SURGCAST: DECODING POLAR TRANSITIONS FOR SURGICAL STATE-CHANGE FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Surgical state-change forecasting predicts changes in activity states within a future window from past observations. Such forecasts can support instrument preparation and intraoperative coordination. Direct categorical decoding leaves transition timing and semantic effects implicit: timing determines whether a change falls within the forecast window, while semantic effects specify which activities are added or removed. We introduce SurgCast, a structured decoder that explicitly models these complementary factors using features from a frozen visual foundation model. TransitionCast parameterizes a learned transition representation in polar coordinates, decoding its magnitude into transition occurrence probabilities over future intervals and its direction into predictions of activity additions and removals. These predictions receive separate supervision and are composed with current-state evidence over the forecast window to construct four-category state-change probabilities. The resulting probabilities are fused with base predictions, after which StateRecast refines the fused logits through an additive residual learned with the first-stage predictor fixed. SurgCast achieves state-of-the-art performance with state mAP scores of 40.33% for surgical actions on CholecT50 and 29.55% for procedural steps on GraSP, exceeding SurgFUTR-TS results by 3.93 and 7.65 percentage points, respectively. On CholecT50, SurgCast improves state mAP over the corresponding baselines by at least 1.67 percentage points across all evaluated backbone types and sizes.

1 INTRODUCTION

In the operating room, experienced surgical teams rely on foresight. Surgeons plan their next actions and prepare for upcoming procedural steps. Equipping surgical assistance systems with such anticipatory capabilities is essential for proactive intraoperative support (Maier-Hein et al., 2017).

Providing such support requires using visual evidence from an ongoing procedure to predict how surgical activities will evolve. Existing surgical forecasting methods address this task by estimating the time to future events (Rivoir et al., 2020; Yuan et al., 2022) and predicting upcoming activities (Ban et al., 2022), enabling anticipation of instrument use and procedural progression. State-change learning (Sharma et al., 2025) makes this evolution explicit: given the observed surgical video, it predicts which recently observed activities will also occur in an upcoming window, which will be absent, and which previously absent activities will emerge. Each activity is assigned one of four categories (continuity, discontinuity, onset, or background) according to its presence in the observed and future window. Under this formulation, SurgFUTR (Sharma et al., 2025) learns predictive state representations through action dynamics modeling and teacher–student supervision before decoding them directly into categorical predictions. Recent work also explores forecasting in the feature space of frozen visual foundation models (VFMs). DINO-Foresight (Karypidis et al., 2025) trains a separate predictor to forecast future VFM features in driving scenes, decoded by off-the-shelf task heads. V-JEPA 2 (Assran et al., 2025) achieves strong action anticipation on egocentric videos by training an attentive probe on a frozen video encoder and predictor. Together, these studies show that pretrained representations can support forecasting without updating the encoder, but it remains unclear how to decode them to explicitly capture changes between observed and future surgical activity states.

For state-change forecasting, direct categorical decoding leaves the roles of transition timing and semantic effect implicit. Transition timing determines whether a change falls within the forecast

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

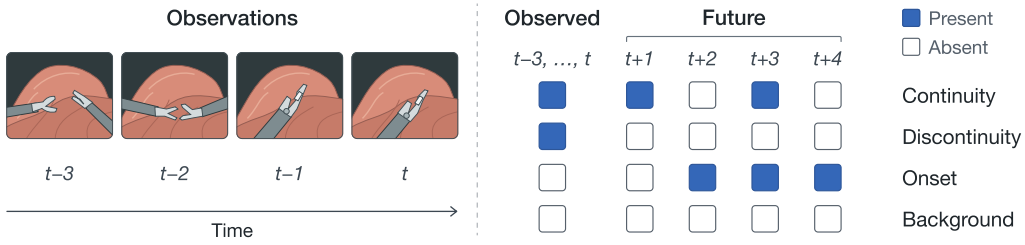

Figure 1: **Surgical state-change forecasting.** Given video up to time t, the model predicts each action or procedural step to continuity (persists), discontinuity (ends), onset (begins), or background (remains absent) in the future window. Filled and empty squares mark presence and absence.

window, while its semantic effect specifies which activities begin or end. Together with the current activity state, these two factors therefore determine each activity's state-change category. We hypothesize that explicitly parameterizing the temporal and semantic factors and composing their predictions with current-state evidence provides an effective inductive bias for this task.

