INTERVENABLE CONCEPT DYNAMICS FOR VIDEO ACTIVITY FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Concept bottleneck models route predictions through human-understandable concepts, making their intermediate states inspectable and editable. Extending this model family to forecasting requires modeling how concept activations evolve over time as objects, actions, and their interactions change, yet existing CBMs ground concepts only in observed inputs and do not forecast their future evolution. We study this problem using videos of ongoing activities and forecast future activities through a named-concept state. We propose **TRACE**-CBM (Temporal Relational Activity Concept Evolution), which first refines observed concept activations with sparse spatio-temporal graphs. It then recursively predicts future concept states, conditioning each transition on the prior activity distribution, and decodes both observed and predicted states with a shared activity classifier. At inference time, the model supports editing concept values, activity activations, and learned edges without parameter updates, exposing the relational spatio-temporal dependencies used for forecasting. Across multiple datasets, TRACE improves observed-activity Top-1 accuracy over concept-based methods by 1.2–8.5 percentage points and attains H3 forecast Top-1 accuracy within 1.9–2.2 points of matched black-box forecasters. Intervention experiments show concept-node edits correct 25.6–34.4% of initially incorrect H3 forecasts without flipping initially correct ones, and concept steering achieves the requested target activity in up to 90.0% of cases.

1 INTRODUCTION

Concept-based forecasting needs a semantic interface. Concept bottleneck models (CBMs) route predictions through human-understandable intermediate concepts (Koh et al., 2020). Therefore, these models expose the semantic evidence behind a prediction that supports inspection and targeted model intervention (Yuksekgonul et al., 2022; Yang et al., 2023; Steinmann et al., 2025; Knab et al., 2026b). However, CBMs typically encode concepts as a static representation of an observed input, without modeling their evolution over time (Oikarinen et al., 2023; Prasse et al., 2025). Forecasting requires modeling this evolution, as future predictions can also depend on the future semantic state. If this evolution takes place in an opaque latent space, the semantic interface is lost, making it difficult to debug forecast errors or steer predictions at inference time (De Felice et al., 2026). An

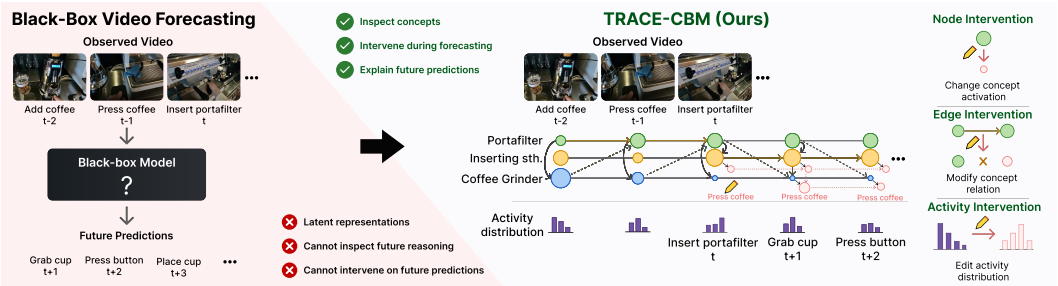

Figure 1: **Latent vs. concept-space forecasting.** Conventional black-box models roll out opaque latents. TRACE-CBM rolls out editable concept states, enabling persistent edits to concepts, edges, and activity beliefs.

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.

intervenable CBM for forecasting should therefore not only expose its current concept state, but preserve a meaningful and editable concept state throughout the prediction horizon.

Videos are challenging for persistent concept forecasting. Activity recognition requires models not only to recognize observed video content, but also to anticipate future activities (Feichtenhofer et al., 2019; Bertasius et al., 2021). To do this, strong video forecasting models commonly propagate latent features whose future evolution cannot be directly inspected or controlled (Liu et al., 2022; Girdhar & Grauman, 2021; Gong et al., 2022). We study whether an explicit concept interface can be maintained while semantically aligned concepts are recursively forecast into the future. In contrast, existing video CBMs learn semantically-isolated concepts for video classification (Lee et al., 2025; Knab et al., 2026a), but do not forecast future concept states or define how interventions propagate over time. Black-box video forecasters can predict future activities, but typically evolve opaque latent representations without a named, editable state (Assran et al., 2023) (see Figure 1). Consequently, no existing approaches provide mechanisms to provide interpretable forecasts nor have a clear mechanism to correct future concept states and thus steer future activity corrections.

Intervenable concept-space forecasting. To address this limitation, we propose **TRACE** (Temporal Relational Activity Concept Evolution), an autoregressive concept bottleneck model for video activity forecasting that recursively updates named concept states, rather than opaque latent representations (see Figure 1). Our method learns spatio-temporal graphs on concept states that refine observed concept activations and represent learned within-window (spatial) and cross-window (spatio-temporal) dependencies. We use a *graph* because its *explicit relation-based propagation* lets an edit to one concept propagate to other concepts, rather than treating concept edits in isolation. The model then recursively predicts future concept states and conditions each transition on the activity distribution predicted at the previous window. A shared concept-to-activity classifier decodes both observed and forecasted states on the same semantic space. This design allows for model edits on concept nodes, activity beliefs, and learned edges without changing model weights. Because the architecture exposes editable concept states during forecasting, we evaluate whether these edits can correct or steer future activity predictions. Our main contributions are as follows:

- We introduce TRACE, an autoregressive CBM for video forecasting that predicts human-readable concept states with sparse spatio-temporal graphs and activity-conditioned transitions (Section 3).
- Because forecast states remain in the same concept space as observed states, interventions on concepts, activity beliefs, and learned relations can be *traced* and propagated over time, extending CBM interventions to future states without retraining (Section 3).
- We show experimentally that graph-structured dynamics preserve localized, inspectable concept contributions, and quantify how edits to concepts, activity beliefs, and relations propagate through the forecasting process (Section 4).

More broadly, TRACE contributes to the development of video forecasting systems that are not only accurate but also transparent and controllable.

2 Related Work

Concept bottleneck models. Concept bottleneck models factor prediction into an input-to-concept encoder and a concept-to-target decoder, such that a named concept vector controls the task decision (Koh et al., 2020; Espinosa Zarlenga et al., 2022; Yuksekgonul et al., 2022; Knab et al., 2026b). Subsequent work reduces manual concept supervision through label-free concepts (Oikarinen et al., 2023), improves concept construction using language guidance (Yang et al., 2023), and explores data-efficient (Prasse et al., 2025) and object-centric representations (Steinmann et al., 2025). Beyond interpretability, concepts provide an interface for a targeted model intervention. For example, Shin et al. (2023) study how correction gains depend on the concept selection and the intervention budget, while Espinosa Zarlenga et al. (2023) explicitly train concept embedding models for sequences of human corrections. However, these interventions primarily modify concepts underlying a current prediction. Standard CBM factorization does not specify how an edit should propagate through recursively predicted future concept states. Concept representations can also retain target-relevant information beyond their intended semantics (Havasi et al., 2022), motivating leakage-aware construction and evaluation.

Temporal modeling and forecasting. Modern video recognition and anticipation models capture temporal structure through convolutional or attention-based representations, including SlowFast (Feichtenhofer et al., 2019) and TimeSformer (Bertasius et al., 2021). Anticipative Video Transformer (Girdhar & Grauman, 2021) and FUTR (Gong et al., 2022) extend such representations toward future activity prediction from partial observations. While effective for temporal prediction, their propagated states do not provide a named semantic interface for inspection or intervention (De Felice et al., 2026). In contrast, recent video CBMs have begun to incorporate temporal concept representations. DANCE (Lee et al., 2025) separates explanatory factors, while Knab et al. (2026a) preserve temporally ordered concept representations with MoTIF. These methods support concept-based temporal video understanding but do not forecast future concept states or enable interventions that persist throughout the forecast. Our TRACE architecture instead uses named concepts as the forecasting state, allowing it to predict future activities and propagate the effects of edits over time.

Structured concept dynamics. Relations among concepts provide another direction beyond the standard independent-concept representation. Graph CBMs explicitly model dependencies between concept variables (Xu et al., 2026), while ReCBM uses uncertainty-gated semantic relations to refine unreliable concept states (Sui et al., 2026). Causally reliable CBMs and Causal Concept Graph Models further investigate concept relations aligned with real-world causal mechanisms (De Felice et al., 2025; Dominici et al., 2025). Across these approaches, relational structure primarily supports concept reasoning or refinement at a given prediction point. In contrast, TRACE-CBM uses sparse spatio-temporal graphs as an inspectable predictive mechanism: it both refines observed concepts and propagates named concept states through the forecast horizon. Consequently, edits to concept nodes, relations, or activity beliefs can persist throughout the forecast trajectory.

3 METHOD

While concept bottleneck models utilize named intermediate concepts, their standard formulation does not specify how these concepts change during forecasting or how interventions (edits) to the current model state should propagate to future concept states. **Research question:** *Can future predictions retain a named, editable state by forecasting directly in concept space while modeling interactions between concepts over time?* **TRACE** (Figure 2) refines observed concepts through normalization, sparse spatio-temporal graphs and then recursively forecasts future, editable concept states, conditioned on preceding activity beliefs. A shared linear predictor (activity head) maps observed and forecast states to activity predictions, enabling an interpretable activity forecasting pipeline by design.

3.1 CONCEPT EXTRACTION AND NORMALIZATION

Let a video x consist of several frames partitioned into N equal, non-overlapping windows, each with one of K activity labels: $y_t \in \{1, \dots, K\}$. At prediction time T, the model gets the observed sequence $x_{1:T}$ and predicts the current activity y_T and future activities y_{T+h} for $h \in \{1, \dots, H\}$, where $T + H \leq N$.

Concept extraction. First, we represent each video window through human-understandable concept evidence. We define a vocabulary $\mathcal{C}$ of $C = |\mathcal{C}|$ concepts as short natural-language phrases (Yang et al., 2023) and embed these concepts and the video windows into a shared vision–language space using CLIP-like models (Radford et al., 2021; Bolya et al., 2025). For window embedding $\mathbf{v}_t$ at time t and concept-text embedding $\mathbf{e}_j$ of concept j, the raw concept score is their cosine similarity, $c_{t,j} = \mathbf{v}_t^\top \mathbf{e}_j / (\|\mathbf{v}_t\| \|\mathbf{e}_j\|)$. The offline vocabulary-construction prompt and verification procedure are provided in Algorithm 1 and Appendix C.1, with details on video embeddings in Appendix D.1. After training-split standardization, $\mathbf{c}_t \in \mathbb{R}^C$ are the concept scores for the t-th window. We treat these named similarity vectors as observations of human-readable concepts rather than anonymous latent vectors. To cleanly batch video sequences of different length, we distinguish observed windows from padding with a validity indicator $m_t \in \{0, 1\}$, where $m_t = 1$ for an observed window and $m_t = 0$ for padding. Padded windows do not affect subsequent computations or losses.

Concept-score normalization. Following VLM-based CBMs (Yang et al., 2023; Oikarinen et al., 2023), we use image–text similarity as concept evidence (Oikarinen et al., 2023), but map the raw scores to bounded concept states (Koh et al., 2020). Since similarity-score distributions can differ

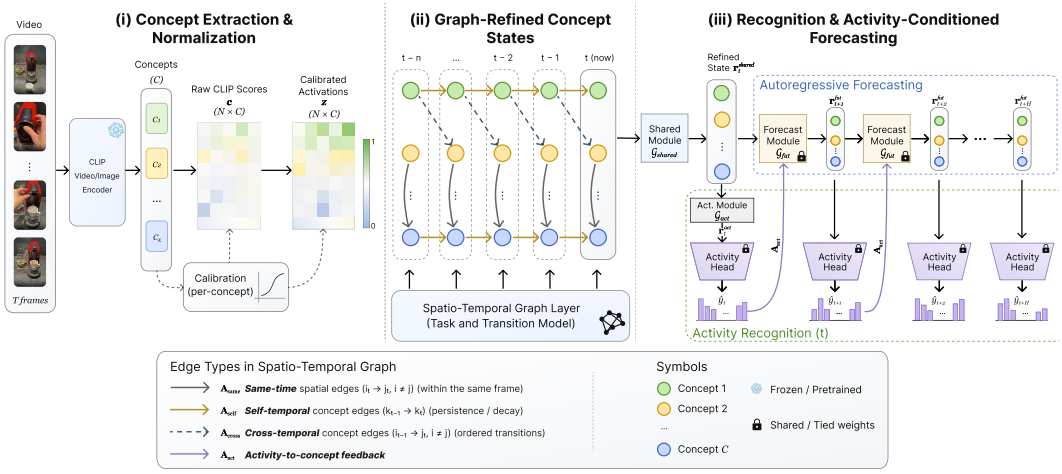

Figure 2: **Overview of the TRACE pipeline.** Normalized concept states are processed by shared-refinement, recognition, and forecast graph modules and decoded by a shared activity head. Activity-to-concept feedback refines the observed state and conditions recursive concept forecasting, making concept, edge, and activity-belief interventions explicit model operations.

across concepts, we learn a separate threshold $\boldsymbol{\tau}$ and a positive scaling factor $\mathbf{s}$ for each concept. For each valid window t, we compute

$$\widetilde{\mathbf{z}}_t = \sigma((\mathbf{c}_t - \boldsymbol{\tau}) \odot \mathbf{s}) \in (0,1)^C, \qquad \mathbf{z}_t = \begin{cases} \widetilde{\mathbf{z}}_t, & m_t = 1, \\ \mathbf{0}, & m_t = 0. \end{cases} \tag{1}$$

where, σ denotes the sigmoid function and $\odot$ denotes elementwise multiplication.

3.2 Graph-Refined Concept States

TRACE performs message-passing computations directly on spatio-temporal graphs of named concept states, allowing each concept to integrate information from related concepts and from its own states over time (Yan et al., 2018). We first introduce the graph modules used in TRACE (Figure 2) before defining the common graph architecture underlying them. Following the pipeline from left to right, the graph in the center illustrates an example of the refined concept states and relations represented by a graph module.

