SWAPCAST: CROSS-VIDEO TRANSITION SUPERVISION FOR CONSISTENT VIDEO RETRIEVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Consistent Video Retrieval (CVR) aims to retrieve a video clip that follows a textual instruction while remaining coherent with the preceding visual history. Residual target prediction provides a viable solution to this task, but does not explicitly encourage learned transitions to remain useful across different visual contexts. We propose **SwapCAST**, a transition-recomposition framework that separates receiver-specific context from observed or instruction-predicted transition codes and combines them through a shared residual recomposer. During training, a transition observed in a compatible donor video is recomposed with the receiver context and supervised toward the receiver continuation. Donor-target discrimination and wrong-transition constraints discourage trivial donor copying and encourage sensitivity to the supplied transition. At inference, retrieval requires only the receiver history and textual instruction, with no donor information.

Using frozen CLIP features, adding cross-video swap supervision to the same factorized base improves mean prediction-only retrieval accuracy by 2.39 and 3.86 percentage points on YouCook2 and CrossTask, respectively, across three training seeds. The complete model achieves 58.53% and 49.30% accuracy with calibrated fusion under our reconstructed CVR protocols. Observed-transition diagnostics further show sensitivity to transition compatibility and direction, while single-seed donor interventions provide supporting evidence for the importance of donor construction and endpoint order.

1 INTRODUCTION

Video–text retrieval ranks video content by its compatibility with a language query and has benefited substantially from large-scale vision–language learning Bain et al. (2021); Luo et al. (2022). Retrieving a continuation of an observed procedure, however, requires more than matching the requested action. A candidate may depict the correct action while changing the actor, objects, or environment; conversely, it may preserve these visual cues while depicting the wrong procedural stage. Consistent Video Retrieval (CVR), introduced by CAST Liu et al. (2026), therefore conditions retrieval on both visual history and a textual instruction. CAST predicts a context-conditioned residual update in a frozen visual embedding space, explicitly modeling visual state evolution. Nevertheless, its within-example target prediction does not explicitly encourage a transition observed in one visual context to remain useful in another.

Procedural videos provide a natural supervisory signal for learning such transferability. Different videos may exhibit compatible state changes despite variations in actors, objects, and appearance. A transition observed in one video can therefore supervise continuation prediction in another, provided that the result retains the receiver's visual context. As illustrated in Fig. 1, the donor supplies evidence about the change, while the desired endpoint remains the receiver continuation. This observation motivates **cross-video transition reuse supervision**, which complements the context-conditioned residual prediction used by CAST.

We introduce **SwapCAST**, which factorizes continuation prediction into a persistent-context pathway and a transition pathway. An observed-transition encoder extracts a code from paired visual states, while an instruction-conditioned predictor estimates a code from receiver context and language. A shared residual recomposer maps either code, together with receiver context and the latest observation, to a target embedding. During swap training, the observed code from a compatible

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.

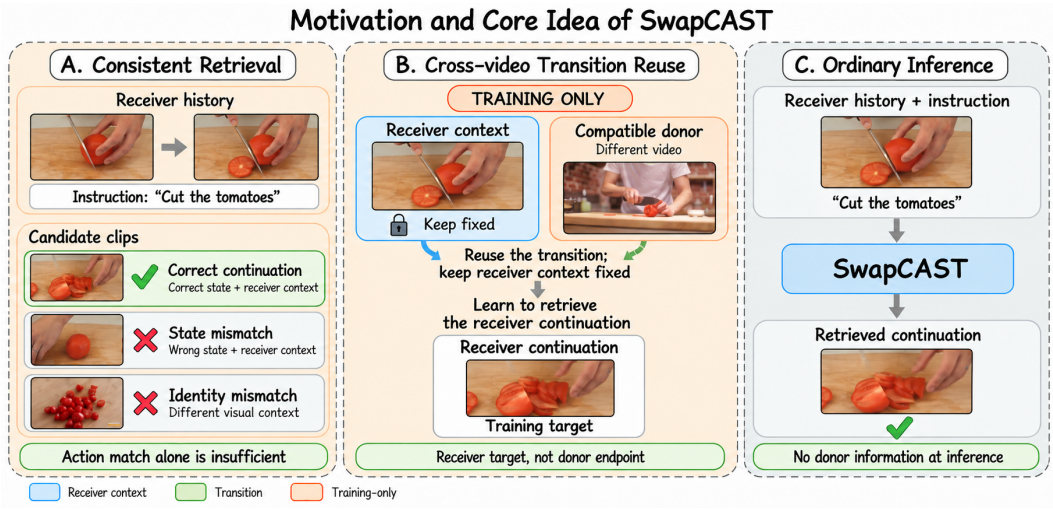

Figure 1: Motivation and core idea. (A) A valid continuation must match both the requested procedural state and receiver context. (B) Training reuses a compatible transition from another video while keeping the receiver continuation as the target. (C) Ordinary inference uses only receiver history and instruction. Video frames are illustrative.

donor replaces the receiver transition code, while the receiver context and continuation target remain unchanged. Donor-endpoint negatives discourage copying the donor, and wrong-transition and relational constraints encourage sensitivity to transition content and separation of representation roles. The observed-transition encoder and donor records are used only during training and are unnecessary for ordinary inference.

We evaluate SwapCAST on CVR protocols reconstructed from YouCook2 Zhou et al. (2018) and CrossTask Zhukov et al. (2019). Our main evidence is a controlled comparison within the same factorized architecture: adding the complete swap objective, combining transition recomposition and donor-endpoint discrimination, to the base model improves three-seed mean prediction-only accuracy by **2.39 and 3.86 percentage points**. The complete model reaches **54.16% and 48.96% before fusion**, and **58.53% and 49.30% with calibrated fusion**. Published CAST results provide external context, not controlled head-to-head comparisons on our candidate pools.

Our contributions are threefold:

- We introduce **cross-video transition reuse with receiver-target supervision** for procedural CVR, enabling a transition observed in one video to supervise continuation prediction in another while preserving the receiver's visual context.
- We develop a **shared residual-recomposition architecture** that combines receiver context with observed, instruction-predicted, or donor-derived transition codes, providing a unified prediction target across all three pathways.
- We conduct **controlled training-objective comparisons and transition-focused diagnostics** that characterize both the benefits and limitations of this supervision under donor-free ordinary inference.

2 Related Work

Video–text retrieval. Video–text retrieval ranks clips by their relevance to a textual query. Frozen in Time learns joint video–text embeddings from image and video data Bain et al. (2021), while CLIP4Clip adapts pretrained CLIP representations to video retrieval Luo et al. (2022). Beyond global embedding alignment, HERO combines cross-modal and temporal transformers with local/global alignment and frame-order objectives Li et al. (2020). Such temporal modeling enriches video–language representations, but text–clip relevance alone does not require a candidate to con-

tinue a particular receiver history. SwapCAST addresses this additional requirement by predicting a receiver-specific continuation and supervising transition reuse across compatible videos.

Composed retrieval. Composed retrieval combines a visual reference with a textual modification: TIRG learns residual composition Vo et al. (2019), CoVR-2 constructs video retrieval triplets Ventura et al. (2024), and TF-CoVR emphasizes temporally fine-grained differences Gupta et al. (2025). CVR additionally requires a retrieved clip to continue an observed procedural history.

Procedural states and transitions. Prior work studies transferable visual transformations Isola et al. (2015), jointly learns object states and state-modifying actions Souček et al. (2022), and uses state-change descriptions and counterfactuals to learn procedure-aware representations Kung et al. (2025). CAST Liu et al. (2026) introduces CVR and predicts history- and instruction-conditioned residual state updates. We retain this formulation and add explicit supervision for reusing observed transitions across compatible videos.

Factorization and latent exchange. Video representations have been separated into stationary and time-varying components Denton & Birodkar (2017); Swapping Autoencoder exchanges structure and texture codes Park et al. (2020). Visual analogy making also transfers the change in an example pair to another input with corresponding-target supervision Reed et al. (2015). SwapCAST applies these ideas to procedural retrieval through compatible cross-video supervision and a recomposer shared by observed and instruction-predicted codes. Its factorization denotes roles encouraged by training, not identifiable or fully independent semantic factors.

3 METHODS

Figure 2 outlines SwapCAST. The model encodes receiver history as a context code and uses this code with the instruction to predict a transition. A shared residual recomposer combines the receiver context and latest-frame anchor with one transition code at a time to produce a candidate-scoring embedding. During training, receiver-observed codes support reconstruction, compatible donor codes provide swap supervision toward the receiver continuation, and wrong-donor codes provide a relative-score constraint; donor-endpoint negatives discourage donor copying. At inference, only the predicted transition is used, requiring receiver history and instruction without donor information. We detail the task formulation (Sec. 3.1), representations and recomposition (Sec. 3.2), cross-video swapping (Sec. 3.3), and joint training and inference (Sec. 3.4) below.

3.1 PROBLEM FORMULATION

For receiver i, let $\mathcal{H}_i = (v_{i,1}, \ldots, v_{i,\ell_i})$ be its observed history, $1 \le \ell_i \le L$, and q_i the requested instruction. The fixed candidate pool $\mathcal{C}_i$ contains one annotated continuation v_i^+ and nine distractors, including within-video state negatives and cross-video identity negatives. Let $\mathbf{x}(v)$ and $\mathbf{t}_i$ be the L_2-normalized frozen CLIP visual and instruction embeddings Radford et al. (2021), and $\mathbf{x}_i = \mathbf{x}(v_i^+)$. For a predicted endpoint $\hat{\mathbf{x}}_i$, prediction-only retrieval is

$$v_i^* = \arg\max_{v \in \mathcal{C}_i} \langle \hat{\mathbf{x}}_i, \mathbf{x}(v) \rangle. \tag{1}$$

The target identity is unavailable to the scorer. The separate empty-history diagnostic uses the text-anchor fallback specified in Appendix A.2.