We introduce SurgCast, a framework that implements this inductive bias as a structured decoder on frozen VFM representations, learning only the components built on the pretrained features. Its first module, TransitionCast, decouples transition timing from semantic effects through a polar transition parameterization. The two factors can differ in predictability: once the cystic duct has been clipped, cutting typically follows, although the time at which it begins varies. TransitionCast therefore represents the upcoming transition as a learned vector whose magnitude and direction assume temporal and semantic predictive roles, respectively. The magnitude acts as the total transition intensity over the forecast window and is decoded into transition occurrence probabilities over future time intervals. The direction carries no scale information and predicts which activities begin or end. These two predictions receive separate supervision.

Within TransitionCast, the State Composer integrates these predictions over the forecast window and combines them with current-state evidence to construct probabilities over the four state-change categories, which are then fused with the predictions of a prediction head. In a second training stage, StateRecast refines the resulting four-state logits through a zero-initialized residual while the first-stage predictor is frozen. Our contributions are threefold:

- **Decoupling when from what in polar coordinates.** We parameterize a learned transition representation in polar coordinates within TransitionCast, using its magnitude to predict when a change occurs within the forecast window and its direction to predict what changes.
- **Decoding state changes from frozen representations.** We decode state changes from frozen VFM representations, learning only a lightweight predictor on the pretrained features. SurgCast adds approximately a million trainable parameters over direct decoding.
- **Demonstrating gains across datasets and backbones.** We achieve state-of-the-art State mAP of 40.33% for surgical actions on CholecT50 and 29.55% for procedural steps on GraSP, exceeding SurgFUTR-TS by 3.93 and 7.65 percentage points. On CholecT50, SurgCast improves over each backbone's baseline by at least 1.67 percentage points.

2 Task Formulation

Given surgical video observed up to the current time, the task is to anticipate which activities will occur in the next few seconds (Figure 1). For an action such as clipping, the prediction distinguishes four outcomes. *Continuity* means that clipping occurs in both the recent observed window and the future window. *Discontinuity* means that it occurs only in the observed window. *Onset* means that it occurs only in the future window. *Background* means that it is absent from both windows, even if other activities occur. The model outputs a probability distribution over these four states for each activity, which may be a surgical action or procedural step. These states describe window-level occurrence rather than uninterrupted execution or the first or final occurrence in a procedure.

3 METHOD

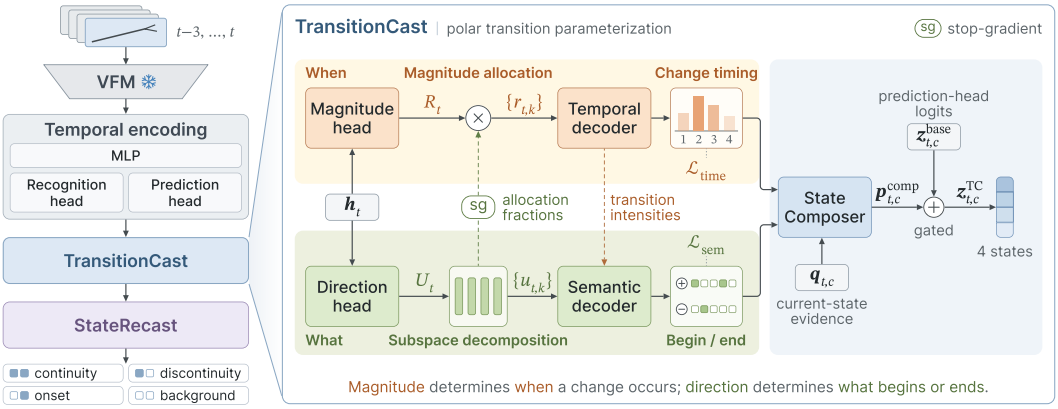

Figure 2: **SurgCast architecture. Left:** A frozen VFM feeds temporal encoding and prediction heads, followed by TransitionCast and StateRecast. **Right:** TransitionCast decouples *when* a state changes (magnitude R_t) from *what* begins or ends (direction $\boldsymbol{U}_t$). The State Composer combines both with current-state evidence into four-state probabilities that correct the base logits.