Graph modules. The pipeline comprises three modules: a shared-refinement module $\mathcal{G}_{\text{shared}}$ that refines the sequence of concept states into a common representation $\mathbf{r}$, an activity module $\mathcal{G}_{\text{act}}$ that specializes this representation for current-activity prediction, and a forecasting module $\mathcal{G}_{\text{fut}}$ that recursively predicts futurestates. Each of their observed states are

$$\mathbf{r}^{\text{shared}}_{1:T} = \mathcal{G}_{\text{shared}}(\mathbf{z}_{1:T}), \qquad \mathbf{r}^{\text{act}}_{1:T} = \mathcal{G}_{\text{act}}(\mathbf{r}^{\text{shared}}_{1:T}), \qquad \mathbf{r}^{\text{fut}}_{1:t+1} = \mathcal{G}_{\text{fut}}(\mathbf{r}^{fut}_{1:t}). \tag{2}$$

The activity module $\mathcal{G}_{\text{act}}$ updates the current-activity state based on $\mathbf{r}^{\text{shared}}_{1:T}$ at $h = 0$, whereas $\mathcal{G}_{\text{fut}}$ recursively predicts future concept states, starting from $\mathbf{r}^{fut}_{1:T} = \mathbf{r}^{\text{shared}}_{1:T}$. For $h \geq 1$, the activity head predicts the future activity labels $\hat{\mathbf{y}}_t$ from the $\mathcal{G}_{\text{fut}}$-updated forecast state without reapplying $\mathcal{G}_{\text{act}}$.

Spatio-temporal concept graph module. A node (t, j) in the graph represents concept j at window t, with initial state $z_{t,j}$. The graph modules perform message passing over the sequence of concept states and update their representations. Thereby, $\mathcal{G}^{(l)}$ denotes one graph message-passing layer at depth l, and throughout this paper, we use a single layer ($l = 1$) for all modules, but additional layers can be added. Each module operates on the full sequence and applies the same local relation patterns at every window: $\mathbf{r}^{(0)}_{1:t}$ is full the module-specific input state (Eq. 2), and $\mathbf{r}^{(l)}_{1:t} = \mathcal{G}^{(l)}\left(\mathbf{r}^{(l-1)}_{1:t}\right)$. Each module contains three relation families (Figure 2): (i) *same-time* relations, $\mathbf{A}^{(l)}_{\text{same}} \in \mathbb{R}^{C \times C}$, between distinct concepts at the same time window; (ii) *self-temporal* relations, $\mathbf{a}^{(l)}_{\text{self}} \in \mathbb{R}^{C}$, carrying each

concept from $t-1$ to itself at t; and (iii) *cross-temporal* relations, $\mathbf{A}^{(l)}_{\text{cross}} \in \mathbb{R}^{C\times C}$, carrying one concept at $t-1$ to a different concept at t. Temporal edges connect only adjacent windows, so longer-range dependencies arise by composing these local updates across time and message-passing depth. Because concept states are bounded between 0 and 1, the modules applies additive updates with logit transformation $\phi(\mathbf{u}) = \log(\mathbf{u}/(1-\mathbf{u}))$ and maps the result back to $(0,1)^C$ with a sigmoid. For target window t, a spatio-temporal concept graph module l computes

$$\mathbf{r}_t^{(l)} = \sigma\Big(\phi\Big(\mathbf{r}_t^{(l-1)}\Big) + \boldsymbol{\lambda}_{\mathcal{G}}^{(l)} \odot \Big[\mathbf{r}_t^{(l-1)}\mathbf{A}^{(l)}_{\text{same}} + \mathbf{r}_{t-1}^{(l-1)} \odot \mathbf{a}^{(l)}_{\text{self}} + \mathbf{r}_{t-1}^{(l-1)}\mathbf{A}^{(l)}_{\text{cross}}\Big]\Big), \tag{3}$$

where $\boldsymbol{\lambda}_{\mathcal{G}}^{(l)}$ is a learned per-concept update gate. The first term preserves an identity path for every concept coordinate, whereas the remaining terms aggregate relational messages. We apply top-q pruning to the additive messages, and leave the identity path unpruned. For same-time and cross-temporal relations, each target retains its strongest incoming sources under the sparsification parameters q_{same} and q_{cross}, respectively, and for self-temporal relations, q_{self} retains the strongest same-concept channels. The sparsity budgets are ramped up during training rather than imposed from the start: pruned relations contribute zero, while the weights of retained relations remain differentiable.

3.3 Recognition and activity-conditioned forecasting

Activity-to-concept feedback. To enable interventions on the activity distribution, we learn direct activity-to-concept feedback. TRACE represents this feedback with a sparse directed bipartite graph from K activity classes to C named concepts. Its rows correspond to activity classes, its columns to concepts, and A_{kj} is the raw message strength from activity k to concept j. Its effective adjacency matrix is

$$\mathbf{A}_{\text{act}} = \mathbf{A} \odot \sigma(\boldsymbol{\lambda}_A) \odot \mathbf{M}_{\text{q}_{\text{fb}}} \in \mathbb{R}^{K\times C}, \tag{4}$$

where $\boldsymbol{\lambda}_A$ contains learned gate logits and $\mathbf{M}_{\text{q}_{\text{fb}}}$ is a binary top-q_{fb} mask that retains the strongest outgoing concept relations for each activity class. Given initial predicted activity probabilities $\hat{\mathbf{y}}_t$ (before feedback refinement), $\hat{\mathbf{y}}_t\mathbf{A}_{\text{act}}$ is the activity-derived concept feedback. At window t, we aggregate this signal from the preceding L_{hist} windows:

$$\mathbf{d}_t^{\text{hist}} = \sum_{\delta=1}^{L_{\text{hist}}} \sigma(\boldsymbol{\lambda}_\delta)\, m_{t-\delta}\, \hat{\mathbf{y}}_{t-\delta}\mathbf{A}_{\text{act}}. \tag{5}$$

where δ indexes the temporal lag, λ_δ is a learned lag gate, and $m_{t-\delta} \in \{0,1\}$ is a validity indicator that equals 1 for an observed or already generated rollout state and 0 for padding or unavailable history. This feedback refines the activity state and the forecast state:

$$\widetilde{\mathbf{r}}_t^{\alpha} = \sigma\big(\phi(\mathbf{r}_t^{\alpha}) + \mathbf{d}_t^{\text{hist}}\big), \qquad \alpha \in \{\text{act}, \text{fut}\}. \tag{6}$$

For $t < T$, we set $\widetilde{\mathbf{r}}_t^{\alpha} = \mathbf{r}_t^{\alpha}$. Thus, $\widetilde{\mathbf{r}}_t^{\alpha}$ denotes the refined concept state after activity feedback.

Activity-conditioned autoregressive forecasting. The forecast module then recursively updates the concept state starting from $\widetilde{\mathbf{r}}^{\text{fut}}_{1:T}$, with $\widetilde{\mathbf{r}}^{\text{fut}}_{1:T+h}$ denoting the forecast at $T+h$:

$$\widetilde{\mathbf{r}}^{\text{fut}}_{1:t+1} = \sigma\big(\phi\big(\mathcal{G}_{\text{fut}}\big(\widetilde{\mathbf{r}}^{\text{fut}}_{1:t}\big)\big) + \mathbf{d}^{\text{hist}}_{t+1}\big). \tag{7}$$

At each time step, feedback is applied conditioned on the history of past activity states $\mathbf{d}_t^{\text{hist}}$.

Shared activity head. Since we want to have a shared predictor that maps concept states to activity labels, both current activity recognition at time T and future activity forecasting use the same linear predictor (Activity Head), parameterized by weights $\mathbf{W} \in \mathbb{R}^{K\times C}$ and bias $\mathbf{b} \in \mathbb{R}^K$:

$$\boldsymbol{\ell}_t = \phi(\widetilde{\mathbf{r}}_t^{\alpha})\,\mathbf{W}^\top + \mathbf{b}, \qquad \hat{\mathbf{y}}_t = \text{softmax}(\boldsymbol{\ell}_t), \quad m_t = 1. \tag{8}$$

Accordingly, the activity head predicts the activity at the future time $T+h$ from the forecasted concept state $\widetilde{\mathbf{r}}^{\text{fut}}_{T+h}$.

3.4 Training Objective

Future concept targets are obtained by applying the learned normalizer (Eq. 1) to the CLIP-based concept scores of future states. The learned normalizer is excluded from gradient computation and, therefore, does not update its parameters. We train the model with a multi-task objective:

$$\mathcal{L} = \mathcal{L}_{\text{act}} + \mathcal{L}_{\text{forecast}} + \lambda_{\text{c}}\mathcal{L}_{\text{concept}} + \lambda_{\text{cls}}\|\mathbf{W}_{\text{act}}\|_1. \tag{9}$$

Here, $\mathcal{L}_{\text{act}}$ supervises the current activity prediction, $\mathcal{L}_{\text{forecast}}$ supervises future activities across prediction horizons, and $\mathcal{L}_{\text{concept}}$ aligns predicted concept states with the stop-gradient future concept targets. The ℓ_1 penalty $\lambda_{\text{cls}}\|\mathbf{W}_{\text{act}}\|_1$ promotes sparsity in the activity-classification weights. The activity and forecasting terms use background-silence (SIL)-aware supervision. Forecasting additionally employs transition-tolerant targets for near-boundary predictions. Full loss definitions, horizon weights, and tolerance sets appear in Appendix E.

3.5 Interventions

TRACE's interpretable design enables targeted investigations and interventions (edits) without changing its model parameters. We support three types of interventions: (i) *What if a concept were different?*: replace or clamp a named concept activation in $[0, 1]$. (ii) *How does changing an activity belief affect future concept states?*: modify a source activity distribution by setting one class to a chosen probability and redistributing the remaining mass. (iii) *Which relational dependencies does the TRACE use?*: modify a selected effective same-time, self-temporal, or cross-temporal relation while leaving the residual path unchanged. Since all states remain in the same named concept space, predictions remain interpretable through the shared decoder, while the graphs expose the learned relations through which an intervention acts. An edit at T changes the initial concept state, activity belief, or relation for $h = 1$, thereby affecting the prediction at $T + 1$ (see the node intervention in Fig. 1). The resulting state and activity distribution then feed into the $h = 2$ transition, recursively propagating the intervention through the same autoregressive rollout up to $h = H$.

4 Evaluation

In this section, we address the research questions on TRACE's intervenability introduced in Section 3.5. We first describe the experimental setup, then present quantitative results in Section 4.1 and evaluate TRACE's intervenability in Section 4.2. Additional experiments and ablations are provided in the Appendix.

Datasets. We evaluate TRACE on three video activity datasets: Breakfast (Kuehne et al., 2014), MPII Cooking 2 (Rohrbach et al., 2015), and BARISTA (Knab et al., 2026c). The main comparison uses BARISTA, averages the official Breakfast s1–s4 train-test splits, and uses MPII Cooking 2 (Attr). Ablations and intervention experiments use the same datasets and protocols, except that Breakfast is evaluated on the official s1 split only. Further datasets are evaluated in Appendices A.8 and A.10. All methods use five observed 32-frame windows and forecast H=3 steps ahead. We use a frozen PE-L14 (Bolya et al., 2025) backbone to obtain video and concept embeddings and action-label targets, with SIL assigned to otherwise unlabeled windows. Appendix D.1 gives the target definitions and complete protocol details, while Appendix A.6 and Appendix A.9 report the PE-G14 backbone.

Concepts. For each dataset, we construct a $|C|$=128 textual concept vocabulary from training windows using a VLM-based pipeline to propose domain-specific concepts (Oikarinen et al., 2023; Yang et al., 2023). Qwen3.5-27B (Team, 2026) receives four frames of a window, and its proposed concepts are retrieved, visually verified, deduplicated, and filtered for label overlap before training (see Appendix C.1).

Baselines. We compare TRACE against a training-free zero-shot baseline (PE-L/14). For black-box models, we use SlowFast-style and feature-transformer architectures (Feichtenhofer et al., 2019; Bertasius et al., 2021). As concept-based approaches, we use a Sparse Linear baseline with matched concept-space sparsity and a decoder that replaces the graph component, and MoTIF (Knab et al., 2026a) for activity recognition only. The zero-shot baseline predicts the activity with cosine similarity between its text prompt "a video of *label*" and the frozen video-window feature (Bolya et al., 2025). The black-box models operate on the same frozen video features without concept activations or graph structure; Appendix D provides the implementation and complete configurations.

Table 1: **Predictive performance across model families.** Mean ± sample standard deviation over five seeds for activity classification and H3 forecasting on BARISTA, Breakfast s1–s4 (averaged within seed), and MPII Attr. MoTIF reports classification only; black-box baselines use the same window features without concept activations.

		BARISTA				Breakfast (s1–s4)				MPII Cooking 2 (Attr)			
		Activity		H3		Activity		H3		Activity		H3	
Family	Method	Top-1	Top-3	Top-1	Top-3	Top-1	Top-3	Top-1	Top-3	Top-1	Top-3	Top-1	Top-3
Zero-shot	Cos. sim.	2.2	5.4	n/a	n/a	18.2±2.8	37.3±4.7	n/a	n/a	6.8	15.9	n/a	n/a
Black-box	Feat. Transf.	79.2±0.3	89.3±0.5	69.6±0.3	82.9±0.3	47.3±0.4	69.6±0.6	39.7±0.6	63.7±0.2	50.1±1.2	70.3±1.4	40.7±0.7	61.7±0.8
	SlowFast	79.6±0.5	90.3±0.9	67.6±1.0	82.5±0.8	46.8±0.5	69.5±0.6	39.7±0.6	64.1±0.4	49.8±0.6	70.4±0.5	40.4±0.4	62.3±0.5
Concept	Sparse Lin.	67.5±1.4	79.3±1.5	**68.5±0.4**	**82.5±0.3**	43.4±0.2	67.9±0.1	36.8±0.2	61.5±0.2	43.3±0.3	61.5±0.5	37.7±0.2	58.3±0.3
	MoTIF	72.9±1.9	86.2±0.8	n/a	n/a	44.3±0.3	69.9±0.3	n/a	n/a	44.4±0.9	65.6±0.7	n/a	n/a
	TRACE (ours)	**76.0±2.2**	**88.9±0.8**	67.6±0.3	81.5±1.1	**45.5±0.2**	**70.6±0.2**	**37.5±0.1**	**62.7±0.4**	**47.0±0.4**	**67.4±0.2**	**38.8±0.5**	**58.6±0.3**

4.1 Quantitative Evaluation

Concept-space forecasting remains competitive with black-box baselines. Table 1 shows that TRACE achieves the strongest activity accuracy among concept-based methods across all three datasets and substantially narrows the gap to black-box feature baselines, while additionally providing interpretability and intervention capabilities (Section 4.2). Among concept-based methods, our model improves activity top-1 accuracy over MoTIF on all three datasets and remains the overall best on H3 forecasting, while Sparse Linear performs comparably by capturing direct concept-to-concept dependencies through its sparse coefficient matrix but lacks structured concept/activity intervention mechanisms. The substantially weaker zero-shot baseline further highlights the importance of learning over concept representations. Thus, TRACE remains competitive with black-box models while exposing internal semantic states that can be investigated and intervened on.