3.2 PERSISTENT–TRANSITION REPRESENTATION

The anchor $\mathbf{a}_i = \mathbf{x}(v_{i,\ell_i})$ is the latest valid history embedding, and $\bar{\mathbf{h}}_i$ is the mean over valid history embeddings, excluding padding. The persistent-context code is

$$\mathbf{p}_i = \mathcal{N}\big(g_p([\mathbf{a}_i; \bar{\mathbf{h}}_i])\big), \tag{2}$$

where $\mathcal{N}$ denotes L_2 normalization and $[\cdot;\cdot]$ concatenation. For nonempty histories, g_p does not receive the instruction. During training, an observed code uses the anchor, target, and their difference;

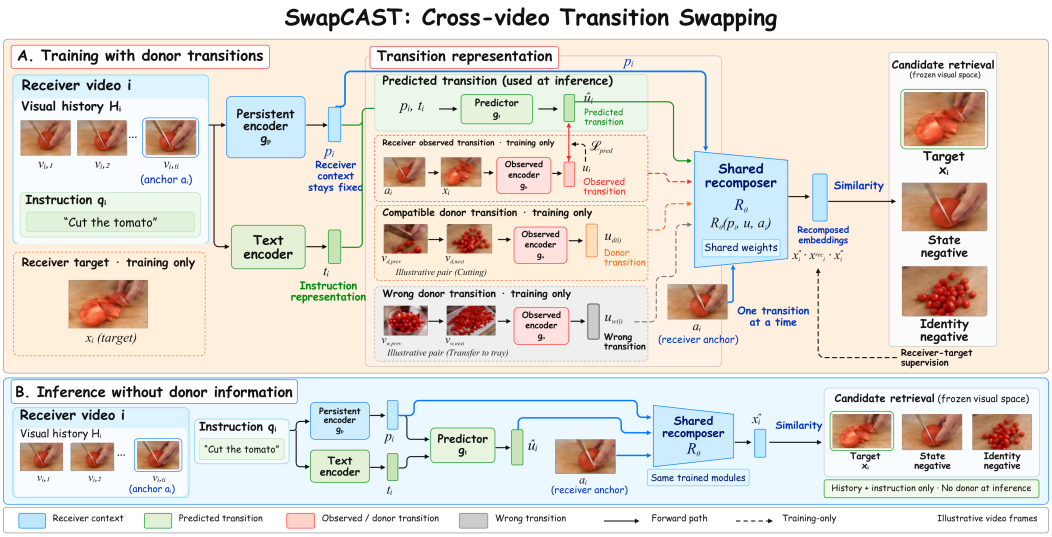

Figure 2: SwapCAST. The shared recomposer combines receiver context and anchor with one transition code at a time. Predicted codes support ordinary retrieval; receiver-observed codes support reconstruction; compatible donor codes support receiver-target swap supervision. Wrong codes provide a relative-score constraint. Observed encoders and donor branches are used only during training or explicit mechanism diagnostics.

an instruction-conditioned code predicts the transition without accessing the target:

$$\begin{aligned} \mathbf{u}_i &= \mathcal{N}\big(g_o([\mathbf{a}_i; \mathbf{x}_i; \mathbf{x}_i - \mathbf{a}_i])\big), \\ \hat{\mathbf{u}}_i &= \mathcal{N}\big(g_t([\mathbf{p}_i; \mathbf{t}_i])\big). \end{aligned} \tag{3}$$

A shared recomposer adds a learned residual to the receiver anchor:

$$\begin{aligned} R_\theta(\mathbf{p}, \mathbf{u}, \mathbf{a}) &= \mathcal{N}\big(\mathbf{a} + g_r([\mathbf{p}; \mathbf{u}])\big), \\ \hat{\mathbf{x}}_i = R_\theta(\mathbf{p}_i, \hat{\mathbf{u}}_i, \mathbf{a}_i), &\qquad \mathbf{x}_i^{\mathrm{rec}} = R_\theta(\mathbf{p}_i, \mathbf{u}_i, \mathbf{a}_i). \end{aligned} \tag{4}$$

This interface connects instruction-conditioned prediction and observed-transition reconstruction (Fig. 2). The observed code provides a visual description of a transition during training, whereas the predicted code must estimate that change from the available instruction and receiver history. Sharing R_θ makes the two codes act through the same endpoint prediction function. The residual anchor supplies an explicit path for receiver appearance; the persistent code additionally summarizes its history. These paths allow a donor code to be evaluated by how well it predicts the receiver continuation, without asking the donor encoder to reproduce the receiver's appearance directly. The four mappings are two-layer MLPs; their configuration is given in Appendix A. Architectural separation specifies functional roles, without guaranteeing complete disentanglement.

3.3 Cross-Video Transition Swapping

Compatible donors. Donors are drawn from different videos in the same task, separately within each data split. Compatibility considers both preceding and current procedural steps: CrossTask uses matching ordered canonical step pairs, while YouCook2 uses semantic correspondences between both caption pairs. These are approximate procedural matches, not guarantees of identical visual transformations. Full matching, confidence, and sampling rules are in Appendix A.3. Matching only the current action would not identify the change from the preceding state, so donor construction uses both endpoints. Candidate donors are ranked by confidence and sampled from the retained compatible set. Receivers without a compatible donor still contribute ordinary retrieval, prediction alignment, and reconstruction supervision.

Receiver-target supervision. For compatible donor $d(i)$, we exchange only the transition code:

$$\tilde{\mathbf{x}}_i = R_\theta(\mathbf{p}_i, \mathbf{u}_{d(i)}, \mathbf{a}_i). \tag{5}$$

Both receiver context $\mathbf{p}_i$ and anchor $\mathbf{a}_i$ remain fixed. The supervised target is $\mathbf{x}_i$, even though the transition code is extracted from another video's annotated endpoints. This separates the source of transition evidence from the appearance of the desired continuation.

Let $\mathcal{B}$ index the receiver minibatch, $\mathcal{D} \subseteq \mathcal{B}$ its donor-valid receivers, and $\mathcal{Q}_i^s, \mathcal{Q}_i^{\mathrm{id}}$ the state- and identity-negative sequences. For cosine similarity c, define the negative-set contrastive loss

$$\ell_{\mathrm{neg}}(\mathbf{y}, \mathbf{x}; \mathcal{Q}) = -\log \frac{e^{c(\mathbf{y},\mathbf{x})/\tau}}{e^{c(\mathbf{y},\mathbf{x})/\tau} + \sum_{\mathbf{n}\in\mathcal{Q}} e^{c(\mathbf{y},\mathbf{n})/\tau}}. \tag{6}$$

For receiver index set $\mathcal{I}$, type-aware retrieval combines in-batch targets with the two negative types:

$$\mathcal{R}_{\mathcal{I}}(\mathbf{y}; \mathcal{Q}^s, \mathcal{Q}^{\mathrm{id}}) = \frac{1}{|\mathcal{I}|} \sum_{i\in\mathcal{I}} \Bigg[-\log \frac{e^{c(\mathbf{y}_i,\mathbf{x}_i)/\tau}}{\sum_{j\in\mathcal{I}} e^{c(\mathbf{y}_i,\mathbf{x}_j)/\tau}} + \beta_s \ell_{\mathrm{neg}}(\mathbf{y}_i, \mathbf{x}_i; \mathcal{Q}_i^s) + \beta_{\mathrm{id}} \ell_{\mathrm{neg}}(\mathbf{y}_i, \mathbf{x}_i; \mathcal{Q}_i^{\mathrm{id}}) \Bigg]. \tag{7}$$

We use $\tau = 0.07$, $\beta_s = 5$, and $\beta_{\mathrm{id}} = 1$. The donor endpoint is appended to the identity negatives:

$$\begin{aligned} \mathcal{Q}_i^{\mathrm{id,swap}} &= \mathcal{Q}_i^{\mathrm{id}} \parallel [\mathbf{x}_{d(i)}], \\ \mathcal{L}_{\mathrm{swap}} &= \bar{\omega}_{\mathcal{D}}\, \mathcal{R}_{\mathcal{D}}(\tilde{\mathbf{x}}; \mathcal{Q}^s, \mathcal{Q}^{\mathrm{id,swap}}). \end{aligned} \tag{8}$$

The in-batch denominator in this loss contains only receiver targets indexed by $\mathcal{D}$, always using the *receiver target* $\mathbf{x}_i$ as the positive. The donor endpoint is a negative, not the supervision target. The mean donor confidence $\bar{\omega}_{\mathcal{D}} = |\mathcal{D}|^{-1} \sum_{i\in\mathcal{D}} \omega_{i,d(i)}$ scales the aggregate loss rather than weighting individual pairs. Concatenation $\parallel$ preserves multiplicity when the donor endpoint is already a negative. Padded negatives are masked; empty negative sequences and donor-empty minibatches contribute zero to their corresponding terms. Appendix A.4 provides the complete training formulations.

Wrong-transition supervision compares recompositions from matched and sampled wrong-step donors, encouraging a higher receiver-target similarity for the matched code. Both recompositions retain the same receiver context and anchor. The donor-endpoint negative discourages donor copying, while the wrong-code comparison discourages predicting from receiver context without using the supplied transition. The wrong-donor criteria and margin loss are specified in Appendix A.4.

3.4 Joint Training and Inference

Ordinary retrieval $\mathcal{L}_{\mathrm{ret}}$ supervises $\hat{\mathbf{x}}_i$. Prediction alignment $\mathcal{L}_{\mathrm{pred}}$ aligns $\hat{\mathbf{u}}_i$ with $\mathbf{u}_i$, and reconstruction $\mathcal{L}_{\mathrm{rec}}$ aligns $\mathbf{x}_i^{\mathrm{rec}}$ with $\mathbf{x}_i$. Relational losses encourage persistent-code similarity within videos ($\mathcal{L}_p$) and transition-code similarity between compatible videos ($\mathcal{L}_u$). Together with swap and wrong-transition supervision, the objective is

$$\begin{aligned} \mathcal{L} = \mathcal{L}_{\mathrm{ret}} &+ \lambda_{\mathrm{pred}}\mathcal{L}_{\mathrm{pred}} + \lambda_{\mathrm{rec}}\mathcal{L}_{\mathrm{rec}} \\ &+ h(e)\big(\lambda_{\mathrm{swap}}\mathcal{L}_{\mathrm{swap}} + \lambda_{\mathrm{wrong}}\mathcal{L}_{\mathrm{wrong}} + \lambda_p\mathcal{L}_p + \lambda_u\mathcal{L}_u\big). \end{aligned} \tag{9}$$

Here $h(e) = 0$ for the first two epochs and 1 thereafter; this activates losses without changing the learning rate. The frozen backbone is unchanged, while learned branches are optimized jointly without detaching the observed code during alignment. Swap gradients pass through the receiver persistent encoder, donor observed-transition encoder, and shared recomposer. Thus, the swapped branch can improve the context and recomposition functions used at inference, while prediction alignment connects the instruction-conditioned code to the learned observed-code space. All auxiliary formulations, weights, and optimization details are in Appendices A.4 and A.5.

