation in user-interactive visual systems. Unlike models like SAM2, which receive feedback but cannot learn from it, LIT enables online updates during inference through lightweight and modular components. Our implementation, LIT-LoRA, instantiated in video segmentation on top of SAM2, effectively reduces user correction effort on challenging videos. We further demonstrate the generality of LIT by applying it to multiple SAM2 variants and CLIP-based image classification tasks, achieving consistent improvements on correction efficiency across models and domains. We establish a step toward efficient, adaptive, and collaborative human–AI visual systems for complex real-world scenarios.

Acknowledgements

This research is supported in part by the Cornell–LinkedIn Partnership and the National Science Foundation (IIS-2144117, IIS-2107161, and IIS-2505098). Yihong Sun is supported by an NSF Graduate Research Fellowship.

References

[1] Barsegh Atanyan, Levon Khachatryan, Shant Navasardyan, Yunchao Wei, and Humphrey Shi. Continuous adaptation for interactive segmentation using teacher-student architecture. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 789–799, 2024. 3

[2] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023. 2

[3] Pia Bideau and Erik Learned-Miller. It's moving! a probabilistic model for causal motion segmentation in moving camera videos. In *European Conference on Computer Vision*, pages 433–449. Springer, 2016. 2

[4] Tianrun Chen, Ankang Lu, Lanyun Zhu, Chaotao Ding, Chunan Yu, Deyi Ji, Zejian Li, Lingyun Sun, Papa Mao, and Ying Zang. Sam2-adapter: Evaluating & adapting segment anything 2 in downstream tasks: Camouflage, shadow, medical image segmentation, and more. *arXiv preprint arXiv:2408.04579*, 2024. 2

[5] Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *European Conference on Computer Vision*, pages 640–658. Springer, 2022. 2

[6] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5559–5568, 2021. 2, 5

[7] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Joon-Young Lee, and Alexander Schwing. Putting the object back into video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3151–3161, 2024. 2

[8] Xuelian Cheng, Huan Xiong, Deng-Ping Fan, Yiran Zhong, Mehrtash Harandi, Tom Drummond, and Zongyuan Ge. Implicit motion handling for video camouflaged object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13864–13873, 2022. 2

[9] Thanos Delatolas, Vicky Kalogeiton, and Dim P Papadopoulos. Learning the what and how of annotation in video object segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6951–6961, 2024. 2, 5, 7

[10] Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, Philip HS Torr, and Song Bai. Mose: A new dataset for video object segmentation in complex scenes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 20224–20234, 2023. 12

[11] Henghui Ding, Kaining Ying, Chang Liu, Shuting He, Xudong Jiang, Yu-Gang Jiang, Philip HS Torr, and Song Bai. Mosev2: A more challenging dataset for video object segmentation in complex scenes. *arXiv preprint arXiv:2508.05630*, 2025. 5, 12

[12] Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature machine intelligence*, 5(3):220–235, 2023. 3

[13] Shuangrui Ding, Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Yuwei Guo, Dahua Lin, and Jiaqi Wang. Sam2long: Enhancing sam 2 for long video segmentation with a training-free memory tree. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13614–13624, 2025. 1, 2

[14] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024. 2

[15] Yossi Gandelsman, Yu Sun, Xinlei Chen, and Alexei Efros. Test-time training with masked autoencoders. *Advances in Neural Information Processing Systems*, 35:29374–29385, 2022. 3

[16] Ziqi Gao, Qichao Wang, Aochuan Chen, Zijing Liu, Bingzhe Wu, Liang Chen, and Jia Li. Parameter-efficient fine-tuning with discrete fourier transform. *arXiv preprint arXiv:2405.03003*, 2024. 3

[17] Moritz Hardt and Yu Sun. Test-time training on nearest neighbors for large language models. *arXiv preprint arXiv:2305.18466*, 2023. 3

[18] Jiangpeng He, Zhihao Duan, and Fengqing Zhu. Cl-lora: Continual low-rank adaptation for rehearsal-free class-incremental learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 30534–30544, 2025. 3