Component ablation. Here, we ablate whether the full model benefits from all of its components. Table 2 separately removes recursive transitions, graph structure, activity feedback, and score normalization on BARISTA, Breakfast s1, and MPII Cooking 2. The complete model achieves the highest mean H3 top-1 accuracy across the three datasets (49.6%); while score normalization consistently helps to improve forecasting performance. Dense instead of graph rollout improves BARISTA but reduces Breakfast and MPII performance by 2.7 and 2.8 points, respectively. However, as shown by the dependency-concentration metric in Appendix B.1, the dense rollout is substantially less sparse (87 effective H3 parents vs. 5 for graph rollout), and the weight of concept relations is less concentrated (15% influence of the top 5 concepts compared to 75% for graph rollout). While the sparsity of the graph rollout is generally desirable, TRACE can also operate with the dense rollout when desired. Further ablations in Appendix A.

Table 2: **Component ablation.** H3 top-1 accuracy, reported as mean ± sample standard deviation over three seeds. Activity and H3 top-3 results are in Appendix A.2.

Variant	BARISTA	Breakfast	MPII Attr
Full model	67.4±0.3	**42.9±0.2**	**38.7±0.6**
Persistence	66.0±0.2	**42.9±0.1**	38.2±0.3
Dense rollout, no graph	**70.2±0.3**	40.2±0.3	35.9±0.3
Graph, no activity feedback	67.0±0.1	42.5±0.6	**38.7±0.4**
No score normalization	64.3±0.5	39.6±0.7	35.0±0.3

4.2 Intervention Evaluation

What are the predictive effects of interventions? In Table 3 we evaluate the intervenability with up to a budget of $b = 5$ cumulative edits on 30 initially wrong (W) and correct (C) cases per dataset, recomputing the full rollout after each edit, answering questions (i) and (ii) in Section 3.5. Activity edits use ground-truth activity with $b = 3$; node edits use future H3 concept targets and the true H3 label for ranking; edge edits flip the strongest learned relations without future labels. This extends ground-truth (*oracle*) test-time concept interventions (Koh et al., 2020; Shin et al., 2023; Espinosa Zarlenga et al., 2023) to forecasting. Activity-belief and concept-node edits both improve H3 forecasts. However, while activity-belief edits correct more cases on Breakfast and MPII node edits consistently increase true-class probability without inducing misclassifications. The negligible effect of single-edge flips indicates robustness to individual relational perturbations, meaning the edges serve more for inspection (see Figure 3). Together, these results demonstrate that TRACE can be corrected through interventions on activity-conditioned transitions and named concept states.

Table 3: **Model edits for level H3.** Interventions on 30 wrong and correct cases per seed, with activity-belief (b=3), concept-node (b=5), and strongest edge sign edits (b=5). $\Delta p(y)$ is the true-class probability change (%); W→C and C→W are cumulative correction and harm rates.

	Activity belief (A, b=3)			Concept node (N, b=5)			Edge sign flip (E, b=5)		
Dataset	$\Delta p(y)$	W→C	C→W	$\Delta p(y)$	W→C	C→W	$\Delta p(y)$	W→C	C→W
BARISTA	$+3.0_{\pm 0.5}$	$12.2_{\pm 8.4}$	$0.0_{\pm 0.0}$	$+7.2_{\pm 5.3}$	$25.6_{\pm 18.4}$	$0.0_{\pm 0.0}$	$-0.02_{\pm 0.02}$	$0.0_{\pm 0.0}$	$1.1_{\pm 1.9}$
Breakfast s1	$+28.3_{\pm 4.2}$	$61.1_{\pm 5.1}$	$4.4_{\pm 5.1}$	$+11.3_{\pm 1.2}$	$25.6_{\pm 3.8}$	$0.0_{\pm 0.0}$	$-0.00_{\pm 0.11}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$
MPII Attr	$+21.1_{\pm 1.7}$	$46.7_{\pm 8.8}$	$2.2_{\pm 3.8}$	$+9.8_{\pm 1.5}$	$34.4_{\pm 6.9}$	$0.0_{\pm 0.0}$	$+0.04_{\pm 0.05}$	$2.2_{\pm 1.9}$	$1.1_{\pm 1.9}$

Can a VLM plan concept edits? TRACE functionality allows to test whether a VLM, Qwen3.6-35B-A3B (Team, 2026), can act as an external intervention agent for the observed window T and forecast $T{+}1$ and $T{+}2$. For this, we construct four different test scenarios: blind Qwen receives the model state and candidate edits, while oracle-assisted variants additionally receive the true H3 label and counterfactual edit effects. We compare one-shot intervention, sequential intervention with updated effects ($b = 5$), and a greedy oracle that reruns each next edit and selects the one maximizing held-out true H3 probability in Table 4; Appendix E.1.2 provides the full setup and prompt.

Table 4: **Qwen planning for H3 concept-node interventions.** Mean ± standard deviation after five cumulative concept-node edits over three seeds, with 30 initially wrong and 30 correct cases per dataset and seed. $\Delta p(y)$ is in percentage points; W→C is the cumulative correction rate. Blind Qwen receives neither the future label nor counterfactual effects; one-shot and sequential are oracle-assisted.

	Blind Qwen		One-shot Qwen		Sequential Qwen		Greedy oracle	
Dataset	$\Delta p(y)$	W→C	$\Delta p(y)$	W→C	$\Delta p(y)$	W→C	$\Delta p(y)$	W→C
BARISTA	$-2.9_{\pm 1.7}$	$10.0_{\pm 8.8}$	$+12.9_{\pm 4.7}$	$47.8_{\pm 13.9}$	$+16.6_{\pm 6.1}$	$\mathbf{57.8_{\pm 10.7}}$	$\mathbf{+17.8_{\pm 6.8}}$	$\mathbf{57.8_{\pm 16.4}}$
Breakfast s1	$-13.0_{\pm 5.7}$	$13.3_{\pm 6.7}$	$+23.1_{\pm 3.2}$	$56.7_{\pm 12.0}$	$+32.4_{\pm 4.7}$	$\mathbf{75.6_{\pm 3.8}}$	$\mathbf{+34.4_{\pm 5.2}}$	$72.2_{\pm 5.1}$
MPII Attr	$-9.0_{\pm 1.6}$	$12.2_{\pm 3.8}$	$+20.6_{\pm 2.4}$	$52.2_{\pm 10.7}$	$+25.7_{\pm 3.3}$	$58.9_{\pm 7.7}$	$\mathbf{+28.5_{\pm 4.0}}$	$\mathbf{61.1_{\pm 9.6}}$

Blind Qwen corrects some cases, but overall lowers $p(y)$. Oracle information makes one-shot and sequential planning effective, with sequential being close to greedy. Together, these results show that Qwen can plan corrective concept edits when provided with information about the future label and counterfactual edit effects. The next experiment tests a related non-oracle setting by asking Qwen to steer the model toward a specified target activity.

Table 5: **H3 steering**. Mean ± sample standard deviation over three seeds with 60 test windows per dataset and seed. $\Delta p(a^\star)$ is in percentage points; $p\uparrow$ is the target-increase rate.

Dataset	Policy	$\Delta p(a^\star)$	$p\uparrow$	Target top-1
BARISTA	Random	$-0.4_{\pm 0.7}$	$41.7_{\pm 7.3}$	$1.7_{\pm 1.7}$
	Target heuristic	$+4.0_{\pm 1.5}$	$80.0_{\pm 10.9}$	$3.9_{\pm 4.2}$
	Qwen steering	$+5.8_{\pm 3.3}$	$87.2_{\pm 14.9}$	$10.6_{\pm 8.6}$
Breakfast s1	Random	$+3.7_{\pm 2.9}$	$54.4_{\pm 7.5}$	$11.7_{\pm 1.7}$
	Target heuristic	$+68.9_{\pm 1.6}$	$99.4_{\pm 1.0}$	$90.0_{\pm 3.3}$
	Qwen steering	$+64.0_{\pm 3.3}$	$98.9_{\pm 1.0}$	$88.9_{\pm 2.5}$
MPII Attr	Random	$+2.0_{\pm 0.7}$	$43.3_{\pm 3.3}$	$8.3_{\pm 0.0}$
	Target heuristic	$+51.9_{\pm 6.5}$	$100.0_{\pm 0.0}$	$84.4_{\pm 8.6}$
	Qwen steering	$+47.7_{\pm 4.4}$	$98.3_{\pm 1.7}$	$78.3_{\pm 1.7}$

Can a VLM steer a requested H3 target? We select a requested activity from the three most likely train-only H3 transitions under the model's current activity belief, excluding its default prediction. For each case, all policies receive on/off edits for 10 concepts selected by their contribution to the requested activity and activity-belief edits for three candidates, including the target. Random shuffles these edits, the target heuristic prioritizes target-activity and then concept-on edits, and Qwen selects five nonconflicting cumulative edits from the observed storyboard and model state. Future labels, and intervention effects remain hidden (Appendix E.1.2). Both steering strategies outperform the random baseline (Table 5). Although Qwen consistently increases the requested activity's probability, the target heuristic is stronger on Breakfast and MPII. BARISTA also exhibits positive target gains, but its activity and concept interventions have more distributed effects.

Which relational dependencies does the TRACE use? Figure 3 visualizes representative BARISTA and Breakfast sequences using the same backbone as the quantitative evaluation. The figure shows the information flow of TRACE: observed windows, activity distributions decoded from refined and forecasted concept states, and learned graph relations used for state refinement. By inspecting these relations and concept scores, we can answer: *Which relational dependencies does the forecaster*

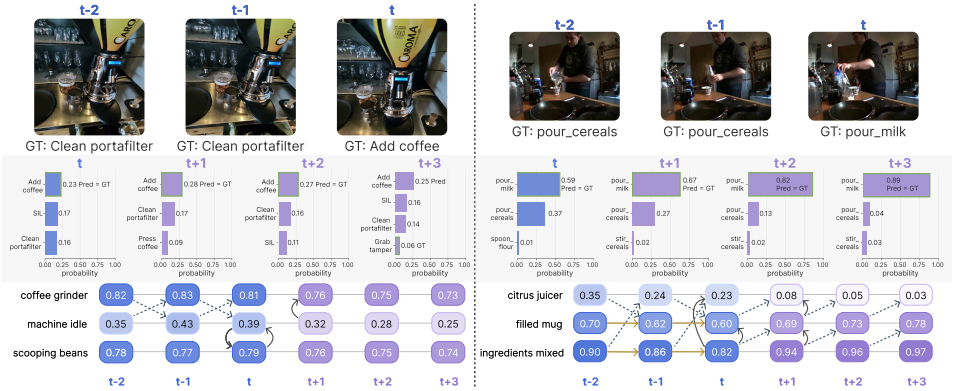

Figure 3: **Exemplary forecasts.** Representative BARISTA and Breakfast sequences show observed windows, activity distributions decoded from refined and forecasted concept states, and selected learned relations used by the model.

use? Unlike an opaque latent trajectory, future predictions remain readable through named concept states, activity beliefs, and graph dependencies. Each example displays a subset of the most relevant concept nodes and their learned relations (we omit edge strengths for visual clarity). As motivated in Section 3, the sparse graphs connect only selected concept pairs rather than imposing dependencies between all concepts. This sparsity still lets us *trace* how interventions propagate through learned relations and, for BARISTA, why the model falsely predicts *add coffee* at $t{+}3$.

What drives TRACE's predictions? Complementing the previous intervention studies and the examples shown in Figure 3, we now assess the interpretability of TRACE's concept-based predictions through targeted manual interventions on influential concepts and activities. For BARISTA, suppressing *coffee grinder* to 0.1 changes the preferred activity from *add coffee* to *SIL* for H2 and H3. For Breakfast, suppressing *filled mug* lowers p for *pour_milk* and increases *pour_cereals*. After a second intervention by suppressing *ingredients mixed*, TRACE predicts the activity *pour_cereals* for H2 and H3. These interventions show that we humans can steer its predictions through the concept space and why TRACE prefers its original prediction over a plausible alternative.

5 Conclusion and Future Work

Conclusion. We introduced TRACE, a concept-rollout model for intervenable video activity forecasting. The model refines observed concept evidence using sparse spatio-temporal graph modules and recursively forecasts named concept states with activity-conditioned feedback. This shared semantic state supports retraining-free edits to concepts, activity beliefs, and learned edges, while sparse graph dynamics preserve localized, inspectable concept contributions throughout forecasting. TRACE therefore provides an editable interface beyond the observed input and across the forecast horizon.

Limitations. Although this interface makes forecasts easier to inspect and modify, several limitations remain. The frozen video encoder and fixed 32-frame windows may miss the task-specific appearance or fine temporal changes; adapted representations and variable temporal context could improve both. Text–video-aligned concepts can introduce label noise, and direct concept supervision would enable a stronger evaluation of grounding and leakage (Espinosa Zarlenga et al., 2022). The exact learned edges between concepts vary across seeds, although shared edges usually retain their direction. The graphs should therefore be read as predictive dependencies rather than verified causal structure.

Future work. Promising directions include adaptive concept structures, longer-horizon and uncertainty-aware forecasting, and transfer to multivariate time series. Our interventions expose mechanisms for modifying forecasts, but a deployable policy to detect unreliable concepts and select corrections without future labels remains open (Steinmann et al., 2024; Espinosa Zarlenga et al., 2023). Improving forecasting accuracy while retaining interpretability and intervenability is another important direction. Finally, extending persistent concept interventions from image generation (Ismail et al., 2024; Martínez-García et al., 2026) to generative video models could enable semantic edits that remain consistent across generated frames (Labs et al., 2025).

AI USE STATEMENT

In this work, we used generative AI tools for several tasks that fall under the required disclosure categories. Specifically, generative AI assisted in (i) generating a synthetic test set, (ii) partially implementing the proposed conceptual framework, and (iii) providing feedback on the experimental design by brainstorming additional experiments and identifying potential gaps in our evaluation. Throughout the development process, generative AI was also used to accelerate software implementation. Generative AI is also an integral component of the proposed methodology itself. We employ a VLM during concept generation and use it also as a judge for the corresponding evaluation protocol. Additionally, we used generative AI to polish the readability and clarity of pre-written text, refine the formatting LaTeX (e.g., tables), and perform AI-assisted manuscript reviews to identify inconsistencies, unclear explanations or missing details prior to submission, supplementing human feedback. All scientific figures were designed and created by the authors without AI.

ETHICS STATEMENT

This work does not involve ethical considerations beyond those associated with the publicly available datasets and pretrained models used in our experiments.