At inference, only receiver history and instruction produce $\mathbf{p}_i$, $\hat{\mathbf{u}}_i$, and $\hat{\mathbf{x}}_i$; the observed encoder and donor records are unnecessary. Controlled ablations use prediction-only scoring $C_i(v)$. The main retrieval results additionally use calibrated fusion:

$$\begin{aligned} A_i(v) &= \langle \mathbf{t}_i, \mathbf{x}(v) \rangle, \quad B_i(v) = \langle \mathbf{a}_i, \mathbf{x}(v) \rangle, \\ C_i(v) &= \langle \hat{\mathbf{x}}_i, \mathbf{x}(v) \rangle, \quad S_i(v) = A_i(v) + w_v B_i(v) + w_p C_i(v). \end{aligned} \tag{10}$$

Calibration fixes w_v, w_p before final evaluation (Appendix B.4). All scores use the same precomputed frozen candidate embeddings.

Table 1: Main retrieval results with frozen CLIP ViT-B/32 features. SwapCAST uses calibrated fusion (mean ± sample SD over three seeds). † marks published CAST references Liu et al. (2026). The blocks use different candidate pools and protocols and are not controlled comparisons.

Method	YC Acc. ↑	YC MnR ↓	CT Acc. ↑	CT MnR ↓
Published references: original CAST protocol				
Zero-Shot (CLIP)†	25.03	3.60	16.83	4.15
Late Fusion (Heuristic)†	31.10	2.56	22.05	2.86
Late Fusion (Learned)†	36.60	2.53	25.52	2.86
Early Fusion†	35.99	2.28	35.29	2.36
CAST†	44.77	2.15	47.39	2.14
Our evaluation: reconstructed CVR protocol				
SwapCAST	58.53 ± 0.29	1.827 ± 0.022	49.30 ± 0.08	1.967 ± 0.016

4 Experiments

4.1 Experimental Setup

Data and protocol. We construct CVR evaluations from YouCook2 Zhou et al. (2018) and CrossTask Zhukov et al. (2019). For YouCook2, we split the original training videos approximately 90%/10% for training/calibration and evaluate on the original validation set (2,594 queries). For CrossTask, videos within each of the 18 tasks are split approximately 72%/8%/20% for training/calibration/evaluation (2,515 evaluation queries). The subsets are video-disjoint, with split seed 1337. Each query has a fixed pool of one target and nine negatives, typically three each of state, identity, and easy negatives, with backfilling when a category is insufficient. State negatives come from other steps in the receiver video; identity negatives come from other videos and are matched by caption similarity (YouCook2) or task/step identifiers (CrossTask). Candidates remain within the query's split. Full construction details are provided in Appendix B.

Training and evaluation. We use frozen CLIP ViT-B/32 features Radford et al. (2021), at most $L = 5$ history clips, and 20 training epochs with batch size 512. The checkpoint is selected by prediction-only (C-only) Top-1 accuracy on the calibration split. Fusion weights in Eq. 10 are then selected independently for each training seed on that split; evaluation scores guide neither selection. Our primary results use calibrated fusion, while controlled objective ablations use prediction-only scoring. Ordinary inference always uses only receiver history and instruction. We report mean ± sample SD over training seeds 1337, 2027, and 3407, keeping the split and candidate pools fixed. Metrics are Top-1 accuracy (Acc., %) and mean target rank (MnR). State/Identity Accuracy measures whether the target outranks every distractor of the corresponding type, averaged over queries containing that type. Appendices A.5 and B.4 give optimization and calibration details. Published CAST results Liu et al. (2026) are external references from a different protocol; claims about our objectives rely on controlled variants under our common reconstructed protocol.

4.2 Quantitative Results on the CVR Benchmark

With calibrated fusion, SwapCAST achieves 58.53% accuracy on YouCook2 and 49.30% on CrossTask (Table 1). To assess the predicted continuation representation independently of additional instruction–candidate and anchor–candidate scores, we also evaluate prediction-only ranking. This yields $54.16 \pm 0.59\%$ accuracy on YouCook2 and $48.96 \pm 0.44\%$ on CrossTask, respectively 4.37 and 0.34 percentage points below calibrated fusion. Table 2 reports YouCook2 consistency diagnostics. State and identity discrimination are compatible with the intended factorization but do not identify its components in isolation. The published reference values provide external context rather than a controlled estimate of improvement over CAST. Complete scoring comparisons appear in Appendix C.2; qualitative retrieval examples appear in Sec. 4.5. In particular, equal-weight fusion improves YouCook2 accuracy to 58.27% but reduces CrossTask accuracy to 48.22%. The benefit of auxiliary scores is therefore dataset-dependent; their weights are calibrated separately for each dataset. The controlled prediction-only comparisons below assess learning effects without changes to inference-time fusion.

Table 2: Prediction-only retrieval and consistency diagnostics on YouCook2 (%). Protocols differ between blocks, as in Table 1. SwapCAST reports three-seed mean ± sample SD.

Method	Acc. ↑	State Acc. ↑	Ident. Acc. ↑
Published reference: original CAST protocol			
CAST†	42.60	50.81	75.99
Our evaluation: reconstructed CVR protocol			
SwapCAST	54.16 ± 0.59	62.40 ± 0.31	80.85 ± 0.52

Table 3: Controlled objective ablation. All variants share the factorized architecture and retrieval loss. Results are prediction-only accuracy (%; three-seed mean ± sample SD). "Relations" includes both persistent and transition regularization; M_2 and M_3 are separate additions to M_1. "Swap" denotes transition recomposition with donor-endpoint discrimination.

Variant	Training configuration	YouCook2 ↑	CrossTask ↑
M_0	Retrieval only	45.44 ± 0.45	41.37 ± 0.35
M_1	Base: M_0 + prediction + reconstruction	49.99 ± 0.64	43.51 ± 0.78
M_2	Base + relations	51.06 ± 0.69	45.81 ± 0.43
M_3	Base + swap objective	52.38 ± 0.36	47.37 ± 1.10
M_4	Base + swap + wrong	53.87 ± 0.79	48.23 ± 0.32
M_5	**Full**: base + swap + wrong + relations	$\mathbf{54.16 \pm 0.59}$	$\mathbf{48.96 \pm 0.44}$

4.3 Ablation Studies

Table 3 compares six training configurations with identical factorized architecture, frozen features, candidate pools, and prediction-only inference. These comparisons isolate training objectives within this architecture, not the contribution of factorization itself. M_0 retains the full prediction pathway with retrieval supervision alone. Adding transition prediction and target reconstruction jointly gives the base model M_1, improving accuracy by 4.55 points on YouCook2 and 2.14 on CrossTask; this comparison measures their combined contribution.

$M_1 \rightarrow M_3$ **evaluates the complete swap objective.** Transition recomposition and donor-endpoint discrimination are added jointly; relational and wrong-transition losses remain disabled. Three-seed mean accuracy increases by 2.39 and 3.86 points, respectively, under ordinary inference without donor information. These gains measure the joint effect of the objective, not the isolated contribution of transition transfer. Adding wrong-transition supervision ($M_3 \rightarrow M_4$) gives a further 1.49/0.86 points. Relations improve the base ($M_1 \rightarrow M_2$) by 1.07/2.30 points and the swap-plus-wrong model ($M_4 \rightarrow M_5$) by 0.29/0.73 points. Full SwapCAST has the highest mean accuracy, exceeding M_1 by 4.17/5.45 points across all additional objectives. Appendix C.1 reports individual loss deletions and swap-weight sensitivity; Appendix C.3 analyzes inference history truncation for models trained with $L = 5$. The supplementary deletions lower accuracy for each tested auxiliary loss on both datasets in seed 1337, while the swap-weight response is non-monotonic and dataset-dependent. These diagnostics do not replace repeated-run estimates, and the main configuration retains the predefined swap weight $\lambda_{\text{swap}} = 1$ rather than selecting it from evaluation results. History truncation almost eliminates the full model's benefit over the base at $L = 0$, while gains become larger once visual history is supplied. This is consistent with learning to use receiver context, although it does not establish that the persistent code alone causes the gains.

4.4 Analysis of Cross-Video Transition Swapping

We distinguish training interventions from observed-code diagnostics. Single-seed (1337) interventions train the base-plus-swap model M_3 with alternative donor construction, keeping relational and wrong-transition losses disabled (Appendix C.4). Compatible donors yield 52.38%/47.40% accuracy on YouCook2/CrossTask; random same-task donors reduce this to 50.31%/42.56%, and reversed donors to 51.78%/46.32%. The random control retains receiver coverage and confidence, while the reversed control retains the donor-endpoint negative. Removing that negative yields 51.62%/46.36%. These single-seed interventions support sensitivity to donor construction and a contribution from donor-target discrimination. They neither establish average effects across runs nor separate the three-seed gain into transition transfer and donor-negative contributions.

Table 4: Observed-transition diagnostics (Top-1, %). Base is M_1; Full is M_5. Annotated donor endpoints are supplied only for this analysis. Results average donor pairings within receivers, then equally over receivers and three model seeds.

Donor code	YC Base	YC Full	CT Base	CT Full
Compatible	40.42	48.33	36.53	43.56
Random same-task	27.50	37.08	30.99	35.23
Wrong	27.50	27.50	30.10	34.10
Reversed	26.67	25.83	30.51	35.99

Table 4 instead fixes receiver context and substitutes transition codes from annotated donor endpoints. These mechanism diagnostics are distinct from ordinary inference. They use 153 receiver queries on YouCook2 and 2,327 on CrossTask. Top-1 hits are averaged first over donor pairings within each receiver, then equally over receivers and over the three model seeds.

Full SwapCAST improves compatible-donor recomposition over the base by 7.91/7.03 points, reflecting the combined additional objectives. Compatible codes also exceed random same-task, wrong, and reversed codes on both datasets, supporting sensitivity to compatibility and temporal order. The wrong-donor heuristics do not guarantee semantic incompatibility in every case. These gains do not establish complete disentanglement: target-informed receiver self-reconstruction decreases on CrossTask from 62.94% to 57.97% despite improved cross-video recomposition. Appendix C.5 preserves all controls, including direct donor copying and raw feature-difference transfer, with full sampling details; Appendix D provides qualitative diagnostics.