3.1 TEMPORAL ENCODING

We extract frame features $\boldsymbol{x}_t \in \mathbb{R}^D$ using a frozen VFM. To capture temporal context, we concatenate the current frame features $\boldsymbol{x}_t$, their difference from the preceding frame $\boldsymbol{x}_t - \boldsymbol{x}_{t-1}$, and the mean of recent frame features $\bar{\boldsymbol{x}}_t$, yielding $\boldsymbol{x}_t^{\text{temp}} = [\boldsymbol{x}_t \,\|\, \boldsymbol{x}_t - \boldsymbol{x}_{t-1} \,\|\, \bar{\boldsymbol{x}}_t] \in \mathbb{R}^{3D}$, where $\|$ denotes concatenation. A shared MLP maps $\boldsymbol{x}_t^{\text{temp}}$ to a temporal observation representation $\boldsymbol{h}_t \in \mathbb{R}^d$. The recognition and prediction heads share $\boldsymbol{h}_t$ and are jointly trained. For each activity c, the recognition head predicts its current-state probability $p_{t,c}^{\text{rec}}$, while the prediction head outputs logits $\boldsymbol{z}_{t,c}^{\text{base}} \in \mathbb{R}^4$ over the four state-change categories. From this shared representation, TransitionCast constructs a latent state-change representation $\boldsymbol{v}_t \in \mathbb{R}^d$ through the polar parameterization described below.

3.2 TRANSITIONCAST: POLAR TRANSITION PARAMETERIZATION

Polar parameterization. TransitionCast parameterizes the latent transition vector $\boldsymbol{v}_t$ in polar form, separating its magnitude R_t from its direction $\boldsymbol{U}_t$:

$$\boldsymbol{v}_t = R_t \boldsymbol{U}_t, \qquad R_t > 0, \qquad \|\boldsymbol{U}_t\|_2 = 1.$$

To construct this representation, we predict its magnitude and direction using two separate, jointly trained heads applied to the shared temporal representation $\boldsymbol{h}_t$:

$$R_t = \epsilon + \text{softplus}(f_r(\boldsymbol{h}_t)), \qquad \boldsymbol{U}_t = \frac{f_u(\boldsymbol{h}_t)}{\|f_u(\boldsymbol{h}_t)\|_2}, \tag{1}$$

where f_r and f_u denote the magnitude and direction heads, respectively, and $\epsilon > 0$ sets a positive lower bound on R_t for numerical stability. The magnitude R_t determines the probability of a change within the forecast window, while the direction $\boldsymbol{U}_t$ encodes which activities may begin or end. To capture distinct transition patterns, we partition the transition representation space into K orthogonal subspaces, indexed by $k \in \{1, \ldots, K\}$. For each subspace k, we project $\boldsymbol{U}_t$ onto that subspace and normalize the resulting component to obtain its direction $\boldsymbol{u}_{t,k}$. We allocate the total magnitude R_t across these subspaces using subspace weights, yielding magnitudes $r_{t,k}$ that satisfy $\sum_{k=1}^{K} r_{t,k} = R_t$. We apply stop-gradient to these weights when computing $r_{t,k}$, preventing gradients through the subspace magnitudes from updating the direction head. The subspace magnitudes and directions are then passed to the temporal and semantic decoders, respectively.

Magnitude and temporal decoding. To predict when a change occurs within the forecast window, we convert the magnitudes into transition intensities over time. The temporal decoder distributes each magnitude $r_{t,k}$ over the L seconds after the last observed time t, indexed by $\ell \in \{1, \dots, L\}$. For each subspace k, we distribute its magnitude $r_{t,k}$ across these seconds. We normalize the learned temporal weights $w_{k,\ell}$ with a softmax over time, so that $\sum_{\ell=1}^{L} w_{k,\ell} = 1$. Multiplying these weights by $r_{t,k}$ gives the transition intensity assigned to each second: $\lambda_{t,k,\ell} = r_{t,k} w_{k,\ell}$. Because the weights sum to one, this allocation preserves the total magnitude $r_{t,k}$. Summing over subspaces gives the total transition intensity integrated over future second ℓ:

$$\lambda_{t,\ell} = \sum_{k=1}^{K} \lambda_{t,k,\ell}. \tag{2}$$

To obtain the distribution of the first change, we first compute the probability that no activity-set change has occurred since time t. Let $S_{t,\ell}$ denote this probability at the end of future second ℓ. We convert the accumulated transition intensity into this probability using

$$S_{t,\ell} = \exp\left(- \sum_{j=1}^{\ell} \lambda_{t,j} \right), \qquad S_{t,0} = 1. \tag{3}$$

As transition intensity accumulates, the probability of no change decreases. The difference $S_{t,\ell-1} - S_{t,\ell}$ is the probability that no change occurs before second ℓ, but the first change occurs during that second.