REPRODUCIBILITY STATEMENT

Sections 3 and 4 specify the model, intervention semantics, datasets, and evaluation protocols. The appendix provides concept-construction details, complete hyperparameters, additional datasets and rollouts, vocabulary diagnostics, and the intervention metrics recorded for every edit prefix. The supplementary code contains the fixed feature preparation, training configurations, and evaluation scripts used to generate the reported tables.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629. IEEE, 2023.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 813–824. PMLR, 18–24 Jul 2021.

Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Abdul Rasheed, Junke Wang, Marco Monteiro, Hu Xu, Shiyu Dong, Nikhila Ravi, Shang-Wen Li, Piotr Dollar, and Christoph Feichtenhofer. Perception encoder: The best visual embeddings are not at the output of the network. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. The epic-kitchens dataset: Collection, challenges and baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):4125–4141, 2020.

Giovanni De Felice, Arianna Casanova Flores, Francesco De Santis, Silvia Santini, Johannes Schneider, Pietro Barbiero, and Alberto Termine. Causally reliable concept bottleneck models. *Advances in Neural Information Processing Systems*, 38:149099–149139, 2025.

Giovanni De Felice, Riccardo Giuseppe D'Elia, Alberto Termine, Pietro Barbiero, Giuseppe Marra, and Silvia Santini. Position: Interpretability in deep time series models demands semantic alignment. In *Forty-third International Conference on Machine Learning Position Paper Track*, 2026.

Gabriele Dominici, Pietro Barbiero, Mateo Espinosa Zarlenga, Alberto Termine, Martin Gjoreski, Giuseppe Marra, and Marc Langheinrich. Causal concept graph models: Beyond causal opacity in deep learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Zohreh Shams, Frederic Precioso, Stefano Melacci, Adrian Weller, et al. Concept embedding models: Beyond the accuracy-explainability trade-off. *Advances in neural information processing systems*, 35:21400–21413, 2022.

Mateo Espinosa Zarlenga, Katherine M. Collins, Krishnamurthy Dvijotham, Adrian Weller, Zohreh Shams, and Mateja Jamnik. Learning to receive help: Intervention-aware concept embedding models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6202–6211, 2019.

Rohit Girdhar and Kristen Grauman. Anticipative video transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 13505–13515, 2021.

Dayoung Gong, Joonseok Lee, Manjin Kim, Seong Jong Ha, and Minsu Cho. Future transformer for long-term action anticipation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3052–3061, 2022.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Marton Havasi, Sonali Parbhoo, and Finale Doshi-Velez. Addressing leakage in concept bottleneck models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022.

Aya Abdelsalam Ismail, Julius Adebayo, Hector Corrada Bravo, Stephen Ra, and Kyunghyun Cho. Concept bottleneck generative models. In *International Conference on Learning Representations*, volume 2024, pp. 34114–34132, 2024.

Patrick Knab, Sascha Marton, Philipp Johannes Schubert, Drago Andres Guggiana Nilo, and Christian Bartelt. Concepts in motion: Temporal concept bottleneck model for interpretable video classification. In *2nd Workshop on Compositional Learning: Safety, Interpretability, and Agents*, 2026a.

Patrick Knab, David Steinmann, Christian Bartelt, Kristian Kersting, Bernt Schiele, Thomas Seidl, Udo Schlegel, and Wolfgang Stammer. What's in the bottle? a survey and roadmap of concept bottleneck models. *Transactions on Machine Learning Research*, 2026b. ISSN 2835-8856.

Patrick Knab, Orgest Xhelili, Inis Buzi, Drago Andres Guggiana Nilo, Mohd Saquib Khan, Lorenz Kolb, Manuel Scherzer, Kerem Yildirir, Christian Bartelt, and Philipp Johannes Schubert. BARISTA: A multi-task egocentric benchmark for compositional visual understanding. In *ICML 2026 Workshop on Combining Theory and Benchmarks: Towards A Virtuous Cycle to Understand and Guarantee Foundation Model Performance*, 2026c.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5338–5348. PMLR, 13–18 Jul 2020.

H. Kuehne, A. B. Arslan, and T. Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of Computer Vision and Pattern Recognition Conference (CVPR)*, 2014.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

Jongseo Lee, Wooil Lee, Gyeong-Moon Park, Seong Tae Kim, and Jinwoo Choi. Disentangled concepts speak louder than words: Explainable video action recognition. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Yin Li, Miao Liu, and James M Rehg. In the eye of the beholder: Gaze and actions in first person video. *IEEE transactions on pattern analysis and machine intelligence*, 45(6):6731–6747, 2021.

Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3202–3211, 2022.

María Martínez-García, Ricardo Vazquez Alvarez, Alejandro Lancho, Pablo Olmos, and Isabel Valera. A probabilistic hard concept bottleneck for steerable generative models. In *International Conference on Learning Representations*, volume 2026, pp. 83758–83790, 2026.

Tuomas Oikarinen, Subhro Das, Lam M. Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. In *The Eleventh International Conference on Learning Representations*, 2023.

Katharina Prasse, Patrick Knab, Sascha Marton, Christian Bartelt, and Margret Keuper. DCBM: Data-efficient visual concept bottleneck models. In *Forty-second International Conference on Machine Learning*, 2025.

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 Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021.

Marcus Rohrbach, Anna Rohrbach, Michaela Regneri, Sikandar Amin, Mykhaylo Andriluka, Manfred Pinkal, and Bernt Schiele. Recognizing fine-grained and composite activities using hand-centric features and script data. *International Journal of Computer Vision*, pp. 1–28, 2015. ISSN 0920-5691. doi: 10.1007/s11263-015-0851-8.

Sungbin Shin, Yohan Jo, Sungsoo Ahn, and Namhoon Lee. A closer look at the intervention procedure of concept bottleneck models. In *International conference on machine learning*, pp. 31504–31520. PMLR, 2023.

David Steinmann, Wolfgang Stammer, Felix Friedrich, and Kristian Kersting. Learning to intervene on concept bottlenecks. In *Forty-first International Conference on Machine Learning*, 2024.

David Steinmann, Wolfgang Stammer, Antonia Wüst, and Kristian Kersting. Object-centric concept-bottlenecks. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

An Sui, Yuzhu Li, Fuping Wu, and Xiahai Zhuang. Recbm: Uncertainty-gated relational reasoning for concept bottleneck models, 2026.

Qwen Team. Qwen3.5-omni technical report, 2026.

Haotian Xu, Tsui-Wei Weng, Lam M. Nguyen, and Tengfei Ma. Graph concept bottleneck models. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856.

Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19187–19197, 2023.

Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. In *ICLR 2022 Workshop on PAIR^2Struct: Privacy, Accountability, Interpretability, Robustness, Reasoning on Structured Data*, 2022.

A Additional quantitative evaluation

A.1 Architecture ablations: graph layers and top-k

TRACE uses one graph layer in the shared stack and one layer in each task-specific branch. It retains k=20 spatial and cross-temporal messages per target concept, and k=20 same-concept temporal channels globally, following the spatio-temporal message-passing perspective of Yan et al. (2018). The activity-feedback graph retains k=40 label-to-concept messages. These are the default settings for the proposed model's main and supplementary results. Table 26 gives the complete topology, including the gradual top-k warm-up schedule. The transition-factorial structural ablation in the main paper separately removes recursive transitions, graph structure, activity feedback, and score normalization while retaining the rest of the training protocol. We additionally ran in Table 6 a sensitivity sweep on the three single-split main protocols. The sweep varies graph depth, isolates each concept-relation family while keeping activity feedback and other settings fixed, and doubles all three concept-relation budgets.

Table 6: **Graph architecture and topology sensitivity.** PE-L14, 128-concept results on the official BARISTA split, Breakfast s1, and MPII Attr single-split protocols. Values are H3 top-1 accuracy in percent, reported as mean ± sample standard deviation over seeds 42-44. Depth rows jointly set the shared and task-specific graph-stack depth; one-hot rows retain only the named concept-relation family while keeping activity feedback at its default setting; the final row uses k=40 for all three concept-relation families. Best result per dataset is bold; second best is underlined.

Variant	BARISTA	Breakfast s1	MPII Attr
Full (1 layer, k=20)	67.4±0.3	<u>42.9±0.2</u>	38.7±0.6
2 graph layers	68.0±0.4	42.8±1.1	38.5±1.0
3 graph layers	<u>68.2±0.3</u>	**43.3±0.6**	<u>38.8±0.3</u>
4 graph layers	**68.5±0.1**	<u>42.9±1.3</u>	38.0±0.3
Spatial only	67.5±0.5	42.4±0.7	38.5±0.5
Same-concept temporal only	67.3±0.3	42.8±0.6	**39.2±0.3**
Cross-temporal only	67.4±0.6	<u>42.9±0.2</u>	38.4±0.5
All families, k=40	67.8±0.2	42.8±0.4	38.5±0.5

The depth-3 arm has the highest unweighted dataset-macro H3 top-1 accuracy, about 0.4 percentage points above the full setting. Depth 4 improves BARISTA but reduces MPII, while one-hot relation effects are family- and dataset-dependent. Doubling all concept-relation budgets to k=40 changes H3 top-1 by at most 0.5 percentage points. The selected one-layer-per-stack, k=20 configuration is therefore a stable balanced default setting rather than an accuracy-optimal choice on every dataset.

A.2 Matched transition-factorial ablation

Table 7 gives the complete activity and H3 statistics underlying Table 2. The dense/no-graph arm is not parameter matched to the graph arm and is interpreted as a structural comparison. Persistence copies the final observed concept state at every future horizon and disables graph processing and activity feedback. Dense rollout uses a dense observed-state refiner and dense recursive forecast, with activity feedback disabled. The graph-no-feedback arm retains graph refinement and recursive graph rollout but removes predicted-activity feedback, while the no-score-normalization arm disables the learned concept-score normalizer. The full arm is graph rollout with activity feedback and score normalization, so it is not duplicated as a separate graph-with-feedback row.

The full configuration has the highest unweighted average H3 top-1 accuracy across the three protocols, but no single transition component improves every dataset. In particular, score normalization is consistently important, whereas dense rollout helps BARISTA but harms the two cooking datasets and activity feedback has little short-horizon accuracy effect. The graph should therefore be interpreted as an inspectable transition parameterization for the model-edit interface rather than as a universally accuracy-improving or causal structure.

Table 7: **Transition-factorial structural ablation.** PE-L14, 128-concept results on official BARISTA, Breakfast s1, and MPII Attr protocols, reported as mean ± sample standard deviation over seeds 42–44 in percent. The dense/no-graph arm is not parameter matched to the graph arm. Best result per dataset and metric is bold; second best is underlined.

Dataset	Variant	Activity top-1	H3 top-1	H3 top-3
BARISTA	Full model	76.6±2.9	67.4±0.3	81.8±1.5
	Persistence	77.1±0.8	66.0±0.2	79.5±0.2
	Dense rollout, no graph	**81.1±0.8**	**70.2±0.3**	**85.1±0.6**
	Graph, no activity feedback	76.4±1.8	67.0±0.1	81.4±0.9
	No score normalization	75.3±0.5	64.3±0.5	78.5±0.4
Breakfast s1	Full model	**51.7±0.4**	**42.9±0.2**	**68.9±0.7**
	Persistence	49.1±0.2	**42.9±0.1**	66.2±0.0
	Dense rollout, no graph	50.5±0.4	40.2±0.3	66.5±0.7
	Graph, no activity feedback	50.9±0.3	42.5±0.6	67.1±0.3
	No score normalization	46.5±0.8	39.6±0.7	64.6±0.8
MPII Attr	Full model	47.0±0.6	**38.7±0.6**	58.5±0.4
	Persistence	45.6±0.2	38.2±0.3	**59.2±0.1**
	Dense rollout, no graph	**48.8±0.4**	35.9±0.3	57.2±1.0
	Graph, no activity feedback	46.6±0.1	**38.7±0.4**	**59.2±0.5**
	No score normalization	43.0±0.2	35.0±0.3	55.4±0.3

A.3 Loss-objective ablation

Table 8 reports a loss-objective ablation each non-full arm removes one loss component: concept-forecast supervision, transition tolerance, the two SIL false-positive penalties, or classifier L_1 regularization. We designed the ablation as a component diagnostic rather than a search for a dataset-specific accuracy-maximizing objective: each term supports a different modeling requirement, and removing one changes the preferred operating point rather than uniformly improving the model. The full objective is the fixed configuration used in the main experiments. Each term can be turned off for diagnosis, but the resulting arms optimize different and sometimes competing properties: some removals improve selected accuracy values, whereas the SIL penalty reduces false positives and classifier L_1 preserves selective named-concept contributions (Tables 9 and 10).

Table 8: **Loss-objective ablation.** PE-L14, 128-concept results on the official BARISTA split, Breakfast s1, and MPII Attr, reported as mean ± sample standard deviation over seeds 42–44. Each non-full arm removes one loss component while keeping the architecture and training protocol fixed. Metrics are activity top-1 (Act.), H3 top-1 (H3-1), and H3 top-3 (H3-3) accuracy in percent. Best result per dataset and metric is bold.

	BARISTA			Breakfast s1			MPII Attr		
Loss variant	Act.	H3-1	H3-3	Act.	H3-1	H3-3	Act.	H3-1	H3-3
Full loss	76.6±2.9	67.4±0.3	81.8±1.5	51.7±0.4	**42.9±0.2**	68.9±0.7	47.0±0.6	38.7±0.6	58.5±0.4
No concept-forecast loss	78.0±2.4	67.6±0.4	82.3±1.4	51.7±0.4	42.8±0.4	**69.0±0.5**	46.9±0.5	38.6±0.7	58.6±0.5
No transition tolerance	78.0±2.7	67.3±0.5	82.5±1.5	51.7±0.4	**42.9±0.3**	**69.0±0.6**	47.2±0.4	38.6±0.7	58.4±0.7
No SIL penalty	77.9±0.6	**68.5±0.2**	**84.3±0.4**	51.5±0.4	42.7±0.5	68.8±0.6	47.0±0.5	38.3±0.7	**59.2±0.5**
No classifier L_1	**80.3±0.3**	67.7±0.5	83.9±0.2	**51.9±0.1**	42.9±0.1	68.2±0.8	**48.1±0.5**	**39.1±0.5**	58.8±0.3

Table 9 reports diagnostics targeted to the intended loss behaviors. The SIL false-positive rate measures background calibration, while predictor effective support and top-5 mass measure how selectively the shared activity predictor uses named concepts. For a predictor row w, effective support is $\|w\|_1^2/\|w\|_2^2$; lower support and higher top-5 mass indicate a more concentrated, inspectable predictor.