4.5 Qualitative Retrieval Examples

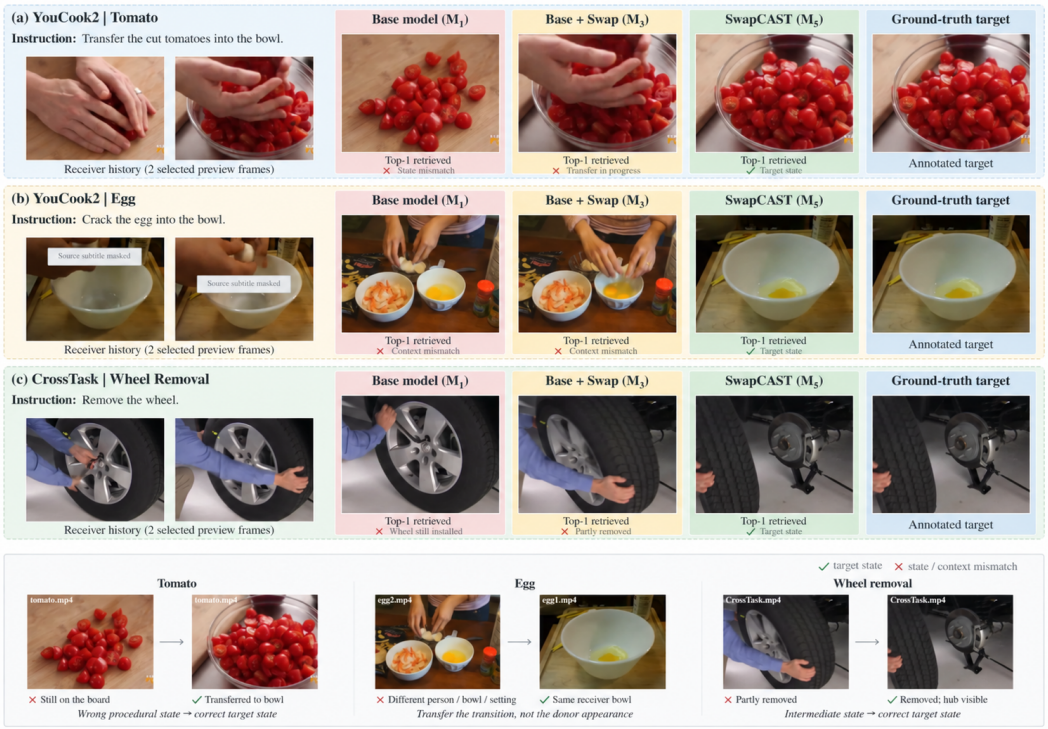

Figure 3: Qualitative retrieval examples comparing the factorized base model M_1, the corresponding model with the complete swap objective M_3, and the complete SwapCAST model M_5. All variants receive the same receiver history, instruction, and candidate pool, and use prediction-only scoring. For visualization, only representative history frames are displayed, while inference uses the complete available history with $L = 5$. The ground-truth continuation is shown for reference. No donor information is used at inference.

Figure 3 presents selected retrieval examples from YouCook2 and CrossTask, comparing the factorized base model, the corresponding variant augmented with swap supervision, and the complete SwapCAST model. All variants receive identical visual histories, textual instructions, and candidate pools, and candidates are ranked using the predicted target representation alone. For each query, we show the annotated target together with the top-ranked candidate produced by each variant. Visual annotations highlight the procedural change required by the instruction and the receiver-specific contextual cues that should remain consistent with the observed history.

The comparison between the factorized base and the base+swap variant provides a controlled view of how cross-video transition supervision is reflected in the learned retrieval representation, while the complete SwapCAST model shows the resulting behavior under the full training objective. In particular, the examples allow us to examine whether swap supervision helps the model favor candidates that realize the requested procedural transition without discarding contextual information inherited from the receiver. The selected examples show cases where the base or base+swap variant misses the target and the full model retrieves it. They illustrate candidate-level behavior rather than estimate the frequency of such corrections.

Importantly, no donor information is available at inference time in any of the displayed examples. All retrieval decisions are produced solely from the receiver's visual history and instruction. The qualitative comparison therefore examines how training with cross-video transition supervision influences ordinary instruction-conditioned retrieval, rather than demonstrating a retrieval procedure that directly accesses donor transitions.

5 Conclusion

SwapCAST trains observed procedural transitions for reuse across videos, recomposing compatible donor codes with receiver context while retaining the receiver continuation as the target. A shared residual recomposer connects observed and instruction-predicted transitions, allowing this training signal to improve retrieval that uses only history and language at inference. Controlled experiments on YouCook2 and CrossTask support the joint benefit of transition recomposition and donor-endpoint discrimination; donor interventions and observed-code diagnostics provide supporting evidence of sensitivity to transition compatibility and direction. Our ablations evaluate training objectives within a fixed factorized architecture; they do not establish an independent advantage of factorization over a matched non-factorized residual predictor. Limitations include approximate donor matching, frozen features that may miss subtle state changes, and evaluation on two procedural datasets with one backbone and short histories. The learned factorization is not complete disentanglement, and cross-video gains do not uniformly improve self-reconstruction. Broader datasets, longer histories, and more accurate transition matching remain directions for future study.

References

Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1728–1738, 2021. doi: 10.1109/ICCV48922.2021.00175. URL https://openaccess.thecvf.com/content/ICCV2021/html/Bain_Frozen_in_Time_A_Joint_Video_and_Image_Encoder_for_ICCV_2021_paper.html.

Emily L. Denton and Vighnesh Birodkar. Unsupervised learning of disentangled representations from video. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/2d2ca7eedf739ef4c3800713ec482e1a-Abstract.html.

Animesh Gupta, Jay Parmar, Ishan Rajendrakumar Dave, and Mubarak Shah. From play to replay: Composed video retrieval for temporally fine-grained videos. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4450. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/c192d0aefb044150948577bf6e7b3f62-Abstract-Datasets_and_Benchmarks_Track.html. Datasets and Benchmarks Track.

Phillip Isola, Joseph J. Lim, and Edward H. Adelson. Discovering states and transformations in image collections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1383–1391, 2015. doi: 10.1109/CVPR.2015.7298744. URL https://openaccess.thecvf.com/content_cvpr_2015/html/Isola_Discovering_States_and_2015_CVPR_paper.html.

Chi-Hsi Kung, Frangil Ramirez, Juhyung Ha, Yi-Ting Chen, David Crandall, and Yi-Hsuan Tsai. What changed and what could have changed? state-change counterfactuals for procedure-aware video representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12294–12306, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Kung_What_Changed_and_What_Could_Have_Changed_State-Change_Counterfactuals_for_ICCV_2025_paper.html.

Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. HERO: Hierarchical encoder for Video+Language omni-representation pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2046–2065. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.161. URL https://aclanthology.org/2020.emnlp-main.161/.

Yanqing Liu, Yingcheng Liu, Fanghong Dong, Budianto Budianto, Cihang Xie, and Yan Jiao. CAST: Modeling visual state transitions for consistent video retrieval. In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2603.08648.

Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. CLIP4Clip: An empirical study of CLIP for end to end video clip retrieval and captioning. *Neurocomputing*, 508:293–304, 2022. ISSN 0925-2312. doi: 10.1016/j.neucom.2022.07.028. URL https://www.sciencedirect.com/science/article/pii/S0925231222008876.

Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei A. Efros, and Richard Zhang. Swapping autoencoder for deep image manipulation. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/50905d7b2216bfeccb5b41016357176b-Abstract.html.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Scott E. Reed, Yi Zhang, Yuting Zhang, and Honglak Lee. Deep visual analogy-making. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/e07413354875be01a996dc560274708e-Abstract.html.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410/.

Tomáš Souček, Jean-Baptiste Alayrac, Antoine Miech, Ivan Laptev, and Josef Sivic. Look for the change: Learning object states and state-modifying actions from untrimmed web videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13956–13966, 2022. doi: 10.1109/CVPR52688.2022.01357. URL https://openaccess.thecvf.com/content/CVPR2022/html/Soucek_Look_for_the_Change_Learning_Object_States_and_State-Modifying_Actions_CVPR_2022_paper.html.

Lucas Ventura, Antoine Yang, Cordelia Schmid, and Gül Varol. CoVR-2: Automatic data construction for composed video retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):11409–11421, 2024. doi: 10.1109/TPAMI.2024.3463799. URL https://doi.org/10.1109/TPAMI.2024.3463799.

Nam Vo, Lu Jiang, Chen Sun, Kevin Murphy, Li-Jia Li, Li Fei-Fei, and James Hays. Composing text and image for image retrieval - an empirical odyssey. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6439–6448, 2019. doi: 10.1109/CVPR.2019.00660. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Vo_Composing_Text_and_Image_for_Image_Retrieval_-_an_Empirical_CVPR_2019_paper.html.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Luowei Zhou, Chenliang Xu, and Jason J. Corso. Towards automatic learning of procedures from web instructional videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. doi: 10.1609/aaai.v32i1.12342. URL https://ojs.aaai.org/index.php/AAAI/article/view/12342.

Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3537–3545, 2019. doi: 10.1109/CVPR.2019.00365. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Zhukov_Cross-Task_Weakly_Supervised_Learning_From_Instructional_Videos_CVPR_2019_paper.html.

AI USE STATEMENT

During the preparation of this manuscript, we used generative AI tools, including DeepSeek and ChatGPT, solely to assist with language polishing, grammatical correction, readability improvement, and the refinement of presentation and wording. Generative AI was not used to generate the core research ideas, methodological contributions, experimental results, data, or scientific claims of this work. In particular, all research hypotheses, model designs, experimental protocols, analyses, interpretations, and conclusions were developed and validated by the authors. Any AI-assisted suggestions were critically reviewed, verified, and revised by the authors before inclusion in the manuscript. The authors take full responsibility for the scientific content, accuracy, and integrity of the final submission.

REPRODUCIBILITY STATEMENT

Appendix A specifies the architecture, donor construction, loss definitions, gradient paths, and optimization settings. Appendix B documents video-disjoint data splits, candidate construction, frozen feature extraction, checkpoint selection, score calibration, evaluation metrics, and diagnostic sampling. The main controlled results report three training seeds with fixed splits and candidate pools; supplementary single-seed results are identified in Appendix C.

APPENDIX

A FULL METHOD AND TRAINING DETAILS

This appendix gives the implementation details, donor construction, and complete objectives underlying the compact formulation in Section 3.

A.1 ARCHITECTURE AND HISTORY REPRESENTATION

For a nonempty history, the anchor is the most recent valid clip embedding, $\mathbf{a}_i = \mathbf{x}(v_{i,\ell_i})$, and the aggregated history is $\overline{\mathbf{h}}_i = \ell_i^{-1} \sum_{k=1}^{\ell_i} \mathbf{x}(v_{i,k})$. A Boolean validity mask excludes padding from both the sum and denominator. The persistent encoder in Eq. 2 does not directly receive the instruction for nonempty histories. The observed and predicted transition pathways in Eq. 3 share the residual recomposer defined in Eq. 4; the recomposer always retains the receiver anchor, including when a donor transition is supplied. Each mapping g_p, g_o, g_t, and g_r is implemented as Linear–LayerNorm–GELU–Dropout–Linear, with hidden width 1024 and dropout rate 0.1. The factorization specifies the roles encouraged by training; it does not assume that architectural separation alone identifies independent semantic factors.