[19] Lingyi Hong, Zhongying Liu, Wenchao Chen, Chenzhi Tan, Yuang Feng, Xinyu Zhou, Pinxue Guo, Jinglun Li, Zhaoyu Chen, Shuyong Gao, Wei Zhang, and Wenqiang Zhang. Lvos: A benchmark for large-scale long-term video object segmentation, 2024. 2, 5, 12

[20] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 3

[21] Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Lee. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 5254–5276, 2023. 3

[22] Zi-Yuan Hu, Yanyang Li, Michael R Lyu, and Liwei Wang. Vl-pet: Vision-and-language parameter-efficient tuning via granularity control. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3010–3020, 2023. 3

[23] Devanish N Kamtam, Joseph B Shrager, Satya Deepya Malla, Xiaohan Wang, Nicole Lin, Juan J Cardona, Serena Yeung-Levy, and Clarence Hu. A fine-tuned foundational model surgisam2 for surgical video anatomy segmentation and detection. *Scientific Reports*, 15(1):35961, 2025. 2

[24] Lei Ke, Mingqiao Ye, Martin Danelljan, Yifan Liu, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Segment anything in high quality. In *NeurIPS*, 2023. 2

[25] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023. 1, 2, 5

[26] Theodora Kontogianni, Michael Gygli, Jasper Uijlings, and Vittorio Ferrari. Continuous adaptation for interactive object segmentation by learning from corrections. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 579–596. Springer, 2020. 3

[27] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 8

[28] Minglei Li, Peng Ye, Yongqi Huang, Lin Zhang, Tao Chen, Tong He, Jiayuan Fan, and Wanli Ouyang. Adapter-x: A novel general parameter-efficient fine-tuning framework for vision. *arXiv preprint arXiv:2406.03051*, 2024. 3

[29] Yixiao Liu. Fine-tuning sam2 for generalizable polyp segmentation with a channel attention-enhanced decoder. *Advanced Medical Research*, 4(1):1–9, 2025. 2

[30] Zichen Liu, Yue Yu, Hao Ouyang, Qiuyu Wang, Ka Leong Cheng, Wen Wang, Zhiheng Liu, Qifeng Chen, and Yujun Shen. Magicquill: An intelligent interactive image editing system. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13072–13082, 2025. 2

[31] Anusha Nagabandi, Chelsea Finn, and Sergey Levine. Deep online learning via meta-learning: Continual adaptation for model-based rl. *arXiv preprint arXiv:1812.07671*, 2018. 2

[32] Yao Ni, Shan Zhang, and Piotr Koniusz. Pace: Marrying generalization in parameter-efficient fine-tuning with consistency regularization. *Advances in Neural Information Processing Systems*, 37:61238–61266, 2024. 3

[33] Mahdi Nikdan, Soroush Tabesh, Elvir Crnčević, and Dan Alistarh. Rosa: Accurate parameter-efficient fine-tuning via robust adaptation. *arXiv preprint arXiv:2401.04679*, 2024. 3

[34] Xiaohan Peng, Janin Koch, and Wendy E Mackay. Designprompt: Using multimodal interaction for design exploration with generative ai. In *Proceedings of the 2024 ACM Designing Interactive Systems Conference*, pages 804–818, 2024. 2

[35] Yu-Yang Qian, Peng Zhao, Yu-Jie Zhang, Masashi Sugiyama, and Zhi-Hua Zhou. Efficient non-stationary online learning by wavelets with applications to online distribution shift adaptation. In *Forty-first International Conference on Machine Learning*, 2024. 3

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 2

[37] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 1, 2, 3, 5, 12

[38] Doyen Sahoo, Quang Pham, Jing Lu, and Steven CH Hoi. Online deep learning: Learning deep neural networks on the fly. *arXiv preprint arXiv:1711.03705*, 2017. 2

[39] Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012. 3

[40] Ziqi Shi, Fan Lyu, Ye Liu, Fanhua Shang, Fuyuan Hu, Wei Feng, Zhang Zhang, and Liang Wang. Controllable continual test-time adaptation. *arXiv preprint arXiv:2405.14602*, 2024. 3

[41] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. Dinov3, 2025. 2

[42] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pages 9229–9248. PMLR, 2020. 3

[43] Yihong Sun, Xinyu Yang, Jennifer J Sun, and Bharath Hariharan. Tracking and understanding object transformations. *Advances in Neural Information Processing Systems*, 2025. 2