Removing classifier L_1 directly makes the predictor less selective: effective support expands from 31.8 to 82.4 concepts on BARISTA, from 42.6 to 83.2 on Breakfast, and from 30.5 to 81.1 on MPII, while top-5 mass falls to 11.4–12.0%. Without this penalty, each activity prediction depends on roughly 81–83 concepts, making its contribution pattern difficult to inspect; the classifier penalty is therefore needed to keep named-concept explanations selective. Removing the SIL penalty raises H3 SIL false positives from 39.1% to 43.5% on BARISTA, from 7.1% to 9.9% on Breakfast, and from 12.6% to 23.0% on MPII.

Table 9: **SIL calibration and predictor selectivity in the loss ablation.** Each cell reports H3 SIL false-positive rate ↓ / predictor effective support ↓ / predictor top-5 mass ↑. Values are mean ± sample standard deviation over seeds 42–44 from the matched PE-L14, 128-concept checkpoints. SIL-FP is a percentage over true non-SIL targets; effective support is a concept count; top-5 mass is a percentage of absolute predictor weight.

Loss variant	BARISTA (SIL-FP / N_{eff} / Top-5)	Breakfast s1 (SIL-FP / N_{eff} / Top-5)	MPII Attr (SIL-FP / N_{eff} / Top-5)
Full loss	$39.1_{\pm 5.4}$ / $31.8_{\pm 10.1}$ / $37.8_{\pm 15.8}$	$7.1_{\pm 2.2}$ / $42.6_{\pm 0.7}$ / $27.7_{\pm 0.5}$	$12.6_{\pm 1.3}$ / $30.5_{\pm 0.5}$ / $40.8_{\pm 0.3}$
No concept-forecast loss	$36.0_{\pm 4.5}$ / $26.3_{\pm 9.3}$ / $46.6_{\pm 14.5}$	$7.1_{\pm 2.4}$ / $42.8_{\pm 0.7}$ / $27.6_{\pm 0.6}$	$12.1_{\pm 1.2}$ / $30.8_{\pm 0.6}$ / $40.6_{\pm 0.4}$
No transition tolerance	$36.0_{\pm 5.8}$ / $25.9_{\pm 10.3}$ / $46.2_{\pm 14.6}$	$7.0_{\pm 2.3}$ / $42.7_{\pm 0.6}$ / $27.6_{\pm 0.5}$	$12.6_{\pm 1.2}$ / $30.3_{\pm 1.0}$ / $40.9_{\pm 0.9}$
No SIL penalty	$43.5_{\pm 0.9}$ / $20.6_{\pm 1.7}$ / $\mathbf{56.0_{\pm 1.2}}$	$9.9_{\pm 2.8}$ / $42.5_{\pm 0.6}$ / $27.8_{\pm 0.5}$	$23.0_{\pm 2.1}$ / $30.6_{\pm 0.5}$ / $40.8_{\pm 0.5}$
No classifier L_1	$31.0_{\pm 0.8}$ / $82.4_{\pm 0.3}$ / $11.6_{\pm 0.0}$	$7.3_{\pm 2.3}$ / $83.2_{\pm 0.5}$ / $11.4_{\pm 0.1}$	$13.4_{\pm 0.9}$ / $81.1_{\pm 0.8}$ / $12.0_{\pm 0.2}$

Table 10 evaluates the same loss arms with matched oracle-ranked concept-node edits. The 30 initially wrong and 30 initially correct video windows are selected once from the full-loss checkpoint for each dataset and seed, then reused for every loss arm. This isolates loss effects from changes in the sampled cases. The first value in each cell is the wrong-to-correct rate: for the full-loss arm it is 25.6%, 25.6%, and 34.4% on BARISTA, Breakfast, and MPII, respectively, matching the concept-node results in Table 3. The bold 0.0 values are the second entry, correct-to-wrong rate, and indicate that these edits caused no observed harm rather than no correction.

Table 10: **Matched oracle node edits in the loss ablation.** Budget-5 directional-oracle concept-node edits on the same 30 wrong and 30 correct video windows per dataset and seed selected from the full-loss checkpoint. Entries report wrong-to-correct rate ↑ / correct-to-wrong rate ↓ / true-class probability change ↑ in percentage points, mean ± sample standard deviation over seeds 42–44. The future concept target is privileged, so these are oracle upper bounds.

Loss variant	BARISTA (W→C / C→W / Δp)	Breakfast s1 (W→C / C→W / Δp)	MPII Attr (W→C / C→W / Δp)
Full loss	$25.6_{\pm 18.4}$ / $0.0_{\pm 0.0}$ / $7.17_{\pm 5.26}$	$25.6_{\pm 3.8}$ / $0.0_{\pm 0.0}$ / $11.35_{\pm 1.20}$	$34.4_{\pm 6.9}$ / $0.0_{\pm 0.0}$ / $9.84_{\pm 1.53}$
No concept-forecast loss	$29.9_{\pm 22.0}$ / $0.0_{\pm 0.0}$ / $9.33_{\pm 4.54}$	$23.6_{\pm 3.0}$ / $0.0_{\pm 0.0}$ / $11.89_{\pm 1.10}$	$38.4_{\pm 6.9}$ / $0.0_{\pm 0.0}$ / $11.71_{\pm 1.51}$
No transition tolerance	$31.1_{\pm 15.9}$ / $0.0_{\pm 0.0}$ / $9.10_{\pm 3.93}$	$26.9_{\pm 5.4}$ / $0.0_{\pm 0.0}$ / $11.43_{\pm 1.30}$	$36.3_{\pm 7.2}$ / $0.0_{\pm 0.0}$ / $9.81_{\pm 1.46}$
No SIL penalty	$40.7_{\pm 12.9}$ / $0.0_{\pm 0.0}$ / $10.61_{\pm 2.91}$	$25.8_{\pm 1.7}$ / $0.0_{\pm 0.0}$ / $11.56_{\pm 1.19}$	$34.1_{\pm 7.1}$ / $0.0_{\pm 0.0}$ / $10.25_{\pm 1.27}$
No classifier L_1	$42.2_{\pm 13.2}$ / $0.0_{\pm 0.0}$ / $12.06_{\pm 1.46}$	$33.0_{\pm 6.7}$ / $1.3_{\pm 2.3}$ / $13.71_{\pm 1.84}$	$42.1_{\pm 11.7}$ / $1.1_{\pm 2.0}$ / $11.26_{\pm 0.77}$

The oracle node edits improve initially wrong forecasts for every loss arm, but removing classifier L_1 produces the largest gains and also introduces small correct-to-wrong rates on Breakfast and MPII. This is consistent with the selectivity result: a less concentrated predictor is more sensitive to edits, but its edits are less controlled. The SIL and predictor diagnostics therefore expose loss effects that endpoint accuracy alone does not reveal.

A.4 Matched non-graph dynamics control

Table 11 reports the retrained graph-free control with exactly the same trainable parameter count as the graph model for each dataset. The control replaces both observed-state and recursive forecast graph layers with graph-layer-matched dense residual layers, while retaining the same concept vocabulary, losses, splits, history, horizon, and seeds. The existing dense/no-graph arm in Table 7 is not parameter matched and should not be used for this comparison.

Exact parameter matching does not produce a uniform forecasting advantage for either transition mechanism. The graph-free control is higher on BARISTA, whereas the graph is higher on Breakfast. On MPII Attr, the graph-free control is higher on activity top-1, while the graph is higher on H3 top-1 and top-3. As discussed above, the graph does not automatically improve predictive performance, but it provides an explicit path for concept propagation.

A.5 Forecasting and probability diagnostics

Table 12 reports the full-model H1–H3 diagnostics. NLL measures the negative log probability assigned to the target class, multiclass Brier measures squared probability-vector error, and 15-bin ECE measures the discrepancy between confidence and empirical correctness (Guo et al., 2017), so lower values are better. The metrics are computed over every valid H3 evaluation window.

Table 11: **Exact-capacity graph-free dynamics control.** PE-L14, 128-concept results on official BARISTA, Breakfast s1, and MPII Attr protocols, reported over seeds 42–44. The graph-free control uses dense residual transitions with exactly the graph model's trainable parameter count; all other protocol settings are held fixed.

Dataset	Dynamics	Trainable params	Activity top-1	H3 top-1	H3 top-3
BARISTA	Graph	295,153	76.56±2.89	67.36±0.30	81.78±1.52
	Exact-capacity graph-free	295,153	81.08±0.83	70.37±0.38	85.27±0.60
Breakfast	Graph	271,668	51.70±0.35	42.89±0.23	68.89±0.73
	Exact-capacity graph-free	271,668	50.64±0.80	41.29±0.92	66.75±2.61
MPII Attr	Graph	286,683	47.03±0.58	38.65±0.64	58.54±0.39
	Exact-capacity graph-free	286,683	48.27±0.47	38.05±0.79	58.01±1.23

Table 12: **Full-model forecasting and probability diagnostics.** PE-L14, 128-concept results on official BARISTA, Breakfast s1, and MPII Attr protocols for H1–H3, reported over seeds 42–44. NLL is negative log-likelihood, Brier is the multiclass Brier score, and ECE is fixed-width 15-bin expected calibration error.

Dataset	Horizon	Top-1	Top-3	NLL ↓	Brier ↓	ECE ↓
BARISTA	H1	74.8±1.1	87.2±1.3	1.054±0.109	0.357±0.026	0.093±0.044
	H2	69.5±0.9	83.6±1.3	1.174±0.071	0.406±0.012	0.039±0.006
	H3	67.4±0.3	81.8±1.5	1.284±0.059	0.444±0.009	0.064±0.002
Breakfast s1	H1	48.8±0.2	73.9±0.6	1.768±0.015	0.669±0.003	0.058±0.006
	H2	45.8±0.3	71.5±0.7	1.981±0.021	0.734±0.005	0.157±0.005
	H3	42.9±0.2	68.9±0.7	2.303±0.028	0.811±0.005	0.251±0.004
MPII Attr	H1	44.8±0.5	65.2±0.2	2.080±0.010	0.702±0.004	0.032±0.010
	H2	41.7±0.5	62.0±0.5	2.285±0.020	0.756±0.006	0.100±0.011
	H3	38.7±0.6	58.5±0.4	2.558±0.032	0.819±0.009	0.170±0.012

Table 13 isolates the effect of unlabeled windows. Rates are computed over valid H3 sliding-window test examples; non-SIL accuracy is conditioned on true non-SIL targets, and non-SIL macro-F1 averages only the non-SIL activity classes while treating SIL predictions as errors.

Table 13: **Main-dataset H3 SIL diagnostics.** Results use 128 concepts, PE-L14, the official BARISTA split, Breakfast s1, and MPII Attr, with seeds 42–44. Values are mean ± sample standard deviation across seeds; N is the number of valid H3 windows per seed and $N_{\text{non-SIL}}$ is the number whose ground-truth target is non-SIL.

Dataset	N	$N_{\text{non-SIL}}$	True SIL %	Predicted SIL %	Non-SIL acc. / macro-F1 %
BARISTA	2,415	853	64.68	72.17±2.57	25.48±2.72 / 5.65±0.73
Breakfast	16,770	15,780	5.90	7.69±2.30	44.50±0.43 / 23.66±0.27
MPII Attr	12,956	10,425	19.54	19.56±1.39	36.31±1.22 / 11.08±0.30

Table 14 provides a direct horizon-wise last-label reference for the forecasting claims. It copies the activity label at the end of the observed history to every future horizon; unlike the learned concept-persistence arm in the transition factorial, it has no trainable parameters. As the table illustrates, activities often persist across consecutive horizons, which explains why simply repeating the last observed label is such a strong baseline. Although TRACE underperforms this baseline at horizons H1 and H2, this does not mean that its forecasts are inferior. Rather, TRACE learns to predict transitions to different activity classes instead of merely preserving the previous label.

A.6 Further backbone ablations

Table 15 reports the PE-G14 result matrix; PE-G14 and PE-L14 are variants of the Perception Encoder visual representation (Bolya et al., 2025).

Within this PE-G14 matrix, opaque feature models are strongest on aggregate accuracy, while TRACE remains competitive with the interpretable alternatives. The Feature Transformer is best for H3 top-1 on all three datasets, and SlowFast-style is best for H3 top-3 on Breakfast and MPII Cooking 2; the

Table 14: **Matched last-label forecasting baseline.** Top-1 accuracy for H1–H3 on the same 128-concept PE-L14 protocols and valid windows as the forecasting diagnostics, using seeds 42–44. The baseline repeats the final observed activity label; because the targets and windows are identical across seeds, the seed-level standard deviation is zero.

Dataset	N per horizon	H1 top-1 %	H2 top-1 %	H3 top-1 %
BARISTA	2,415	80.66	67.87±0.00	62.36±0.00
Breakfast	16,770	91.12	82.61±0.00	75.01±0.00
MPII Attr	12,956	81.24	66.89±0.00	57.25±0.00

Table 15: **PE-G14 128-concept result matrix.** PE-G14 results on the official BARISTA split, Breakfast s1–s4, and MPII Attr/Dishes, with activity classification and H3 forecast accuracy reported over seeds 42–46. Multi-split datasets are averaged within seed. Best result per dataset and metric is bold; second best is underlined.

		BARISTA				Breakfast				MPII Cooking 2			
		Activity		H3		Activity		H3		Activity		H3	
Group	Method	Top-1	Top-3	Top-1	Top-3	Top-1	Top-3	Top-1	Top-3	Top-1	Top-3	Top-1	Top-3
Linear	Sparse Linear	68.6±0.7	79.0±0.4	68.8±0.3	82.9±0.6	46.4±0.4	70.9±0.4	39.6±0.3	64.2±0.4	47.3±0.5	66.2±0.7	41.1±0.7	62.2±0.2
Concept-based	MoTIF	73.7±2.4	87.1±1.1	n/a	n/a	47.8±0.4	72.1±0.4	n/a	n/a	49.6±1.4	69.8±0.6	n/a	n/a
	TRACE (ours)	76.2±0.5	89.3±0.3	68.0±0.1	82.1±0.5	48.7±0.3	**73.4±0.1**	40.0±0.3	64.7±0.3	51.4±0.5	71.4±0.2	42.0±0.6	61.3±0.3
Black-box	Feature Transformer	**82.0±0.4**	**91.5±0.2**	**71.3±0.3**	**84.3±0.2**	**50.6±0.4**	72.8±0.4	**42.3±0.4**	**67.0±0.3**	**52.2±0.3**	**71.7±0.3**	**42.4±0.4**	62.4±0.4
	SlowFast-style	81.9±0.5	91.3±0.4	69.6±0.5	83.5±0.4	49.4±0.4	72.2±0.6	41.6±0.3	66.9±0.2	51.2±0.4	71.1±0.6	41.9±0.5	**62.8±0.5**

two black-box models are nearly tied on BARISTA activity accuracy. Among concept-based models, TRACE improves activity top-1 over MoTIF on BARISTA and MPII Cooking 2 and provides the comparable H3 concept rollout that MoTIF does not report.