A.2 EMPTY-HISTORY CONVENTION

When $L = 0$, we set $\mathbf{a}_i = \mathbf{t}_i$ and $\overline{\mathbf{h}}_i = \mathbf{0} \in \mathbb{R}^{512}$. Here, $\mathbf{t}_i$ is the L_2-normalized CLIP text embedding of the current instruction q_i. For batch size b, the history input has shape $[b, 0, 512]$ and its Boolean validity mask has shape $[b, 0]$, with no valid history positions. The aggregation function handles this empty input directly by returning a zero tensor of shape $[b, 512]$, without dividing by a history count. Thus, the persistent encoder receives a text anchor concatenated with a zero history vector, giving $\mathbf{p}_i = \mathcal{N}(g_p([\mathbf{t}_i; \mathbf{0}]))$, and the same text anchor is used by the residual recomposer.

At $L = 0$, the text-anchor fallback gives $B_i(v) = A_i(v)$, so fusion reduces to $S_i(v) = (1 + w_v)A_i(v) + w_p C_i(v)$. The anchor score therefore supplies no independent visual-continuity signal in this condition. Prediction-only evaluation uses $C_i(v)$, while the equal-weight ensemble sets $w_v = w_p = 1$.

A.3 COMPATIBLE DONOR CONSTRUCTION

Transition records and donor pools are constructed separately within each data split. Donors must originate from a different video belonging to the same task. Matching considers both the preceding and current procedural steps, since agreement on the current action alone does not establish compatible initial conditions.

For CrossTask, we require identical task identifiers and identical ordered pairs of preceding and current canonical step identifiers. Adjacent segments with repeated step identifiers or overlapping temporal intervals are excluded before matching. For a receiver transition in a source video with N_v temporally ordered clips, let $\mathrm{position}(i)$ be its zero-based target-clip index; transition targets begin at index 1. The normalized target-step position is $r_i = \mathrm{position}(i)/\max(1, N_v - 1)$. Compatible donors are ranked by the difference in normalized target-step position, $|r_i - r_j|$, and assigned confidence $\omega_{ij} = 1 - |r_i - r_j|$. We retain at most 20 donors per receiver.

For YouCook2, we use semantic caption correspondences. We embed captions with `sentence-transformers/all-MiniLM-L6-v2` (Reimers & Gurevych, 2019; Wang et al., 2020), using attention-mask mean pooling followed by L_2 normalization, and compute cosine similarities between the resulting embeddings. Both the preceding-caption pair and the current-caption pair must be reciprocal neighbors in a top-100 semantic graph. Let s_{ij}^- and s_{ij}^+ denote the minimum of the two directional similarities for the preceding and current captions, respectively. Accepted pairs must satisfy semantic thresholds, preceding-caption lexical overlap, and a relative-position constraint. We additionally require

$$\min\left(s_{ij}^-, s_{ij}^+\right) \geq s_{ij}^{\times} + \delta_{\mathrm{dir}}, \tag{11}$$

where $s_{ij}^{\times}$ is the largest directed cross-aligned similarity between a preceding caption in one video and a current caption in the other. An unavailable cross-aligned score is conservatively bounded by the lowest similarity retained in its source neighbor list.

The confidence of an accepted YouCook2 donor pair is defined as

$$\omega_{ij} = 0.5 \ \mathrm{clip}_{[0,1]}\left(s_{ij}^{+}\right) + 0.3\, s_{ij}^{-} + 0.2\left(1 - |r_i - r_j|\right). \tag{12}$$

The current and preceding semantic thresholds are 0.75 and 0.70, respectively. For the Jaccard test, we lowercase each caption, extract tokens with the regular expression `[A-Za-z0-9]+`, and convert them to a set. For token sets T_a, T_b, we use $J(a,b) = |T_a \cap T_b| / \max(1, |T_a \cup T_b|)$. Punctuation separates tokens; we retain numbers and do not remove stopwords, stem, or lemmatize. The preceding-caption Jaccard threshold is 0.15, the direction margin is 0.15, and the maximum relative-position gap is 0.25. The 20 highest-confidence accepted donors are retained. These rules provide approximate procedural correspondence rather than an assumption of identical visual transformations.

A.4 Complete Training Objectives

Throughout this section, $\mathcal{B}$ denotes a receiver minibatch, $\mathcal{D} \subseteq \mathcal{B}$ its receivers with compatible donors, and $c(\cdot,\cdot)$ cosine similarity. The state-negative and identity-negative embedding sequences for receiver i are $\mathcal{Q}_i^{s}$ and $\mathcal{Q}_i^{\mathrm{id}}$, respectively.

A.4.1 Type-Aware Retrieval

For a predicted representation $\mathbf{y}$, target embedding $\mathbf{x}$, and negative embedding sequence $\mathcal{Q}$, we define

$$\ell_{\mathrm{neg}}\left(\mathbf{y}, \mathbf{x}; \mathcal{Q}\right) = -\log \frac{\exp\left(c(\mathbf{y},\mathbf{x})/\tau\right)}{\exp\left(c(\mathbf{y},\mathbf{x})/\tau\right) + \sum_{\mathbf{n}\in\mathcal{Q}} \exp\left(c(\mathbf{y},\mathbf{n})/\tau\right)}, \tag{13}$$

where $\tau = 0.07$ is the contrastive temperature. Padded negatives are excluded, and an empty negative sequence contributes zero.

For any nonempty receiver index set $\mathcal{I}$, the retrieval objective is

$$\begin{aligned} &\mathcal{R}_{\mathcal{I}}\left(\mathbf{y}; \mathcal{Q}^{s}, \mathcal{Q}^{\mathrm{id}}\right) \\ &= \frac{1}{|\mathcal{I}|} \sum_{i\in\mathcal{I}} \Bigg[-\log \frac{\exp\left(c(\mathbf{y}_i,\mathbf{x}_i)/\tau\right)}{\sum_{j\in\mathcal{I}} \exp\left(c(\mathbf{y}_i,\mathbf{x}_j)/\tau\right)} \\ &\qquad + \beta_s\, \ell_{\mathrm{neg}}\left(\mathbf{y}_i, \mathbf{x}_i; \mathcal{Q}_i^{s}\right) \\ &\qquad + \beta_{\mathrm{id}}\, \ell_{\mathrm{neg}}\left(\mathbf{y}_i, \mathbf{x}_i; \mathcal{Q}_i^{\mathrm{id}}\right) \Bigg]. \end{aligned} \tag{14}$$

The same functional supervises ordinary predictions over $\mathcal{B}$ and swapped predictions over $\mathcal{D}$, with their corresponding negative sequences. We use $\beta_s = 5.0$ and $\beta_{\mathrm{id}} = 1.0$.

A.4.2 Transition Prediction and Reconstruction

We apply ordinary retrieval supervision to the instruction-conditioned endpoint and introduce two auxiliary objectives:

$$\begin{aligned} \mathcal{L}_{\mathrm{ret}} &= \mathcal{R}_{\mathcal{B}}\left(\hat{\mathbf{x}}; \mathcal{Q}^{s}, \mathcal{Q}^{\mathrm{id}}\right), \\ \mathcal{L}_{\mathrm{pred}} &= \frac{1}{|\mathcal{B}|} \sum_{i\in\mathcal{B}} \left[1 - c\left(\hat{\mathbf{u}}_i, \mathbf{u}_i\right)\right], \\ \mathcal{L}_{\mathrm{rec}} &= \frac{1}{|\mathcal{B}|} \sum_{i\in\mathcal{B}} \left[1 - c\left(\mathbf{x}_i^{\mathrm{rec}}, \mathbf{x}_i\right)\right]. \end{aligned} \tag{15}$$

Prediction alignment connects the inference pathway to the observed-transition representation. Reconstruction encourages the observed code and receiver context to retain information needed to recover the target embedding. Both losses include receivers without matched donors.

A.4.3 Swap Supervision and Confidence Scaling

The matched recomposition is defined in Eq. 5 and its receiver-target supervision in Eq. 8. Let $\mathcal{B}$ denote the receiver minibatch and $\mathcal{D} \subseteq \mathcal{B}$ the receivers with compatible donors. For $i \in \mathcal{D}$, the identity-negative sequence is augmented as

$$\mathcal{Q}_i^{\mathrm{id,swap}} = \mathcal{Q}_i^{\mathrm{id}} \parallel [\mathbf{x}_{d(i)}], \qquad \overline{\omega}_{\mathcal{D}} = \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} \omega_i, \quad \omega_i = \omega_{i,d(i)}.$$

The symbol $\parallel$ denotes concatenation and preserves multiplicity if the donor endpoint already appears among the negatives. The positive endpoint remains $\mathbf{x}_i$. Explicitly contrasting it with the donor endpoint discourages a recomposition that simply retrieves donor appearance. The loss is averaged over donor-valid receivers and then multiplied by their mean confidence. Thus, confidence scales the minibatch swap loss rather than weighting receiver–donor pairs individually within that minibatch. Its in-batch contrastive denominator likewise contains only receiver targets indexed by $\mathcal{D}$. We define the loss as zero when no donor is available.

A.4.4 Wrong-Transition Control

For receivers with an available wrong-step donor $w(i)$, we construct

$$\begin{aligned} \tilde{\mathbf{x}}_i^- &= R_\theta \left(\mathbf{p}_i, \mathbf{u}_{w(i)}, \mathbf{a}_i\right), \\ \mathcal{L}_{\mathrm{wrong}} &= \frac{\sum_{i \in \mathcal{W}} \omega_i \left[m - c\left(\tilde{\mathbf{x}}_i, \mathbf{x}_i\right) + c\left(\tilde{\mathbf{x}}_i^-, \mathbf{x}_i\right)\right]_+}{\sum_{i \in \mathcal{W}} \omega_i}. \end{aligned} \tag{16}$$

Here, $[z]_+ = \max(0, z)$, and $\mathcal{W} \subseteq \mathcal{D}$ contains receivers with wrong-step donors. We use the same margin $m = 0.2$ here and in the relational losses below. A zero total weight produces zero loss.

Wrong-step donors come from another video in the same task and exclude accepted matches. They have a different current canonical step in CrossTask, or current-caption Jaccard similarity below 0.25 in YouCook2. For YouCook2, lexical mismatch is an approximate control and does not guarantee a different semantic transition. Because matched and wrong recompositions share the receiver code and anchor, Eq. 16 directly supervises sensitivity to the supplied transition.