Over the full forecast window, the total transition intensity satisfies $\sum_{\ell=1}^{L} \lambda_{t,\ell} = R_t$. Substituting this into Eq. 3 gives $S_{t,L} = \exp(-R_t)$, the probability of no change throughout the window. Its complement is the probability of at least one change:

$$1 - S_{t,L} = 1 - \exp(-R_t).$$

Thus, the total magnitude R_t directly determines the probability of at least one change within the forecast window.

Direction and semantic decoding. To characterize the activities involved in a transition, the semantic decoder maps each subspace direction $\boldsymbol{u}_{t,k}$ to probabilities $m^{+}_{t,c,k,\ell}$ and $m^{-}_{t,c,k,\ell}$. These represent the probabilities that activity c begins or ends, respectively, given a transition during future second ℓ. We apply a sigmoid to each output, allowing multiple activities to change at the same transition.

At each future second ℓ, we combine the subspace predictions using their transition intensities as weights:

$$m^{\pm}_{t,c,\ell} = \frac{\sum_{k=1}^{K} \lambda_{t,k,\ell}\, m^{\pm}_{t,c,k,\ell}}{\lambda_{t,\ell} + \epsilon}, \tag{4}$$

This aggregation gives greater weight to subspaces with higher transition intensity at each future second. The resulting probabilities $m^{\pm}_{t,c,\ell}$ provide the semantic input to the State Composer.

State Composer. The State Composer combines decoded transition information with probabilities from the recognition head to construct a four-state distribution for each activity c. This distribution is then used to adjust the logits produced by the prediction head.

To represent activity presence at the current time and within the observation window, we use the current recognition probability $p^{\mathrm{rec}}_{t,c}$ together with the maximum recognition probability $p^{\mathrm{obs}}_{t,c} = \max_{\tau \in \mathcal{O}_t} p^{\mathrm{rec}}_{\tau,c}$, where $\mathcal{O}_t$ denotes the observation window ending at t. These two probabilities and their difference form the recognition feature vector $\boldsymbol{q}_{t,c} = [p^{\mathrm{rec}}_{t,c}, p^{\mathrm{obs}}_{t,c}, p^{\mathrm{obs}}_{t,c} - p^{\mathrm{rec}}_{t,c}]^{\top}$.

For the future window, we use the transition intensities $\lambda_{t,\ell}$ and semantic probabilities $m^{\pm}_{t,c,\ell}$ to model transitions between activity presence and absence. From these transitions, we compute the probability that activity c is absent at every future sampling time, conditioned separately on its presence and absence at t. Combining these conditional probabilities with the calibrated current and

observation-window recognition probabilities yields the four-state distribution $\boldsymbol{p}_{t,c}^{\mathrm{comp}}$. Appendix C details the calibration and probability construction.

We then use the composed distribution to apply a gated correction to the base forecasting logits:

$$\boldsymbol{z}_{t,c}^{\mathrm{TC}} = \boldsymbol{z}_{t,c}^{\mathrm{base}} + \boldsymbol{\alpha}_{t,c} \odot \left[\log \boldsymbol{p}_{t,c}^{\mathrm{comp}} - \mathrm{logsoftmax}\left(\boldsymbol{z}_{t,c}^{\mathrm{base}}\right)\right], \tag{5}$$

where $\odot$ denotes element-wise multiplication and $\boldsymbol{\alpha}_{t,c} \in (-1, 1)^4$ controls the sign and magnitude of the correction for each state. A gating MLP with a $\tanh$ output predicts $\boldsymbol{\alpha}_{t,c}$ from the calibrated recognition feature vector, the composed and base log-probability vectors, and a learned activity embedding. The calibrated recognition feature vector preserves the structure of $\boldsymbol{q}_{t,c}$: its first two entries are the calibrated recognition probabilities, and its third entry is their recomputed difference.

3.3 StateRecast

To further adapt to the downstream forecasting task, StateRecast combines temporal, recognition, and transition information to learn an additive logit correction. For each activity c at time t, its inputs include the shared temporal representation $\boldsymbol{h}_t$, transition direction $\boldsymbol{U}_t$, fused logits $\boldsymbol{z}_{t,c}^{\mathrm{TC}}$, and recognition feature vector $\boldsymbol{q}_{t,c}$ defined in the State Composer. We also group the log total intensity $\log R_t$ and cumulative change probabilities $1 - S_{t,\ell}$ at selected forecast horizons into a magnitude and timing feature vector.