A.7 Concept ablations: 128, 256

Table 16 isolates TRACE and compares the two concept budgets. This comparison tests concept budget, not independently generated vocabularies.

Table 16: **Matched proposed-model concept-budget comparison.** PE-L14 proposed-model results for 128 versus 256 concepts on official BARISTA, Breakfast s1–s4, and MPII Attr/Dishes, reported as mean ± sample standard deviation over seeds 42–46. Multi-split datasets are averaged within seed.

Dataset	128 activity	256 activity	128 H3	256 H3
BARISTA	76.02±2.18	79.71±0.42	67.55±0.35	68.51±0.33
Breakfast	45.52±0.23	46.36±0.21	37.48±0.08	38.91±0.19
MPII Attr/Dishes	46.15±0.43	47.68±0.41	37.88±0.46	39.40±0.30

Table 17 reports the 256-concept result matrix.

The 256-concept results provide an accuracy-oriented operating point, with especially strong BARISTA activity recognition and Breakfast/MPII H3 forecasting.

A.8 Additional datasets: GTEA and EPIC-KITCHENS-100

Table 18 reports results for the two diagnostic stress-test datasets. The results confirm that aggregate accuracy alone is misleading on these datasets. On GTEA, the models predict SIL for approximately 86–98% of H3 windows despite a true rate of 49.41%. On EPIC-KITCHENS-100, the black-box baselines predict SIL for approximately 67–74% of H3 windows, with 74% for Sparse Linear and 67% for the proposed model; all remain far above the true rate of 25.22%. Consequently, H3 top-1 accuracy near the SIL prevalence does not establish useful procedural anticipation. We therefore retain GTEA and EPIC-KITCHENS-100 as dataset-resolved calibration diagnostics rather than including them in the main aggregate comparison. However, it is important to note that TRACE outperforms most of the other methods in activity classification while remaining close to black-box performance in forecasting.

Table 17: **PE-L14 256-concept result matrix.** PE-L14 results with 256 concepts on the official BARISTA split, Breakfast s1–s4, and MPII Attr/Dishes, with activity classification and H3 forecast accuracy reported over seeds 42–46. Best result per dataset and metric is bold; second best is underlined.

		BARISTA				Breakfast				MPII Cooking 2			
		Activity		H3		Activity		H3		Activity		H3	
Group	Method	Top-1	Top-3	Top-1	Top-3	Top-1	Top-3	Top-1	Top-3	Top-1	Top-3	Top-1	Top-3
Linear	Sparse Linear	67.4±0.6	79.7±0.3	68.3±0.1	82.4±0.3	43.7±0.1	68.3±0.1	37.7±0.2	62.0±0.2	43.9±0.5	61.3±0.9	38.3±0.5	58.8±0.7
Concept-based	MoTIF	75.7±4.4	87.7±2.1	n/a	n/a	45.7±0.5	70.4±0.5	n/a	n/a	45.6±1.0	66.7±0.7	n/a	n/a
	TRACE (ours)	**79.7±0.4**	90.2±0.2	68.5±0.3	**83.6±0.2**	46.4±0.2	**71.2±0.2**	38.9±0.2	63.9±0.3	47.7±0.4	67.2±0.3	39.4±0.3	59.5±0.7
Black-box	Feature Transformer	79.2±0.3	89.3±0.5	**69.6±0.3**	82.9±0.3	**47.3±0.4**	69.6±0.6	39.7±0.6	63.7±0.2	**49.1±1.0**	**69.6±0.7**	**40.3±0.9**	61.0±0.5
	SlowFast-style	79.6±0.5	**90.3±0.9**	67.6±1.0	82.5±0.8	46.8±0.5	69.5±0.6	**39.7±0.6**	**64.1±0.4**	48.1±0.8	69.2±0.7	39.6±0.4	**61.0±0.4**

Table 18: **GTEA and EPIC-KITCHENS-100 diagnostic comparison.** PE-L14, 128-concept results on GTEA s1 verb targets and EPIC-KITCHENS-100 s1 action targets, with activity classification, H3 forecast accuracy, and H3 SIL calibration reported over seeds 42–46.

		Activity		H3		H3 SIL diagnostics	
Dataset	Method	Top-1	Top-3	Top-1	Top-3	Predicted rate	False-positive rate
GTEA	Sparse Linear	21.7±2.5	41.5±3.3	**49.3±0.4**	62.5±0.5	98.02±0.85	97.30±0.90
	MoTIF	**49.5±0.1**	62.1±0.4	n/a	n/a	n/a	n/a
	TRACE (ours)	49.2±0.2	**62.5±0.2**	45.1±0.6	61.7±0.4	85.51±1.56	**82.62±1.76**
	Feature Transformer	45.5±2.3	61.3±0.8	47.6±1.9	61.9±0.7	92.90±3.61	91.25±3.31
	SlowFast-style	46.7±1.6	62.1±0.9	48.2±0.6	**62.6±0.4**	94.03±2.38	92.59±3.07
EPIC-KITCHENS-100	Sparse Linear	24.1±3.0	33.4±3.2	25.33±0.18	36.4±0.2	73.89±0.89	71.24±0.98
	MoTIF	24.7±0.4	36.5±0.3	n/a	n/a	n/a	n/a
	TRACE (ours)	**26.6±0.2**	**39.0±0.2**	25.09±0.18	**37.0±0.3**	66.70±0.35	**63.10±0.34**
	Feature Transformer	26.3±0.3	36.8±0.5	25.34±0.22	35.3±0.4	67.08±1.69	63.54±1.76
	SlowFast-style	26.5±1.1	38.6±0.9	**25.68±0.60**	36.7±0.3	69.15±3.79	65.61±4.03

A.9 Additional dataset protocols

Table 19 reports TRACE on every benchmark protocol used in the five-seed evaluation. The main paper uses the BARISTA official split, averages Breakfast official s1–s4 within each seed, and uses the MPII Attr protocol to match the principal comparison. The per-protocol rows below, the MPII dish protocol, GTEA Gaze, and EPIC-KITCHENS-100 provide additional cross-protocol evidence. GTEA uses verb targets (Li et al., 2021), whereas EPIC-KITCHENS-100 uses composite verb–noun action targets (Damen et al., 2020).

Table 19: **Proposed-model protocol sweep.** PE-L14 proposed-model results with 128-concept vocabularies on the individual official protocols (BARISTA official, Breakfast s1–s4, MPII Attr/Dishes, GTEA s1 verb, and EPIC-KITCHENS-100 s1 action), reported over seeds 42–46. Entries are mean ± standard deviation over five seeds in percent. The matched PE-L14 multi-method matrix in Table 1 is a different run, so its entries are not expected to match these protocol-specific values.

Dataset	Protocol	Activity		H3	
		Top-1	Top-3	Top-1	Top-3
BARISTA	official split	76.6±1.2	89.0±0.5	67.3±0.3	81.2±0.5
Breakfast	official s1	50.4±0.8	75.3±0.6	43.8±0.5	67.8±0.7
Breakfast	official s2	35.9±0.6	60.4±0.4	29.3±0.7	52.4±0.6
Breakfast	official s3	44.0±0.7	69.4±0.4	37.4±0.7	61.2±1.0
Breakfast	official s4	47.8±0.8	73.8±0.8	41.9±0.8	67.9±0.5
MPII Cooking 2	Attr	46.5±0.5	67.3±0.4	39.0±0.3	58.7±0.3
MPII Cooking 2	Dishes	45.1±0.4	67.5±0.6	37.3±0.6	58.1±0.6
GTEA Gaze	verb	49.4±0.1	62.6±0.3	48.3±0.5	62.7±0.1
EPIC-KITCHENS-100	action	26.7±0.2	38.8±0.3	25.9±0.2	36.9±0.3

Performance varies substantially across official folds and target surfaces. For example, Breakfast H3 top-1 ranges from 29.3% on official s2 to 43.8% on official s1. The GTEA and EPIC-KITCHENS results should also be interpreted together with their large unlabeled regions and SIL calibration, rather than compared directly with the main action-recognition results.

A.10 SYNTHETIC DATASET EVALUATION

Table 20 reports a controlled sequence task containing six known delayed concept relations, the concrete implementation of the synthetic dataset is provided in the supplementary material. The experiment isolates whether the temporal graph can exploit a dependency that is unavailable to a current-only model and difficult to represent with flat sparse dynamics.

The temporal graph reaches 77.6% H3 accuracy, compared with 6.6% when the graph is disabled and 7.6% for sparse linear dynamics. It also retrieves 83.3% of the planted relations among its active forecast edges. Precision is lower at 19.8%, showing that predictive success does not imply exact recovery of the generating graph. The experiment establishes that the implementation can exploit and partially identify delayed structure under controlled conditions, but it does not establish causal recovery on real videos.

Table 20: **Synthetic delayed-dependency diagnostic.** Six-relation synthetic delayed-dependency task, trained with seeds 42–44. Activity accuracies, active-edge recall, and active-edge precision are reported in percent.

Model	H1	H3	Edge recall	Edge precision
Temporal graph	$75.0_{\pm 4.0}$	$77.6_{\pm 1.0}$	$83.3_{\pm 16.7}$	$19.8_{\pm 4.3}$
History, graph disabled	$6.6_{\pm 0.2}$	$6.6_{\pm 0.6}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$
Current-only, graph disabled	$6.6_{\pm 0.2}$	$6.6_{\pm 0.6}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$
Sparse linear dynamics	$7.6_{\pm 0.2}$	$7.6_{\pm 0.2}$	n/a	n/a

B GRAPH PROPERTIES

B.1 EFFECTIVE ROLLOUT DEPENDENCY CONCENTRATION

To assess how localized the learned rollout dependencies are, we compute the absolute rollout Jacobian from the input concept state to the predicted concept state, normalize parent contributions, and summarize the resulting influence distribution over eight held-out windows per dataset and seed. Table 21 reports the effective number of influential parents (N_{eff}), cumulative influence in the five strongest parents (Top-5), same-concept influence (Diag), effective density above a 1% influence threshold, and the number of parents needed to explain 90% of influence.

Table 21: **Effective rollout dependency concentration.** PE-L14, 128-concept graph and dense-forecast-rollout checkpoints on official BARISTA, Breakfast s1, and MPII Attr, evaluated on eight held-out windows per dataset and seed for seeds 42–44. Each entry is graph → dense. N_{eff} and 90% support are parent counts; Top-5, Diag, and Density are percentages. Density is the fraction of dependencies carrying at least 1% of an output's normalized influence, not the structural connectivity of the dense model.

Dataset	Horizon	N_{eff}	Top-5 (%)	Diag (%)	Density (%)	90% support
BARISTA	H1	1.9 → 72.7	93.0 → 23.9	90.7 → 15.6	1.1 → 23.5	2.3 → 79.8
	H3	4.1 → 89.1	82.6 → 16.2	76.9 → 6.7	4.8 → 27.8	13.8 → 81.8
Breakfast s1	H1	2.4 → 77.4	89.7 → 21.8	85.9 → 13.1	2.3 → 24.7	5.7 → 80.1
	H3	6.2 → 90.7	75.0 → 15.3	66.5 → 5.7	10.8 → 28.4	16.2 → 82.1
MPII Attr	H1	2.2 → 69.6	90.7 → 25.5	87.3 → 17.3	1.9 → 22.6	4.9 → 78.8
	H3	5.5 → 86.6	77.3 → 17.4	69.5 → 7.7	9.7 → 27.2	14.9 → 81.1

The dense rollout is substantially more entangled by every concentration measure. At H3, its effective parent count is 86.6–90.7, compared with 4.1–6.2 for the graph rollout; only 15.3–17.4% of dense influence lies in the five strongest parents, compared with 75.0–82.6% for the graph. Same-concept influence falls to 5.7–7.7% in the dense model, so more than 92% of its effective influence is off-diagonal, whereas the graph retains 66.5–76.9% same-concept influence. The dense rollout also requires 81.1–82.1 parents to explain 90% of influence, versus 13.8–16.2 for the graph. Thus, the dense arm is not merely harder to visualize: its learned dependency structure is less concentrated and less concept-local. The reported density is effective Jacobian density at the 1% threshold; the dense structural connectivity remains 100%.

B.2 Graph robustness and limits of causal interpretation

Table 22 compares the active-learned edge sets between all pairs of seeds for each main dataset. Jaccard overlap measures whether the same edges are selected, while sign agreement is computed only on edges active in both seeds.

Table 22: **Pairwise graph stability.** PE-L14, 128-concept graph stability on the official BARISTA, Breakfast s1, and MPII Attr protocols, computed across seeds 42–44. Values are means across seed pairs in percent.

Dataset	Edge-set Jaccard	Shared-edge sign agreement
BARISTA	5.3	88.5
Breakfast	11.9	98.8
MPII Cooking 2 Attr	11.7	98.9

The exact selected edge sets have low overlap across seeds, especially on BARISTA. However, edges that recur across seeds usually retain their sign. The learned graphs therefore provide seed-specific sparse explanations with relatively consistent directions among shared relations, rather than a unique recovered graph. The synthetic experiment above tests known delayed dependencies, whereas the real-video stability analysis is a reproducibility diagnostic; neither identifies causal structure in the video-generating process.

B.3 Forecasting diagnostics

We evaluate H1–H3 forecasts to assess error accumulation under recursive rollout and the reliability of decoded activity confidences. Top-1 accuracy declines consistently but moderately with increasing horizon (Fig. 4). BARISTA remains robust at H3 and exhibits comparatively well-aligned confidence estimates, whereas Breakfast and MPII are both less accurate and increasingly overconfident, motivating calibration-aware use of predicted probabilities in less predictable settings.

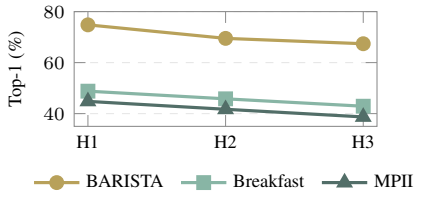

Figure 4: **Forecasting diagnostics.** Top-1 accuracy across forecast horizons H1-H3.