Donor and control records are selected reproducibly using the random seed, epoch, receiver identifier, and relation type. Receivers without donors remain in training, with unavailable swapping terms masked.

A.4.5 Relational Factorization

For receiver i, let $s(i) \neq i$ index another transition record from the same video, selected from the dedicated transition pool. The only exclusion is transition i itself: adjacent transitions and those with the same action or step semantics remain eligible. We select one record deterministically using a hash of the training seed, epoch, receiver transition ID, and relation type. Let $\mathcal{P} \subseteq \mathcal{D} \subseteq \mathcal{B}$ denote the receiver indices for which both a same-video record $s(i)$ and a compatible cross-video donor $d(i)$ are available. Each $i \in \mathcal{P}$ therefore defines a relational triplet $(i, s(i), d(i))$, weighted by the donor confidence $\omega_i = \omega_{i,d(i)}$ defined by the dataset-specific matching rules in Appendix A.3. Receivers outside $\mathcal{P}$ do not contribute to the two relational losses below.

For an index set $\mathcal{I}$ with confidence weights ω_i, define

$$\mathcal{A}_{\mathcal{I}}\left[z_i\right] = \frac{\sum_{i \in \mathcal{I}} \omega_i z_i}{\sum_{i \in \mathcal{I}} \omega_i}, \tag{17}$$

with value zero when the denominator is zero.

The relational objectives are

$$\begin{aligned}\mathcal{L}_p &= \mathcal{A}_{\mathcal{P}}\left[\left[m - c\left(\mathbf{p}_i, \mathbf{p}_{s(i)}\right) + c\left(\mathbf{p}_i, \mathbf{p}_{d(i)}\right)\right]_+\right], \\ \mathcal{L}_u &= \mathcal{A}_{\mathcal{P}}\left[\left[m - c\left(\mathbf{u}_i, \mathbf{u}_{d(i)}\right) + c\left(\mathbf{u}_i, \mathbf{u}_{s(i)}\right)\right]_+\right].\end{aligned} \tag{18}$$

The persistent objective favors contextual similarity within a video, while the transition objective favors similarity between procedurally matched videos. These constraints are constructed from dedicated transition pools rather than directly treating every retrieval identity negative as a transition positive. They provide weak relational regularization: different records from the same video may still contain semantically similar procedures.

A.5 Loss Activation, Gradients, and Optimization

For training epoch e, the activation function in Eq. 9 is

$$h(e) = \begin{cases} 0, & e \leq 2, \\ 1, & e > 2. \end{cases}$$

The first two epochs therefore optimize ordinary retrieval, transition prediction, and reconstruction before activating relational, swapping, and wrong-transition supervision. This schedule controls loss activation rather than learning-rate warmup; the learning rate remains fixed throughout training.

The backbone features remain frozen, while all learned branches are optimized jointly. The observed transition code is not detached during prediction alignment. Swap gradients propagate through the receiver persistent encoder, the donor observed-transition encoder, and the shared recomposer.

The loss weights are set to

$$(\lambda_{\text{pred}}, \lambda_{\text{rec}}, \lambda_p, \lambda_u, \lambda_{\text{swap}}, \lambda_{\text{wrong}}) = (1,\ 1,\ 0.5,\ 0.5,\ 1,\ 0.5).$$

All models are trained for 20 epochs with a batch size of 512 using AdamW, a fixed learning rate of 10^{-4}, weight decay of 10^{-3}, and gradient-norm clipping at 1.0. We use no learning-rate schedule. Checkpoint and score-fusion selection are described in Appendix B.4.

B Data, Evaluation, and Reproducibility

B.1 Data Splits and Candidate Construction

We evaluate SwapCAST on YouCook2 Zhou et al. (2018) and CrossTask Zhukov et al. (2019), two procedural video datasets. Following the Consistent Video Retrieval (CVR) formulation introduced by CAST Liu et al. (2026), each query consists of an observed visual history and a textual instruction, and the model ranks a candidate set to identify the appropriate continuation. The evaluation therefore requires both instruction alignment and consistency with the preceding visual context.

For YouCook2, we split the original training videos into training and calibration subsets at an approximately 90%/10% ratio and use the original validation set for evaluation. The resulting training, calibration, and evaluation subsets contain 6,565, 690, and 2,594 queries, respectively. For CrossTask, we partition videos within each of the 18 tasks into training, calibration, and evaluation subsets at an approximately 72%/8%/20% ratio, yielding 9,268, 1,059, and 2,515 queries, respectively. Both datasets use a fixed split seed of 1337, and the three subsets are disjoint at the video level.

Each query has a fixed pool of 10 candidates: one annotated continuation and nine negatives, typically comprising three state negatives, three identity negatives, and three easy negatives. When a negative category has insufficient candidates, the remaining slots are backfilled to maintain the 10-candidate pool. State negatives are drawn from other steps in the same video. Identity negatives are drawn from other videos using text similarity computed with `all-MiniLM-L6-v2` (Reimers

& Gurevych, 2019; Wang et al., 2020) for YouCook2, and matching task and step identifiers for CrossTask. All candidates are drawn from the same data split as the query.

Evidence about the contributions of the training objectives is obtained from controlled comparisons among SwapCAST variants evaluated under the same reconstructed CVR protocol. All variants share the same candidate construction, frozen features, architecture, inference procedure, and evaluation metrics unless explicitly stated otherwise. This controlled setting isolates the effects of the training objectives within the shared factorized architecture.

B.2 Feature Extraction and Training Configuration

All SwapCAST variants use frozen CLIP ViT-B/32 features Radford et al. (2021). For each clip, we sample three frames at 25%, 50%, and 75% of its duration. We L_2-normalize each frame embedding, average the three embeddings, and normalize the resulting clip representation again. Instructions are encoded using the CLIP text encoder. Both visual and textual features are cached as 512-dimensional vectors.

Models use at most $L = 5$ history clips. The full loss weights, activation schedule, and gradient paths are specified in Appendix A.5. For the history-length analysis, available context is truncated only at inference time; all evaluated models are trained with $L = 5$.

B.3 External Reference Methods

For external context, we additionally report reference results published in CAST Liu et al. (2026). These include several strategies for combining textual and visual context.

Late Fusion (Heuristic) scores each candidate using a fixed combination of instruction similarity and visual continuity:

$$S_{\mathrm{LF},i}(v) = \langle \mathbf{t}_i, \mathbf{x}(v) \rangle + \alpha \langle \mathbf{a}_i, \mathbf{x}(v) \rangle , \tag{19}$$

where $\mathbf{t}_i$ is the instruction representation, $\mathbf{a}_i$ is the most recent visual anchor, and $\mathbf{x}(v)$ is the frozen candidate embedding. The coefficient α controls the contribution of visual continuity.

Late Fusion (Learned) learns the relative weighting of semantic and visual similarity scores using a lightweight MLP. **Early Fusion** combines textual and visual-context features before predicting the continuation representation. **CAST** predicts a context-conditioned residual update to the visual anchor.

All values taken directly from the CAST paper are explicitly marked as **"previously published reference results."** Because these results were obtained under the evaluation protocol reported in the original work, they are used only as external reference points and are not treated as head-to-head evaluations on our reconstructed candidate pools. Accordingly, conclusions about the contribution of SwapCAST are based primarily on controlled comparisons among variants evaluated under our common reconstructed protocol.

B.4 Checkpoint Selection and Score Calibration

Ordinary SwapCAST inference uses only the receiver history and textual instruction. The persistent representation $\mathbf{p}_i$ and predicted transition $\hat{\mathbf{u}}_i$ are used to produce the predicted continuation representation $\hat{\mathbf{x}}_i$. Observed target transitions and donor transition codes are not available to this inference pathway.

Our primary evaluation, including the primary retrieval evaluation reported alongside published CAST reference values, uses the calibrated fusion score $S_i(v)$ defined in Eq. 10. Here, $A_i(v)$ measures instruction–candidate similarity, $B_i(v)$ measures anchor–candidate similarity (visual continuity when history is available, and $B_i(v) = A_i(v)$ under the $L = 0$ text-anchor fallback), and $C_i(v)$ measures similarity to the predicted continuation representation.

We first select the checkpoint with the highest calibration top-1 accuracy under prediction-only scoring, $C_i(v)$. Using this checkpoint, we then search all 36 combinations of $w_v, w_p \in \{0, 0.25, 0.5, 1, 2, 4\}$, with the coefficient of $A_i(v)$ fixed at 1. The pair with the highest calibration Top-1 accuracy under the fused score is then fixed, and the selected checkpoint and weights are

used to report the calibrated-fusion main results on the evaluation split. Evaluation-set scores are not used to select either the checkpoint or the fusion coefficients. Fusion calibration is performed independently for each training seed on the same fixed calibration split; weights are not shared across seeds. Table 5 reports the selected coefficients for the complete SwapCAST model. Each seed is evaluated using its own selected checkpoint and weight pair, and the calibrated-fusion mean and sample SD are computed over these three evaluation results.

Table 5: Selected fusion weights (w_v, w_p) for the complete SwapCAST model, calibrated independently for each training seed. The instruction score coefficient is fixed at 1.

Training seed	YouCook2 (w_v, w_p)	CrossTask (w_v, w_p)
1337	$(0.5, 0.5)$	$(0, 2.0)$
2027	$(0.5, 2.0)$	$(0.5, 4.0)$
3407	$(0.5, 1.0)$	$(0.25, 1.0)$

Prediction-only scoring, $C_i(v)$, is used for controlled training-objective ablations and analyses of the prediction branch. The scoring ablation additionally compares it with the fixed equal-weight combination $A_i(v) + B_i(v) + C_i(v)$ and calibrated fusion to quantify the effect of inference-time scoring. The use of C-only Top-1 for checkpoint selection does not change the calibrated-fusion scoring used for the primary evaluation.

Experiments that use observed transition information are reported separately as mechanism diagnostics and are not included in the ordinary inference results.

B.5 Evaluation Metrics and Repeated Runs

We report top-1 retrieval accuracy (Acc., equivalent to Recall@1) and mean target rank (MnR). Higher accuracy and lower MnR indicate better retrieval performance.

Following CAST Liu et al. (2026), we additionally report State Accuracy and Identity Accuracy. State Accuracy is the proportion of eligible queries for which the target is scored above every state distractor, whereas Identity Accuracy measures the proportion for which the target is scored above every identity distractor. Each diagnostic is averaged only over queries containing the corresponding negative category. All accuracy-based metrics are reported as percentages.

Previously published diagnostic results are reported only when a corresponding dataset-specific value is available. In particular, diagnostic averages reported by CAST across three datasets are not treated as directly comparable to averages over the two datasets evaluated in our reconstructed setting.