To construct the residual input, we first apply LayerNorm to $\boldsymbol{h}_t$ and $\boldsymbol{U}_t$. We then map these two representations, the magnitude and timing feature vector, the fused logits, and the recognition feature vector through separate linear layers. The resulting features are concatenated with a learned activity embedding to form $\boldsymbol{f}_{t,c}$. A residual MLP $\mathcal{R}_\theta$ maps this vector to a four-dimensional logit correction:

$$\boldsymbol{z}_{t,c}^{\mathrm{SR}} = \boldsymbol{z}_{t,c}^{\mathrm{TC}} + \mathcal{R}_\theta(\boldsymbol{f}_{t,c}). \tag{6}$$

Both the TransitionCast gating MLP and the StateRecast residual MLP have zero-initialized output layers, so each module initially preserves its input logits. At inference, applying softmax to $\boldsymbol{z}_{t,c}^{\mathrm{SR}}$ yields the refined four-state probabilities.

3.4 Training objectives

We train the model in two stages. In the first stage, we train TransitionCast and all other modules except the VFM and StateRecast. The objective combines four weighted losses for current state recognition, future four-state prediction, and temporal and semantic modeling through the magnitude and direction defined by the polar parameterization, respectively. In the second stage, we freeze the first-stage model and train only StateRecast to refine the four-state predictions through residual corrections.

The recognition loss supervises whether each activity is currently present. The forecasting loss applies four-state cross-entropy to TransitionCast's predictions, including the background state. We describe the objectives for the polar parameterization and residual refinement below.

Temporal supervision of magnitude. We supervise the temporal decoder using the time to the first change in the activity set or the observed interval without a change. Let ℓ^* denote the time bin containing the first observed change after time t. If no change is observed, it denotes the last fully observed bin. Using the survival probabilities $S_{t,\ell}$ defined above, the loss for a single sample is

$$\ell_{\mathrm{time}} = \begin{cases} -\log(S_{t,\ell^*-1} - S_{t,\ell^*}), & \text{change observed,} \\ -\log S_{t,\ell^*}, & \text{no change observed.} \end{cases} \tag{7}$$

For an observed change, this objective increases the probability assigned to its true time bin. Otherwise, it increases the probability of no change throughout the observed interval, without making assumptions about subsequent changes. We average this loss over valid training samples to obtain $\mathcal{L}_{\mathrm{time}}$.

Semantic supervision of direction. To encourage the subspace directions to encode the semantic effects of a transition, we supervise their decoded probabilities using activity change labels that

Table 1: **Surgical action forecasting on CholecT50.** We report AP (%) per foreground state and mAP as their average. Results are mean ± standard deviation over three seeds, with all except SurgCast and the DINOv3 baseline from SurgFUTR (Sharma et al., 2025). Best results are in **bold**.

Method	Continuity	Discontinuity	Onset	mAP
Representation baselines				
VideoMAEv2-D (Wang et al., 2023)	63.1±0.2	7.8±0.3	8.9±0.3	26.6±0.2
VideoMAEv2-K (Wang et al., 2023)	75.2±0.8	13.5±0.4	14.7±0.2	34.4±0.1
MoCo v2 (Chen et al., 2020)	74.1±1.5	11.9±0.7	13.7±0.2	33.2±0.3
EndoViT (Batić et al., 2024)	59.7±1.8	5.7±0.3	7.0±0.1	24.1±0.6
SurgeNetXL (Jaspers et al., 2026)	74.9±0.8	11.2±1.1	12.8±0.2	33.0±0.2
DINO (Caron et al., 2021)	74.8±0.6	10.9±0.7	13.3±0.5	33.0±0.1
DINOv3 (Siméoni et al., 2026)	79.7±0.4	17.8±0.8	16.4±0.4	38.0±0.5
State-forecasting methods				
SurgFUTR-Lite (Sharma et al., 2025)	76.1±1.0	14.1±0.7	16.5±0.8	35.6±0.7
SurgFUTR-S (Sharma et al., 2025)	76.1±1.0	14.3±0.6	14.9±1.2	35.1±0.6
SurgFUTR-TS (Sharma et al., 2025)	77.8±1.0	14.9±0.8	16.6±0.2	36.4±0.5
SurgCast (Ours)	**81.00**±0.30	**20.80**±0.14	**19.19**±0.33	**40.33**±0.19

indicate which activities begin or end at the first observed change. For this supervision, we average these probabilities uniformly across subspaces, without weighting them by transition intensity:

$$\bar{m}^{\pm}_{t,c,\ell} = \frac{1}{K}\sum_{k=1}^{K} m^{\pm}_{t,c,k,\ell}. \tag{8}$$