B.4 Long-horizon rollout

We test the same learned H3 transition recursively for up to 50 future windows without retraining it for the longer horizons. This experiment therefore evaluates zero-shot rollout stability rather than direct H50 supervision. Table 23 reports activity top-1 accuracy at representative horizons and compares H50 with a persistence baseline that repeats the last observed activity with 500 test windows.

Table 23: **Zero-shot recursive rollout from H3 to H50.** PE-L14, 128-concept rollout on the official BARISTA, Breakfast s1, and MPII Attr protocols, with entries reported over seeds 42–44 in percent. Persist. is the H50 last-observation baseline, and Sat. is the fraction of H50 concept activations near their range boundaries.

Dataset	H3	H10	H20	H30	H40	H50	Persist.	Sat.
BARISTA	68.4±0.2	60.0±3.5	37.6±22.8	31.2±27.7	27.3±29.6	25.5±27.4	39.6±0.0	56.9±5.4
Breakfast	45.7±4.5	40.0±1.9	36.2±4.4	34.0±4.1	31.3±3.3	27.3±2.7	32.7±3.6	56.7±1.6
MPII Cooking 2 Attr	38.3±1.9	21.1±1.1	10.6±1.1	10.7±0.2	9.1±2.6	9.5±0.9	20.2±1.7	52.9±2.1

Accuracy degrades with recursive depth on every dataset. BARISTA remains relatively strong through H10, but its variance increases sharply beyond H20, indicating seed-dependent rollout instability. At H50, TRACE falls below persistence on all three datasets and between 52.9% and 56.9% of concept activations are saturated. The corresponding H50 concept mean absolute errors are 0.426, 0.409, and 0.414 for BARISTA, Breakfast, and MPII Cooking 2 Attr. Thus, the learned dynamics remain informative for moderately longer forecasts but do not support a claim of reliable 40–50-step prediction.

C ALGORITHMIC DESCRIPTION

C.1 VLM CONCEPT PROPOSAL WITH THE GENERATION PROMPT

Concept vocabulary construction is offline and label-blind at the visual-observation stage, following the broader direction of label-free and language-guided concept construction (Oikarinen et al., 2023; Yang et al., 2023). Qwen3.5-27B Team (2026) receives four-frame storyboards sampled in a label-balanced manner, returns structured visible objects, states, and interactions, and then proposes short reusable concepts. Candidate phrases are deduplicated, rejected when they exactly match a normalized activity label, retrieved with text–image similarity, and retained only after image-grounded verification on at least two videos. Each candidate includes supporting storyboard identifiers and a short factual visual rationale before retrieval and verification. Concept embeddings are matched to training-window embeddings by cosine similarity in the shared vision–language space (Radford et al., 2021; Bolya et al., 2025), which supplies the visual evidence used by the verification stage. The storyboard analysis is label-blind; activity labels are attached only afterward for coverage auditing and for the proposal-stage instruction not to restate labels. The compact prompt template below reproduces the operative constraints.

```
You are building a reusable visual concept vocabulary
for dataset [dataset]. Propose exactly [N] short
concepts grounded in the supplied label-blind storyboard
observations. Each phrase must contain 1--3 words and be
reusable across videos. Prefer concrete objects/materials,
visible states/attributes, and spatial or hand-object
relations. Include an action primitive only when motion
is directly supported across the four frames. Do not
repeat, restate, or semantically paraphrase an activity
label. Do not use SIL/background as a concept. For every
candidate, cite supporting storyboard IDs and give one
factual rationale of at most 16 words. Return the required
JSON schema only.
```

C.2 TRAINING AND ROLLOUT ALGORITHMS

The algorithms below summarize the two implementation-critical pipelines: offline vocabulary construction and autoregressive concept rollout.

Algorithm 1 gives the offline procedure for proposing, grounding, deduplicating, and exporting the textual concept vocabulary.

For the reported generation runs, $(C, N, K, R, K_r, V, \delta, s) = (128, 80, 24, 12, 6, 2, 0.95, 42)$; incomplete proposal JSON is retried with a larger token budget.

Algorithm 2 provides our method's procedure for autoregressive forecasting, teacher forcing, and evaluation-time interventions.

The forecast graph module $\mathcal{G}_{\text{fut}}$ is reused as a one-step transition for each autoregressive forecast step; persistence ablations replace it with the identity. Interventions are applied at the corresponding concept, activity-belief, or edge hook, and oracle edit ranking is used only for evaluation.

Algorithm 1 VLM-based textual concept generation.

Input: training-window index with frozen PE-L14 features, Qwen3.5-27B, seed s, target size C
Input: proposal count N, round size K, number of rounds R, retrieval count K_r, verification threshold V, similarity threshold δ
Initialize candidate set $\mathcal{Q} \leftarrow \varnothing$ and label counts
for $r = 1, \ldots, R$ **do**
  Sample K balanced training windows with $\mathrm{Random}(s + r)$; render four-frame storyboards
  Parse storyboards with the label-blind VLM prompt; attach labels only for coverage auditing
  Propose N grounded 1–3 word concepts with categories, evidence IDs, and rationales
  Reject malformed, overlong, duplicate, exact-label, unsupported, or invalid-evidence proposals

  for each accepted proposal q **do**
    Retrieve up to K_r nearest windows from distinct videos using PE-L14 cosine similarity
    VLM-verify the retrieved windows; retain q iff at least V windows support it
  end for
  Select grounded candidates by utility (verification confidence, visual support, category coverage, and brevity), remove text-embedding duplicates above δ, and save round state
  if $|\mathcal{Q}| \geq C$ **then**
    break
  end if
end for
Validate count, phrase length, uniqueness, label exclusion, grounding, and redundancy
Output: concept vocabulary $\mathcal{C}$ and text embeddings

Algorithm 2 Autoregressive concept rollout and training-time teacher forcing.

Input: concept scores $\mathbf{c}_{1:T}$, validity mask $\mathbf{m}_{1:T}$, graph modules $\mathcal{G}_{\mathrm{shared}}, \mathcal{G}_{\mathrm{act}}, \mathcal{G}_{\mathrm{fut}}$, activity head $(\mathbf{W}, \mathbf{b})$, activity-feedback matrix $\mathbf{A}_{\mathrm{act}}$, history length L_{hist}, lag gates $\{\lambda_\delta\}_{\delta=1}^{L_{\mathrm{hist}}}$, and horizon H
Optional: interventions $\mathcal{I}$, training labels $y_{1:T+H}$, and teacher-forcing ratio ρ
$\mathbf{z}_{1:T} \leftarrow \mathrm{Normalize}(\mathbf{c}_{1:T}) \odot \mathbf{m}_{1:T}$; apply interventions at their configured concept, edge, or activity-belief hook
$\mathbf{r}^{\mathrm{shared}}_{1:T} \leftarrow \mathcal{G}_{\mathrm{shared}}(\mathbf{z}_{1:T})$
$\mathbf{r}^{\mathrm{act}}_{1:T} \leftarrow \mathcal{G}_{\mathrm{act}}(\mathbf{r}^{\mathrm{shared}}_{1:T})$; $\mathbf{r}^{\mathrm{fut}}_{1:T} \leftarrow \mathbf{r}^{\mathrm{shared}}_{1:T}$
Compute observed activity beliefs $\hat{\mathbf{y}}_s \leftarrow \mathrm{softmax}\big(\phi(\mathbf{r}^{\mathrm{act}}_s)\mathbf{W}^\top + \mathbf{b}\big)$ for valid $s \leq T$
Compute d^{hist}_T using Eq. (5), and obtain $\widetilde{\mathbf{r}}^{\mathrm{act}}_T$ and $\widetilde{\mathbf{r}}^{\mathrm{fut}}_T$ using Eq. (6)
Set $\widetilde{\mathbf{r}}^{\alpha}_s \leftarrow \mathbf{r}^{\alpha}_s$ for $s < T$ and $\alpha \in \{\mathrm{act}, \mathrm{fut}\}$
$\hat{\mathbf{y}}_T \leftarrow \mathrm{softmax}\big(\phi(\widetilde{\mathbf{r}}^{\mathrm{act}}_T)\mathbf{W}^\top + \mathbf{b}\big)$
for $h = 1, \ldots, H$ **do**
  $t \leftarrow T + h - 1$
  During training, replace each available source belief $\hat{\mathbf{y}}_s$ with $\mathrm{TeacherForce}(\hat{\mathbf{y}}_s, \mathrm{onehot}(y_s); \rho)$; otherwise leave $\hat{\mathbf{y}}_s$ unchanged
  $d^{\mathrm{hist}}_{t+1} \leftarrow \sum_{\delta=1}^{L_{\mathrm{hist}}} \sigma(\lambda_\delta) m_{t+1-\delta} \hat{\mathbf{y}}_{t+1-\delta} \mathbf{A}_{\mathrm{act}}$
  $\widetilde{\mathbf{r}}^{\mathrm{fut}}_{t+1} \leftarrow \sigma\big(\phi\big([\mathcal{G}_{\mathrm{fut}}(\widetilde{\mathbf{r}}^{\mathrm{fut}}_{1:t})]_{t+1}\big) + d^{\mathrm{hist}}_{t+1}\big)$
  $\hat{\mathbf{y}}_{t+1} \leftarrow \mathrm{softmax}\big(\phi(\widetilde{\mathbf{r}}^{\mathrm{fut}}_{t+1})\mathbf{W}^\top + \mathbf{b}\big)$; $m_{t+1} \leftarrow 1$
end for
if training **then**
  Form stop-gradient future concept targets using Eq. (1) and optimize L from Eq. (9), with detailed terms in Eqs. (10)–(14)
end if
Output: current and future activity predictions, forecast concept states, and intervention diagnostics

D HYPERPARAMETERS AND REPRODUCIBILITY

This section specifies the complete training configuration used for the main 128-concept comparison. Unless stated otherwise, the same settings are used for Sparse Linear, MoTIF, the proposed model,

the Feature Transformer, and the SlowFast-style baseline. The 256-concept experiment changes only the dataset-specific concept vocabulary and the resulting concept dimensionality; the black-box models operate on the frozen visual features and are therefore independent of the concept budget.

D.1 Dataset and baseline details

Datasets and targets. Breakfast and MPII Cooking 2 cover cooking and daily manipulation activities, whereas BARISTA focuses on fine-grained beverage preparation. The five-seed main comparison uses the BARISTA official split, averages Breakfast s1–s4 and MPII Cooking 2 Attr within each seed, and evaluates action-label targets with SIL assigned to otherwise unlabeled windows; the dataset protocols follow the corresponding benchmark descriptions (Knab et al., 2026c; Kuehne et al., 2014; Rohrbach et al., 2015). GTEA Gaze uses verb targets (Li et al., 2021) and EPIC-KITCHENS-100 uses composite verb–noun action targets (Damen et al., 2020); their many unlabeled windows make them diagnostic stress tests rather than part of the main aggregate comparison. Their true H3 SIL rates are 49.41% and 25.22%, respectively, and their dataset-resolved results appear in Appendix A.8 and Appendix A.9.

Perception Encoder video embeddings. All datasets are extracted once with pretrained Perception Encoder (Bolya et al., 2025) backbones and are kept frozen. We partition each video into non-overlapping 32-frame windows; the final non-empty partial window is retained. For each window, eight frames are sampled uniformly, including the temporal endpoints when the window contains at least eight frames; shorter windows are completed by repeating their last frame. The frames are converted to RGB and processed with the official PE image transform at the model's 336×336 input resolution, stacked as a $(8, 3, 336, 336)$ clip, and passed to the model's video encoder. The resulting 1024-dimensional clip feature is computed in evaluation mode without gradients. The named concepts (in text) are embedded with the same PE text encoder and tokenizer.

Baselines. The zero-shot baseline is training-free and predicts the activity label whose text prompt "a video of label" has the highest cosine similarity to each frozen video-window feature, using the same vision–language embedding principle as CLIP (Radford et al., 2021) and PE-L14 (Bolya et al., 2025). It uses no concept vocabulary, learned activity head, or temporal forecasting module, and therefore reports current-activity top-1/top-3 accuracy only. Sparse Linear uses the proposed model's shared activity predictor with a sparse linear concept-dynamics head whose sparsity budget matches the active graph-edge count. MoTIF is a concept-based activity-recognition baseline without a comparable concept-rollout H3 endpoint under our protocol (Knab et al., 2026a). The SlowFast-style temporal convolutional model and feature transformer operate directly on the frozen video-window features before concept projection, without concept activations or graph structure (Feichtenhofer et al., 2019; Bertasius et al., 2021). Thus, the black-box baselines test opaque temporal modeling over the same visual evidence, while Sparse Linear tests whether sparse concept dynamics alone explain the graph-based model's forecasting behavior.

D.2 Further implementations details

D.2.1 Shared optimization settings

Table 24 lists the shared optimization and loss hyperparameters. Models are trained with Adam without a learning-rate scheduler. The checkpoint with the highest validation H3 top-3 accuracy is retained, with early stopping after 30 epochs without improvement. Horizon losses are averaged uniformly over H1–H3. Activity and forecast supervision use cross-entropy, while future concept supervision uses Smooth L1 loss. The SIL false-positive terms add a penalty whenever SIL is predicted for a non-SIL target.

D.2.2 Method-specific architectures

Table 25 reports all method-specific parameters outside the shared configuration. Train-split standardization is fitted separately for the Sparse Linear and raw-feature black-box inputs, and the resulting statistics are applied unchanged to validation and test data.

Table 24: **Shared optimization and loss settings.** Shared optimization, sampling, and loss hyperparameters that are active in the reported runs.

Hyperparameter	Value	Hyperparameter	Value
Optimizer	Adam	Learning rate	1×10^{-4}
Weight decay	1×10^{-4}	Batch size	256
Maximum epochs	200	Early-stopping patience	30 epochs
Selection metric	Validation H3 top-3	Random seeds	42, 43, 44, 45, 46
TF32	Enabled	Observed history	5 windows
Forecast horizons	H1, H2, H3	Frames per window	32
Window sampling	Random training windows	Training sampler	Uniform
Activity loss	Cross-entropy	Forecast loss	Cross-entropy
Activity SIL-FP penalty	0.5	Forecast SIL-FP penalty	0.5
Classifier ℓ_1 weight	10^{-3}	Horizon loss weights	Uniform

Table 25: **Method-specific architecture settings.** Method-specific architecture and training hyperparameters for the main comparison.