For the main results and six principal controlled ablation variants, we report the mean and sample standard deviation over three predefined training seeds: 1337, 2027, and 3407. The data split seed remains fixed at 1337 across these training runs, and all runs use the same evaluation protocol. Previously published reference results retain the reporting format of their original source.

B.6 Observed-Transition Diagnostic Protocol

We next directly probe whether the learned transition representation can be applied to a receiver context different from the video from which the transition was observed. For these diagnostics, the receiver persistent representation and anchor are held fixed, while the transition input to the shared recomposer is replaced by different observed or control codes. Unlike ordinary inference, the donor transition code is computed from an annotated donor clip pair. The receiver-observed condition additionally accesses the annotated receiver target and therefore serves only as a target-informed reconstruction reference.

Table 10 summarizes the results. We evaluate checkpoints trained with seeds 1337, 2027, and 3407; the sampling seed for random controls is fixed at 8675309. The compatible-donor diagnostic set contains 153 unique receiver queries and 170 receiver–compatible-donor pairings on YouCook2, and $2{,}327$ unique receiver queries and $23{,}255$ pairings on CrossTask. For each diagnostic condition, we compute Top-1 hits in the fixed candidate pool separately for each receiver–donor pairing. We first average over donor pairings for each receiver, then average equally over receivers, and finally

average equally over the three model seeds and convert the result to a percentage. Thus, receivers have equal weight regardless of their number of donor pairings. We neither pool all pairings into a single unweighted mean nor fuse multiple donors before retrieval.

C Extended Quantitative Results

C.1 Additional Training-Objective Diagnostics

We report two supplementary experiments using the full SwapCAST configuration and the training and evaluation protocol in Section 4.1. All results use a single training seed (1337) and ordinary prediction-only (C-only) retrieval, without donor information at inference time. The values below are individual-run accuracies rather than the three-seed means reported in the main ablation study.

C.1.1 Individual Loss Deletions

Table 6 removes one auxiliary objective at a time from the full model while retaining the shared factorized architecture and the remaining objectives. Removing any of the five evaluated objectives lowers accuracy on both datasets relative to the full-model result for the same seed. These comparisons diagnose the role of each objective in the full training configuration; a single seed does not establish an average effect across runs.

Table 6: Individual loss deletions from full SwapCAST. Entries are prediction-only retrieval accuracy (%) for seed 1337. $\mathcal{L}_p$ and $\mathcal{L}_u$ are the persistent-context and transition relational objectives, respectively.

Training configuration	YouCook2 ↑	CrossTask ↑
Full SwapCAST	53.89	48.93
Without transition prediction ($\mathcal{L}_{\text{pred}}$)	51.00	45.40
Without target reconstruction ($\mathcal{L}_{\text{rec}}$)	51.12	46.36
Without persistent relation ($\mathcal{L}_p$)	52.59	46.04
Without transition relation ($\mathcal{L}_u$)	52.82	47.44
Without wrong-transition constraint ($\mathcal{L}_{\text{wrong}}$)	53.08	48.20

C.1.2 Swap-Loss Weight Sensitivity

Table 7 varies only the swap-retrieval loss weight λ_{swap}, keeping all other full-model loss weights fixed. The response is non-monotonic on both datasets: the highest observed accuracy occurs at 0.1 on YouCook2 and 1.0 on CrossTask. This single-seed sweep is exploratory and is not used to select the main model from evaluation-set results. The main experiments retain the predefined value $\lambda_{\text{swap}} = 1.0$.

Table 7: Sensitivity to the swap-retrieval loss weight. Entries are prediction-only retrieval accuracy (%) for seed 1337; all other loss weights retain their full-model values.

λ_{swap}	YouCook2 ↑	CrossTask ↑
0.05	53.97	47.99
0.1	54.32	47.32
0.5	52.58	46.56
1.0 (default)	53.89	48.93
2.0	53.24	45.61

C.2 Inference Scoring

The preceding training-objective ablations use prediction-only scoring to assess the learned continuation representation. We next examine how this representation interacts with the auxiliary semantic and visual-continuity scores at inference time. Table 8 compares prediction-only scoring, the fixed equal-weight combination, and calibrated fusion.

Table 8: Ablation of inference scoring for the complete SwapCAST model. A, B, and C denote instruction matching, visual continuity, and predicted-target similarity, respectively. Fusion coefficients are selected independently for each training seed using only calibration data (Table 5). Results are mean accuracy (%) $\pm$ sample standard deviation over three seeds.

Inference Strategy	YouCook2 ↑	CrossTask ↑
Prediction only: C	54.16 ± 0.59	48.96 ± 0.44
Fixed fusion: $A + B + C$	58.27 ± 0.82	48.22 ± 0.61
Calibrated fusion: $A + w_v B + w_p C$	$\mathbf{58.53 \pm 0.29}$	$\mathbf{49.30 \pm 0.08}$

On YouCook2, adding the fixed combination $A_i(v)+B_i(v)+C_i(v)$ improves accuracy from 54.16% to 58.27%, and calibration provides a further increase to 58.53%. CrossTask exhibits a different pattern: fixed fusion decreases accuracy from 48.96% to 48.22%, whereas calibrated weighting recovers and slightly improves performance to 49.30%. Thus, auxiliary semantic and visual-continuity signals are not uniformly beneficial under equal weighting; their utility depends on the dataset and their relative contribution.

Notably, calibrated fusion improves over prediction-only scoring by 4.37 percentage points on YouCook2 but only 0.34 points on CrossTask. The small gap on CrossTask reinforces the observation that the predicted continuation representation itself already carries most of the retrieval signal in this setting. More broadly, these results indicate that the performance of SwapCAST is not solely attributable to score fusion: the core prediction pathway remains strong when evaluated independently.

C.3 Dependence on Visual History

Finally, we examine whether the gains of SwapCAST depend on the visual context available to the receiver. Both the base model M_1 and complete SwapCAST M_5 are trained with a maximum history length of $L = 5$. At evaluation time, we truncate the available history to $L \in \{0, 1, 3, 5\}$ without retraining the models. At $L = 0$, we use the instruction's CLIP text embedding as the anchor and set the aggregated history to zero, with an empty Boolean history mask, as defined in Appendix A.2. This condition removes visual history but retains instruction conditioning through both the text anchor and the transition predictor. All history-length comparisons use prediction-only scoring, $C_i(v)$. The results are shown in Fig. 4.

Removing the history almost eliminates the benefit of the complete model. At $L = 0$, SwapCAST achieves 23.72% accuracy on YouCook2 and 16.22% on CrossTask, compared with 23.64% and 15.83% for the base model. The corresponding margins are only 0.08 and 0.39 percentage points. Once the immediately preceding clip becomes available, however, SwapCAST reaches 52.42% on YouCook2 and 47.32% on CrossTask, outperforming the base model by 3.01 and 5.28 points, respectively.

Increasing the available history from one to five clips yields further gains for the complete model, from 52.42% to 54.16% on YouCook2 and from 47.32% to 48.96% on CrossTask. The largest recovery therefore occurs when moving from no visual history to the immediate predecessor, while longer context provides smaller additional benefits.

The key observation is not only the sharp recovery obtained by restoring visual history, but also the changing gap between the two models. At $L = 0$, SwapCAST exceeds the factorized base by only 0.08 percentage points on YouCook2 and 0.39 points on CrossTask. Once the immediate predecessor is available ($L = 1$), these margins increase to 3.01 and 5.28 points, respectively. The advantage remains substantial as more history is restored. The full model's advantage over the base is larger when receiver visual context is available. This comparison jointly adds swapping, wrong-transition supervision, and relational regularization, so it does not isolate the dependence of each objective on history.

Because all models are trained with $L = 5$, this experiment measures robustness to inference-time history truncation. It should not be interpreted as a comparison between separately trained models with different history lengths.

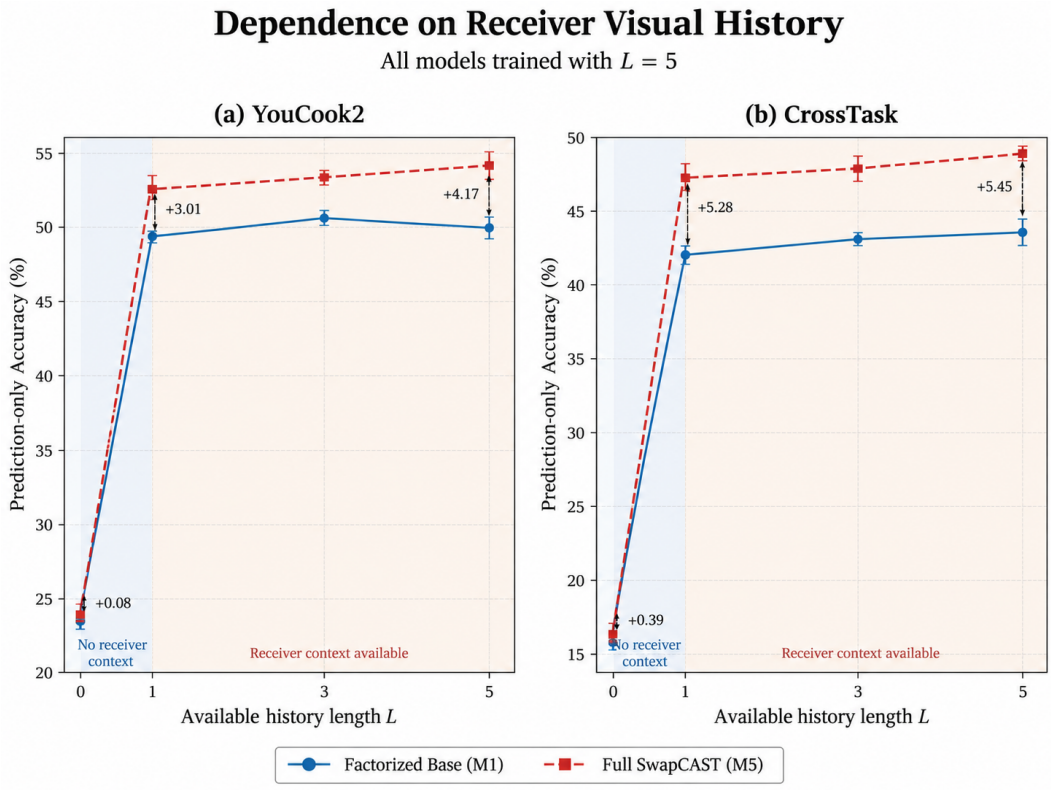

Figure 4: Dependence on the visual history available at inference time. Prediction-only retrieval accuracy is shown for the factorized base model M_1 and complete SwapCAST M_5 on YouCook2 and CrossTask. All models are trained with a maximum history length of $L = 5$; only the receiver history available during inference is truncated. Points and error bars denote means and sample standard deviations over three training seeds. The performance gap between M_1 and M_5 is small without receiver history ($L = 0$) and becomes substantially larger once visual context is available.