The ground-truth activity labels at the last observed time t determine the type of supervision. For samples with an observed change, we use $\bar{m}^{+}_{t,c,\ell^*}$ for activities absent at t and $\bar{m}^{-}_{t,c,\ell^*}$ for those present, where ℓ^* is the bin containing the first observed change. Let p denote the corresponding probability. We set $y = 1$ if activity c undergoes the corresponding change at that transition, and $y = 0$ otherwise. Each prediction is supervised with weighted binary cross-entropy:

$$\ell_{\text{sem}}(p, y) = -\omega_1 y \log p - \omega_0 (1 - y) \log(1 - p). \tag{9}$$

The weights ω_1 and ω_0 balance positive and negative examples for each activity and change type. We sum the weighted contributions separately for each activity and change type, then average equally across all activities and both change types to obtain $\mathcal{L}_{\text{sem}}$.

Residual refinement objective. After the first stage, we freeze the trained model and optimize only StateRecast. Its objective combines a four-state classification loss, a prediction consistency loss, and a residual penalty:

$$\mathcal{L}_{\text{SR}} = \mathcal{L}_{\text{cls}} + \beta_1 \mathcal{L}_{\text{cons}} + \beta_2 \mathcal{L}_{\text{res}}, \tag{10}$$

where β_1 and β_2 weight the consistency loss and residual penalty, respectively.

The classification loss $\mathcal{L}_{\text{cls}}$ carries over the four-state classification objective from the first stage, applying class-weighted cross-entropy to the predictions refined by StateRecast.

The consistency loss $\mathcal{L}_{\text{cons}}$ averages $D_{\text{KL}}\big(\boldsymbol{p}^{\text{TC}}_{t,c} \,\|\, \boldsymbol{p}^{\text{SR}}_{t,c}\big)$ over valid clip–activity pairs whose ground-truth state is continuity or background, discouraging large deviations from the first-stage predictions. Here, $\boldsymbol{p}^{\text{TC}}_{t,c}$ and $\boldsymbol{p}^{\text{SR}}_{t,c}$ are obtained by applying softmax to the logits before and after refinement, respectively.

The residual penalty $\mathcal{L}_{\text{res}}$ controls how much StateRecast adjusts the first-stage logits. Since the residual is added directly to these logits, a zero residual preserves the original predictions, while larger residuals produce larger logit adjustments. We average $\|\mathcal{R}_\theta(\boldsymbol{f}_{t,c})\|_2^2$ over valid samples and activities, penalizing larger corrections more strongly. Together with the classification loss, this penalty encourages improvements in prediction while discouraging unnecessarily large adjustments.

4 EXPERIMENT

4.1 EXPERIMENTAL SETUP

Datasets and evaluation protocol. We evaluate SurgCast on CholecT50 (Nwoye et al., 2022) and GraSP (Ayobi et al., 2025), covering different surgical procedures, platforms, and annotation granularities. CholecT50 comprises 50 laparoscopic cholecystectomy videos, and we forecast state changes for its 10 action-verb categories. GraSP comprises 13 robot-assisted radical prostatectomy videos, and we forecast state changes for its 21 procedural step categories. We follow the video splits provided in the official split and adopt the clip-sampling protocol implemented in the officially released SurgFUTR code, sample frames at 1 fps and construct adjacent, non-overlapping observation and future windows, each containing four frames. Relative to the last observed time t, the sampling offsets are $\{-3, -2, -1, 0\}$ and $\{1, 2, 3, 4\}$ seconds, respectively. We compute average precision (AP) for each state over all test clip–activity pairs and report mean average precision (mAP) as the mean AP across continuity, discontinuity, and onset. Background pairs serve as negatives for these three states, while background AP is excluded from the mean. For SurgCast, we report the mean and standard deviation over three random seeds.

Compared methods. We compare against two groups of external methods: representation baselines based on VideoMAEv2(Wang et al., 2023), MoCo v2(Chen et al., 2020), EndoViT(Batić et al., 2024), and SurgeNetXL(Jaspers et al., 2026), and three SurgFUTR(Sharma et al., 2025) variants for state-change forecasting (Lite, S, and TS). Results for these methods are taken from SurgFUTR (Sharma et al., 2025). We also establish an MLP baseline that directly predicts the four state categories from frozen DINOv3 features(Siméoni et al., 2026). SurgCast builds on this baseline by introducing TransitionCast and StateRecast, allowing us to assess the gains from structured transition modeling and residual refinement.