Method	Hyperparameters
Sparse Linear	Shared activity/forecast predictor; sparse concept-dynamics matrix; top-$k = 20$ incoming dynamics connections per target concept; future-concept Smooth L1 weight 0.5.
MoTIF	One per-concept temporal transformer layer; channel width/dimension 1; dropout 0.1; shared activity head enabled; transition-tolerant forecast radius 1 with mixing weight 0.25. The paper reports its activity endpoint only because its forecast is not a comparable explicit concept rollout.
Feature Transformer	Input projection to 256 dimensions; two Transformer encoder layers; four attention heads; feed-forward dimension 512; GELU activation; dropout 0.1; learned positional embedding over the five observed windows; last valid token pooling.
SlowFast-style	Input projection to 256 dimensions; two fast temporal-convolution layers; kernel size 3; GELU activation; dropout 0.1; slow-path stride 2; last-valid fast pooling and masked-mean slow pooling; concatenated 512-dimensional representation.
Proposed model	Graph and rollout settings are enumerated in Table 26; all parameters are selected during the single joint-training run.

Both black-box architectures use one activity head and separate linear forecast heads for H1–H3. Their inputs are the same frozen window features used upstream of concept projection in the concept-based models.

D.2.3 Graph and rollout settings

The complete active and implementation-default configuration for TRACE is given in Table 26. The observed concept sequence is refined by the graph, and future concepts are then recursively forecast for H1–H3.

D.3 Runtime and complexity

Frozen window embeddings are computed once before model training and can be reused across models. For a sequence of N observed windows, an H-step forecast, C concepts, K activity classes, and l graph message-passing layers applied per rollout step, the initial shared refinement and current-activity decoding cost $O(lNC^2 + NKC)$ time. Historical activity feedback adds at most $O(L_{\text{hist}}KC)$ time.

The current implementation stores graph and activity-feedback matrices as dense tensors and applies top-k masks to their entries. Thus, the masks control the active learned messages but do not reduce the asymptotic cost of the dense matrix multiplications. At every forecast step, the rollout applies l graph layers and the shared activity head to the full N-slot state. The H-step rollout therefore costs

$$O\big(HN\left(lC^2 + KC\right)\big),$$

where the KC term includes both activity decoding and activity-to-concept feedback across all N positions. We use $l = 1$ and $L_{\text{hist}} = 3$ in the reported configuration.

The implementation retains the full concept-state and activity-belief sequence at every rollout step, requiring $O(HN(C + K))$ additional storage. Vocabulary generation is an offline, one-time VLM

Table 26: **Proposed-model graph and rollout settings.** Graph, rollout, and feedback hyperparameters for the proposed model.

Hyperparameter	Value	Hyperparameter	Value
Future concept horizons	1, 2, 3	Representation	Refined concept state only
Concept forecast loss	Smooth L1, weight 0.5	Concept-loss dead zone	0 standard deviations
Teacher forcing schedule	$0.5 \rightarrow 0$ linearly	Transition-tolerant loss	Radius 1, weight 0.25
Observed refiner	Graph	Forecast rollout mode	Fixed-length recursive
Shared graph layers	1	Task-specific graph layers	1
Spatial top-k	20	Spatial soft threshold	0
Same-concept temporal edges	Enabled	Temporal top-k	20
Temporal soft threshold	0	Cross-temporal edges	Enabled
Cross-temporal top-k	20	Cross-temporal threshold	0
Edge-gate initialization	0	Edge reporting threshold	0.2
Residual-gate initialization	0	Message scale	1
State activation	Bounded logit	Prediction transform	Logit
Forecast context	Flat ordered history	Activity context	Flat ordered history
Video-level pooling	None	Past-context mode	None
Motif-z attention layers	0	Concept-score normalizer	Enabled
Top-k training mode	Gradual	Warm-up / ramp	20 / 30 epochs
Activity feedback mode	Sparse label-to-concept	Feedback top-k	40
Feedback gate initialization	0	Feedback history	3 steps
History-gate initialization	0	Feedback hidden dimension	64
Feedback ridge coefficient	0.1	Probability epsilon	10^{-4}
Prototype smoothing	1	Transition prior	Disabled

and retrieval procedure and is not part of forecasting-time cost. All costs are reported per video; for a batch of B videos, multiply time costs by B. We report asymptotic costs rather than hardware-specific wall-clock measurements; all experiments were conducted on NVIDIA RTX PRO 6000 GPUs.

D.4 Concept-vocabulary reproducibility audit

Table 27 audits each main dataset's fixed vocabulary before interpreting downstream graph or intervention results. The audit checks normalized textual duplicates, empirical activation prevalence, direct lexical overlap with target labels, and automated semantic-target warnings from the generation pipeline.

Table 27: **128-concept vocabulary audit.** Automated audit of the 128-concept vocabularies. Lexical overlap and semantic warnings are fractions of concepts in percent. Degenerate concepts are near-zero or near-always-active concepts.

Dataset	Duplicates	Degenerate	Lexical overlap	Semantic warnings
BARISTA	0	0	0.8	7.8
Breakfast	0	0	0.0	1.6
MPII Cooking 2 Attr	0	0	0.0	5.5

The audit finds no normalized duplicates and no concepts with degenerate empirical prevalence. Direct lexical target overlap is absent on Breakfast and MPII Cooking 2 Attr and occurs for one BARISTA concept. The automated semantic check flags a small minority of concepts for possible target leakage or ambiguity, with the highest rate on BARISTA. These checks support vocabulary hygiene, but frozen-encoder activations and VLM verification are automated evidence rather than human concept ground truth.

E Detailed training objective

This appendix provides the full definitions of the loss terms summarized in Section 3.4. For each future horizon h, we define $p_{T+h,a} = \mathrm{softmax}(\boldsymbol{\ell}_{T+h})_a$ as the predicted probability of activity a.

$$\mathcal{L}_{\text{act}} = \text{CE}(\boldsymbol{\ell}_T, y_T) + \beta_{\text{SIL}} \mathcal{L}_{\text{SIL},T}, \tag{10}$$

$$\mathcal{L}_{\text{forecast}} = \sum_{h=1}^{H} \alpha_h \Big[(1 - \gamma_{\text{tol}}) \, \text{CE}(\boldsymbol{\ell}_{T+h}, y_{T+h}) + \gamma_{\text{tol}} \mathcal{L}_h^{\text{tol}} + \beta_{\text{SIL}} \mathcal{L}_{\text{SIL},T+h} \Big], \tag{11}$$

$$\mathcal{L}_{\text{concept}} = \sum_{h=1}^{H} \alpha_h \, \text{SmoothL1}\Big(\mathbf{S}_T^{(h)}, \mathbf{z}_{T+h}^{\text{target}} \Big), \tag{12}$$

$$\mathcal{L}_h^{\text{tol}} = -\log \sum_{a \in \mathcal{A}_h^{(r)}} p_{T+h,a}, \tag{13}$$

$$\mathcal{A}_h^{(r)} = \{ y_{T+h}, \ldots, y_{T+\min(H,h+r)} \}. \tag{14}$$

E.1 Intervention protocol and policy-reproduction details

The intervention evaluation covers BARISTA, Breakfast s1, and MPII Cooking 2 Attr, using seeds 42–44, 30 initially correct cases, and 30 initially incorrect H3 cases for every dataset–seed pair. The evaluation follows established concept-intervention protocols while extending them to recursive forecasts (Koh et al., 2020; Shin et al., 2023; Espinosa Zarlenga et al., 2023). The full evaluation records true-class probability change, label-flip rate, wrong-to-correct rate, correct-to-wrong rate, and future-concept displacement for each cumulative prefix. Table 3 reports the primary interface comparison.

E.1.1 Edit semantics and oracle information

All edits are applied only at evaluation time to the frozen checkpoint, and each reported prefix is re-rolled out through H3 before scoring. For a concept-node edit, the oracle ranks candidate concept coordinates by their directional contribution to the held-out H3 class relative to the predicted H1 state. It then clamps each selected coordinate of the H1 rollout state to the corresponding normalized concept target extracted from the held-out next video window, which lies in $[0, 1]$. For an activity-belief edit, the oracle uses the ground-truth activity at one of the three preceding source steps t, $t + 1$, or $t + 2$ and sets its probability to 0.9. The remaining probability mass is redistributed proportionally among the other classes before the sparse activity-to-concept message is applied. For an edge edit, the evaluator selects active learned relations by effective strength and multiplies each selected relation by -1 during the forward pass. Consequently, edge inversion reverses the additive relation message but leaves the residual state path intact. These definitions make all reported oracle results upper bounds on interface sensitivity rather than evidence that a user can identify the required edit from observed video alone.

Activity-belief interventions are selected from the three source steps that can affect H3. At the maximum feasible activity budget, oracle-ranked edits increase true-class probability by $+3.0$ pp, $+28.3$ pp, and $+21.1$ pp on BARISTA, Breakfast, and MPII, respectively. Oracle-ranked concept-node edits at budget five increase the same quantity by $+7.2$ pp, $+11.3$ pp, and $+9.8$ pp. Strongest learned-edge inversion changes true-class probability by no more than 0.02 pp. This small direct effect is consistent with the local scope of the edit: one relation message is reversed while the residual path and other active relations remain intact. Edge inversion therefore tests local relation sensitivity, while the intervention results overall establish sensitivity of the learned prediction mechanism rather than a causal effect in the video-generating process.

E.1.2 Qwen intervention prompt and visible context

Each request contains one user message with an attached four-frame JPEG storyboard ending at time t and the following template. Bracketed fields are instantiated for each case; the candidate-edit list is provided in full.

```
You are an intervention policy for a concept-based video
forecasting model.  The storyboard contains only the
final observed window at time t and never future frames.
```

```
[MODE-SPECIFIC TASK] Inspect the storyboard and model
state, then rank compatible candidate edits.  Concept
edits change the observed state or a preceding rollout
state; activity-belief edits change an autoregressive source
belief and affect only later forecast steps.  Do not select
conflicting alternatives or edits unsupported by visible
evidence.  Endpoint predictions:  [top-five H3 activities
and probabilities] Learned relational edges:  [up to 20
kind/source/target/signed- weight tuples] Candidate edits:
[full JSON list] Return JSON only:  {''ordered_action_ids'':
[...], ''rationale'':  ''...''}.
```

For Table 4, the candidate list contains concept name, edit location (observed t, forecast $t{+}1$, or forecast $t{+}2$), on/off state, and current activation. Blind correction states that the H3 ground truth is hidden. In the oracle-assisted conditions, the task additionally gives the true H3 activity and, for every compatible candidate, its frozen-model probability effect, post-edit prediction, and correction status. One-shot planning receives effects from the unedited state and returns five edits; sequential planning receives updated conditional effects and the current selected edits, and returns one next edit.

For Table 5, the task instead names the requested H3 activity and explicitly states that the future activity, future concept targets, and counterfactual effects are hidden. It lists all on/off concept edits and activity-belief edits across the three source steps. The strict response schema restricts choices to listed, currently compatible candidate edits; the rationale is limited to 400 characters.

E.1.3 QWEN INTERVENTION-POLICY DETAILS

The Qwen analysis focuses on the H3 concept-node candidate set used in Table 4. It contains 540 cases, comprising 30 initially correct and 30 initially wrong cases for each of three datasets and three seeds. The blind policy receives only observed visual evidence and model state, whereas oracle-assisted policies also receive the future target and frozen-model counterfactual effects.

Qwen provenance and policy settings. Each request uses temperature 0.4, at most 600 output tokens, and a strict JSON schema containing the ordered action IDs and rationale. The base request seed is $20260803 + 100\,i + r$, where i is the case index and r is the policy-run index; sequential requests add $100s + a$ for budget step s and retry index a. The reported evaluation uses one policy run, ten concept candidates, persistent edits, and the H3 endpoint only; malformed responses are retried under the same schema, and sequential requests add the budget step and retry index to the request seed. The server generation defaults retain top-k=20 and top-p=0.95 because the client does not override those fields. In the oracle-assisted condition, the policy additionally receives the held-out H3 target and the conditional single-action counterfactual effects; the blind condition receives neither.

For an explicit control, the same fixed candidate pools are evaluated with a deterministic random action order and with no edit at all. The random order uses seed $20260805 + 100\,i + r$, where i is the case index and r is the policy-run index. Table 28 reports the budget-five control results over the same 30 wrong and 30 correct cases per dataset and seed; the no-op row is the exact zero-edit reference at budget zero.

Table 28: **Qwen intervention controls.** Mean ± sample standard deviation over three seeds at budget five. $\Delta p(y)$ is the true-class probability change in percentage points; W→C and C→W are cumulative correction and harm rates. The no-op row applies no intervention and is therefore exactly zero.

	No-op (b=0)			Random order (b=5)		
Dataset	$\Delta p(y)$	W→C	C→W	$\Delta p(y)$	W→C	C→W
BARISTA	0.0	0.0	0.0	$-4.8_{\pm 4.4}$	$18.9_{\pm 7.7}$	$5.6_{\pm 9.6}$
Breakfast s1	0.0	0.0	0.0	$-13.7_{\pm 2.8}$	$11.1_{\pm 3.8}$	$46.7_{\pm 12.0}$
MPII Attr	0.0	0.0	0.0	$-13.0_{\pm 1.4}$	$16.7_{\pm 1.9}$	$42.2_{\pm 1.9}$

Table 29 reports the sequential oracle-assisted correction curve under the same Qwen policy, candidate pool, cases, seeds, and H3-only protocol described above.

Table 29: **Sequential oracle-assisted Qwen correction curve.** Cumulative H3 wrong-to-correct rate in percent, mean ± sample standard deviation over three seeds. Each seed contains 30 initially wrong cases.

Dataset	b=1	b=2	b=3	b=4	b=5
BARISTA	$23.3_{\pm 14.5}$	$46.7_{\pm 14.5}$	$54.4_{\pm 10.7}$	$55.6_{\pm 9.6}$	$57.8_{\pm 10.7}$
Breakfast s1	$38.9_{\pm 8.4}$	$55.6_{\pm 17.1}$	$67.8_{\pm 10.2}$	$72.2_{\pm 8.4}$	$75.6_{\pm 3.8}$
MPII Attr	$41.1_{\pm 11.7}$	$51.1_{\pm 12.6}$	$55.6_{\pm 13.5}$	$56.7_{\pm 11.5}$	$58.9_{\pm 7.7}$

The sequential correction rate increases monotonically across all three datasets. At budget five, sequential Qwen shares approximately 3.2 of its five selected actions with the greedy sequential oracle on average, compared with approximately 2.1 for one-shot Qwen. Twenty-five of 2,700 Qwen decisions use the logged conditional-best fallback after Qwen repeatedly proposes unavailable edits, so this setting is an oracle-assisted robustness analysis rather than a deployable intervention policy.