C.4 Training-Time Donor Interventions

Training-time interventions modify donor supervision and evaluate ordinary prediction-only retrieval, without donor information at inference. Observed-transition diagnostics in Appendix C.5 instead supply annotated transition endpoints to probe the learned mechanism.

We first intervene on the donor transitions used by the swap-retrieval objective. Table 9 reports results for the base-plus-swap configuration M_3, with relational regularization and the wrong-transition constraint disabled. These comparisons probe changes within the complete swap objective; they do not independently estimate the benefits of transfer and donor negatives.

Replacing compatible donors with randomly selected cross-video transitions from the same task reduces prediction-only accuracy from 52.38% to 50.31% on YouCook2 and from 47.40% to 42.56% on CrossTask, corresponding to decreases of 2.07 and 4.84 percentage points, respectively. The intervention preserves receiver coverage and the original confidence weights, so the observed degradation cannot be explained simply by exposing fewer receivers to swap supervision. This result supports the importance of donor selection, but changing donors can also change the donor-negative endpoints and their difficulty; it does not isolate procedural compatibility from these effects.

A complementary control changes donor endpoint order. Reversing the same donor transition decreases accuracy by 0.60 points on YouCook2 and 1.08 points on CrossTask. Because the donor-target discrimination endpoint remains unchanged in this intervention, the comparison specifically tests whether the direction of the observed change influences the usefulness of the supervision. The

Table 9: Training-time donor interventions. All variants use the base-plus-swap configuration M_3 and are evaluated with ordinary prediction-only retrieval. Relational and wrong-transition losses are disabled. Results are accuracy (%) for seed 1337.

Training Configuration	YouCook2 ↑	CrossTask ↑
Compatible donor transitions	**52.38**	**47.40**
Random same-task cross-video donors	50.31	42.56
Reversed donor transitions	51.78	46.32
Without donor-target discrimination	51.62	46.36

decreases observed for seed 1337 on both datasets are consistent with sensitivity to donor endpoint order. They do not rule out other donor-specific cues.

Finally, removing the donor endpoint from the identity-negative set decreases accuracy by 0.76 points on YouCook2 and 1.04 points on CrossTask. This result is consistent with the intended role of donor-target discrimination: the recomposed representation should match the receiver continuation rather than inherit the visual identity of the donor. Since these interventions are evaluated for a single training seed, we treat them as mechanism-level supporting evidence rather than estimates of average effect size across runs. The negative-removal comparison measures the contribution of donor-target discrimination conditional on swap recomposition, not the effect of adding that negative alone to the base. It therefore cannot be subtracted from the three-seed $M_1 \rightarrow M_3$ gain to estimate an independent transfer effect; the two components may interact.

C.5 Complete Observed-Transition Diagnostics

We hold the receiver persistent code and anchor fixed while varying the transition input. The receiver-observed condition uses the annotated receiver target and is a target-informed reconstruction reference, not an ordinary inference result. Appendix B.6 gives the sampling and aggregation protocol; Table 10 reports all controls for the base model M_1 and full model M_5.

When supplied with compatible donor transitions, the full SwapCAST model achieves 48.33% accuracy on YouCook2, compared with 40.42% for the factorized base model. On CrossTask, the corresponding values are 43.56% and 36.53%. The full model exceeds the base by 7.91 and 7.03 percentage points on the compatible-donor diagnostic subsets, respectively. These results support cross-video recomposition under supplied annotated transitions and reflect the combined effect of the additional objectives.

The identity of the supplied transition is also consequential. For full SwapCAST on YouCook2, compatible donor codes achieve 48.33% accuracy, whereas wrong-transition and reversed-transition codes obtain only 27.50% and 25.83%, respectively. On CrossTask, the corresponding values are 43.56%, 34.10%, and 35.99%. Compatible donor codes therefore outperform both sampled wrong-transition and direction-reversed codes on both datasets. The wrong-transition pool is defined by the control criteria in Appendix A.4.4; its members are not guaranteed to be semantically inappropriate in every case.

A similar pattern is observed relative to random same-task cross-video codes. Full SwapCAST reaches 48.33% with compatible donors versus 37.08% with random same-task donors on YouCook2, and 43.56% versus 35.23% on CrossTask. Taken together, these controls indicate that the learned recomposition is sensitive not only to the presence of a nonzero transition code, but also to its procedural compatibility and temporal direction.

Learned Transition Codes vs. Direct Transfer. Several controls help distinguish learned transition recomposition from simpler alternatives. For full SwapCAST, replacing a compatible donor transition with the zero code reduces accuracy from 48.33% to 30.00% on YouCook2 and from 43.56% to 35.71% on CrossTask. Using the mean observed training code similarly yields only 32.50% and 36.06%. Thus, the benefit of the compatible donor input cannot be explained solely by the presence of a generic transition-like offset.

Direct transfer operations are substantially weaker. Copying the donor target itself achieves only 5.00% on YouCook2 and 12.22% on CrossTask, confirming that simply retrieving toward the donor endpoint does not solve the receiver task. Likewise, directly transferring the raw donor feature dif-

Table 10: Observed-transition diagnostics. "Base" denotes M_1 and "Full" denotes M_5. Observed transition codes are computed from annotated clip endpoints and are used only for mechanism analysis. The receiver-observed condition additionally accesses the receiver target. Values are Top-1 hit rates (%) in fixed candidate pools, averaged over donor pairings within each receiver, equally over receivers, and equally over model seeds 1337, 2027, and 3407. Random-control sampling uses seed 8675309. The diagnostic sets contain 153 unique receiver queries/170 compatible pairings on YouCook2 and 2,327 unique receiver queries/23,255 compatible pairings on CrossTask. Bold indicates the highest value in each column, including the target-informed receiver-observed reference. These diagnostics are distinct from ordinary deployment results.

Diagnostic Input	YC Base	YC Full	CT Base	CT Full
Instruction-predicted code	35.83	38.33	40.91	46.41
Receiver observed code	**73.33**	**79.17**	**62.94**	**57.97**
Compatible donor code	40.42	48.33	36.53	43.56
Wrong-transition code	27.50	27.50	30.10	34.10
Reversed donor code	26.67	25.83	30.51	35.99
Random same-task donor code	27.50	37.08	30.99	35.23
Random cross-video code	20.83	23.33	30.71	36.16
Zero code	26.67	30.00	33.38	35.71
Mean observed training code	30.00	32.50	33.51	36.06
Receiver last history frame	30.00	30.00	35.54	35.54
Direct donor-target copying	5.00	5.00	12.22	12.22
Raw feature-difference transfer	21.25	21.25	37.04	37.04

ference reaches 21.25% and 37.04%. The learned compatible-code recomposition exceeds these values by 27.08 and 6.52 percentage points, respectively. Within this diagnostic setting, these comparisons support the role of the learned transition representation and shared recomposer over direct feature-space transfer.

The diagnostics also reveal an important limitation. The receiver-observed reconstruction condition improves from 73.33% to 79.17% on YouCook2 but decreases from 62.94% to 57.97% on CrossTask when moving from the base model to full SwapCAST. Improved cross-video recomposition therefore does not uniformly improve target-informed self-reconstruction. The cross-video recomposition gains do not establish complete disentanglement between persistent visual context and procedural change.

D Qualitative Recomposition Diagnostics

D.1 Observed-Transition Recomposition

Figure 5 presents selected examples from the observed-transition analysis. For each receiver, we keep the visual context, instruction, and candidate pool fixed while varying only the transition code supplied to the recomposer. This comparison shows how retrieval changes when a compatible transition is reversed or replaced with a sampled wrong-step transition.

The examples illustrate successful recomposition with compatible donor transitions and the different retrieval outcomes obtained when those transitions are reversed or replaced with wrong-step transitions. They complement the quantitative comparisons in Table 10. They do not imply perfect factorization: the aggregate results in Table 10 show that compatible donor codes do not always retrieve the receiver continuation. All donor observations shown in Fig. 5 are used only for mechanism analysis and are never available during ordinary inference.

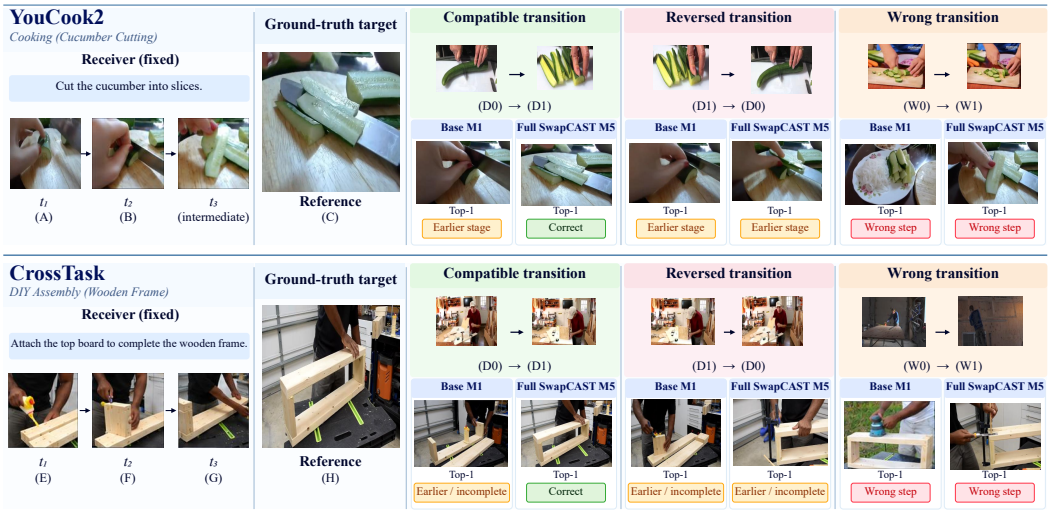

Figure 5: Qualitative diagnostics of cross-video transition recomposition. For each fixed receiver context and candidate pool, compatible and wrong-step transition codes are encoded from annotated clip pairs in other videos of the same task. The reversed code uses the compatible donor pair in reverse order. We compare the resulting Top-1 retrievals of the factorized base model M_1 and full SwapCAST M_5. The ground-truth receiver target is shown only as an evaluation reference. Observed donor codes are used only for this mechanism diagnostic; ordinary SwapCAST inference uses only the receiver history and instruction.