Implementation details. Our main experiments use a frozen DINOv3 ViT-L/16 backbone. On CholecT50, we additionally evaluate DINOv3 ViT-S/16 and ViT-B/16, as well as frozen features from the VideoMAEv2 ViT-S/16 student backbone of our locally reproduced SurgFUTR-TS, to assess SurgCast across backbone scales and representation sources. For CholecT50, we fuse CLS features with mean patch features to incorporate local visual cues for action-level forecasting, whereas GraSP uses only precomputed CLS features to represent global scene context for coarser procedural-step forecasting. Following the two-stage training scheme in Section 3.4, we first keep the VFM frozen and jointly train TransitionCast and the shared prediction modules for 24 epochs. We then freeze the first-stage model and train only StateRecast for 6 epochs. Both stages use AdamW(Loshchilov & Hutter, 2019), with base learning rates of 10^{-4} and 10^{-3} and batch sizes of 64 and 256, respectively. All task-specific parameters are fitted on the training split, and forecasting and refinement checkpoints are selected using validation mAP.

4.2 COMPARISON WITH STATE-OF-THE-ART METHODS

Surgical action forecasting on CholecT50. SurgCast achieves the best mAP among the compared methods at $40.33 \pm 0.19\%$, exceeding SurgFUTR-TS by 3.93 percentage points (Table 1). Our DINOv3-L baseline already achieves 37.97% mAP, surpassing the 36.4% reported for SurgFUTR-TS, and SurgCast further improves upon this strong baseline by 2.36 percentage points, reaching 40.33% mAP. Relative to SurgFUTR-TS, the improvement spans all three foreground states, with gains of 5.90 and 2.59 percentage points in discontinuity and onset AP.

Surgical step forecasting on GraSP. Following the same evaluation setup, we evaluate SurgCast on GraSP using a strong DINOv3 baseline, which achieves 25.0% mAP and exceeds SurgFUTR-TS by 3.10 percentage points. SurgCast yields a further gain of 4.55 percentage points, achieving the best mAP of $29.55 \pm 0.47\%$ among the compared methods (Table 2). It achieves the highest discontinuity and onset AP while maintaining continuity AP of 46.61%, comparable to the baseline's 46.8%. These results extend the advantage of SurgCast in forecasting activity changes from fine-grained surgical actions to procedural steps in robot-assisted surgery.

Table 2: **Surgical step forecasting on GraSP.** Except for SurgCast and the DINOv3 baseline, all results are from SurgFUTR(Sharma et al., 2025).

Method	Continuity	Discontinuity	Onset	mAP
Representation baselines				
VideoMAEv2-D (Wang et al., 2023)	19.9±0.9	5.0±0.2	5.2±0.1	10.1±0.3
VideoMAEv2-K (Wang et al., 2023)	36.8±0.3	10.6±0.5	12.6±0.6	20.0±0.2
MoCo v2 (Chen et al., 2020)	34.1±1.0	13.0±0.9	13.0±0.4	20.0±0.0
DINOv3 (Siméoni et al., 2026)	**46.8**±0.4	11.7±0.6	16.4±1.0	25.0±0.5
EndoViT (Batić et al., 2024)	19.7±1.8	6.1±0.6	6.2±0.8	10.7±1.1
SurgeNetXL (Jaspers et al., 2026)	20.0±11.8	6.7±2.1	7.1±2.4	11.3±5.3
State-forecasting methods				
SurgFUTR-Lite (Sharma et al., 2025)	37.9±1.2	12.3±1.2	14.9±1.1	21.7±1.1
SurgFUTR-S (Sharma et al., 2025)	36.8±2.5	10.1±0.9	12.4±0.9	19.8±0.5
SurgFUTR-TS (Sharma et al., 2025)	38.8±1.8	12.7±1.3	14.2±0.8	21.9±0.5
SurgCast (Ours)	46.61±1.27	**22.00**±0.55	**20.04**±0.75	**29.55**±0.47

Table 3: **SurgCast across backbones on CholecT50.** Shaded rows add SurgCast to the baseline on the same frozen backbone, with Δ as the mAP gain. SurgFUTR-TS uses our reproduced VideoMAEv2 ViT-S/16 students, matched by seed, with their native predictions as the baseline.