[44] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5227–5237, 2022. 3

[45] Pavel Tokmakov, Jie Li, and Adrien Gaidon. Breaking the" object" in video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22836–22845, 2023. 1, 2, 5, 12

[46] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35:10078–10093, 2022. 2

[47] Jovana Videnovic, Alan Lukezic, and Matej Kristan. A distractor-aware memory for visual object tracking with sam2. *arXiv preprint arXiv:2411.17576*, 2024. 1, 2, 8

[48] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for the 2017 davis challenge on video object segmentation. In *The 2017 DAVIS Challenge on Video Object Segmentation-CVPR Workshops*, 2017. 3

[49] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. *arXiv preprint arXiv:1706.09364*, 2017. 3

[50] Stéphane Vujasinović, Stefan Becker, Sebastian Bullinger, Norbert Scherer-Negenborn, Michael Arens, and Rainer Stiefelhagen. Strike the balance: On-the-fly uncertainty based user interactions for long-term video object segmentation. In *Proceedings of the Asian Conference on Computer Vision*, pages 2784–2802, 2024. 2

[51] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 8

[52] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7201–7211, 2022. 3

[53] Yaqing Wang, Sahaj Agarwal, Subhabrata Mukherjee, Xiaodong Liu, Jing Gao, Ahmed Hassan, and Jianfeng Gao. Adamix: Mixture-of-adaptations for parameter-efficient model tuning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5744–5760, 2022. 3

[54] Zhijie Wang, Yuheng Huang, Da Song, Lei Ma, and Tianyi Zhang. Promptcharm: Text-to-image generation through multi-modal prompting and refinement. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–21, 2024. 2

[55] Xiwen Wei, Guihong Li, and Radu Marculescu. Online-lora: Task-free online continual learning via low rank adaptation. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 6634–6645. IEEE, 2025. 3

[56] Yujie Wei, Jiaxin Ye, Zhizhong Huang, Junping Zhang, and Hongming Shan. Online prototype learning for online continual learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 18764–18774, 2023. 3

[57] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492, 2010. 8

[58] Cheng-Yen Yang, Hsiang-Wei Huang, Wenhao Chai, Zhongyu Jiang, and Jenq-Neng Hwang. Samurai: Adapting segment anything model for zero-shot visual tracking with motion-aware memory. *arXiv preprint arXiv:2411.11922*, 2024. 1, 2, 8

[59] Jiangwei Yu, Xiang Li, Xinran Zhao, Hongming Zhang, and Yu-Xiong Wang. Video state-changing object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20439–20448, 2023. 1, 2

[60] Maxime Zanella and Ismail Ben Ayed. Low-rank few-shot adaptation of vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1593–1603, 2024. 3

[61] Zechuan Zhang, Ji Xie, Yu Lu, Zongxin Yang, and Yi Yang. In-context edit: Enabling instructional image editing with in-context generation in large scale diffusion transformer. *arXiv preprint arXiv:2504.20690*, 2025. 3

[62] Long Zhao, Nitesh B Gundavarapu, Liangzhe Yuan, Hao Zhou, Shen Yan, Jennifer J Sun, Luke Friedman, Rui Qian, Tobias Weyand, Yue Zhao, et al. Videoprism: A foundational visual encoder for video understanding. *arXiv preprint arXiv:2402.13217*, 2024. 2

[63] Zihan Zhong, Zhiqiang Tang, Tong He, Haoyang Fang, and Chun Yuan. Convolution meets lora: Parameter efficient finetuning for segment anything model. *arXiv preprint arXiv:2401.17868*, 2024. 3

[64] Junbao Zhou, Ziqi Pang, and Yu-Xiong Wang. Rmem: Restricted memory banks improve video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18602–18611, 2024. 1, 2

[65] Peilin Zhou, Bo Du, and Yongchao Xu. Cellseg1: Robust cell segmentation with one training image. *arXiv preprint arXiv:2412.01410*, 2024. 3

[66] Yuli Zhou, Guolei Sun, Yawei Li, Yuqian Fu, Luca Benini, and Ender Konukoglu. Camsam2: Segment anything accurately in camouflaged videos. *arXiv preprint arXiv:2503.19730*, 2025. 1, 2