Backbone	Method	Continuity	Discontinuity	Onset	mAP	Δ
DINOv3 ViT-S/16	Baseline	79.99±0.03	14.34±0.33	15.27±0.40	36.53±0.13	—
	+ SurgCast	79.88±0.42	17.57±0.52	17.14±0.56	38.20±0.09	**+1.67**
DINOv3 ViT-B/16	Baseline	80.45±0.13	16.41±0.08	15.18±0.31	37.35±0.04	—
	+ SurgCast	81.26±0.09	19.92±0.51	17.51±0.13	39.56±0.13	**+2.21**
DINOv3 ViT-L/16	Baseline	79.72±0.43	17.79±0.81	16.41±0.36	37.97±0.46	—
	+ SurgCast	81.00±0.30	20.80±0.14	19.19±0.33	40.33±0.19	**+2.36**
SurgFUTR-TS student[1]	Baseline	77.88±0.26	14.38±0.42	14.96±0.66	35.74±0.17	—
	+ SurgCast	81.17±0.15	19.20±0.29	16.94±1.14	39.10±0.42	**+3.36**

4.3 Evaluation across Backbone Types and Sizes

Beyond the main comparisons, we examine whether SurgCast consistently improves forecasting across backbone scales and representation sources on CholecT50 (Table 3). For each DINOv3 backbone, we use an MLP baseline that directly predicts the four state-change categories from frozen features, without TransitionCast or StateRecast. Across DINOv3 ViT-S/16, ViT-B/16, and ViT-L/16, SurgCast improves mAP over the corresponding MLP baselines by 1.67, 2.21, and 2.36 percentage points, respectively, with gains in both discontinuity and onset AP at every scale.

To extend this evaluation beyond DINOv3, we apply SurgCast to frozen VideoMAEv2 ViT-S/16 features from our locally reproduced SurgFUTR-TS students, without patch-feature fusion. Each run uses the student trained with the same random seed. Using the native prediction pipeline of SurgFUTR-TS as the baseline, SurgCast improves mAP from 35.74% to 39.10%, a gain of 3.36 percentage points.

These results show that SurgCast's benefits extend across backbone scales and representation sources, including task-trained surgical features. We next examine the contributions of its individual components and the design choices underlying these improvements.

[1]We reproduced SurgFUTR-TS using 32 of the 35 official training videos, excluding VID96, VID103, and VID110 because their raw videos were unavailable to us. The official validation and test splits were unchanged.

Table 4: **Component ablation on CholecT50.** TransitionCast (TC) and StateRecast (SR) are added individually and jointly to the DINOv3-L baseline, with Δ as the mAP gain over the baseline. Best results are in **bold**.

Method	TC	SR	Continuity	Discontinuity	Onset	mAP	Δ
Baseline	✗	✗	79.72±0.43	17.79±0.81	16.41±0.36	37.97±0.46	—
+ TC	✓	✗	80.95±0.05	19.07±0.51	17.24±0.75	39.09±0.40	+1.12
+ SR	✗	✓	80.01±0.60	19.25±1.04	17.31±0.25	38.86±0.47	+0.89
+ Both (SurgCast)	✓	✓	**81.00**±0.30	**20.80**±0.14	**19.19**±0.33	**40.33**±0.19	**+2.36**

Table 5: **Temporal and semantic factors in TransitionCast on CholecT50.** Dir. and Mag. train the semantic (direction) and temporal (magnitude) decoders, respectively. All variants exclude StateRecast, and Δ is the mAP change from the DINOv3-L baseline. Best results are in **bold**.

	TransitionCast		AP / mAP (%)				
Method	Dir.	Mag.	Continuity	Discontinuity	Onset	mAP	Δ
Baseline	✗	✗	79.72±0.43	17.79±0.81	16.41±0.36	37.97±0.46	—
+ Direction	✓	✗	79.83±0.17	16.15±0.40	15.65±1.02	37.21±0.33	−0.76
+ Magnitude	✗	✓	80.74±0.35	**19.09**±0.36	16.69±0.76	38.84±0.46	+0.87
+ Dir. & Mag.	✓	✓	**80.95**±0.05	19.07±0.51	**17.24**±0.75	**39.09**±0.40	+1.12

4.4 Ablation and Design Analysis

Complementary contributions of TransitionCast and StateRecast. TransitionCast improves the initial forecast, while StateRecast further refines predictions from the fixed predictor (Table 4). Individually, TC and SR improve mAP over the baseline by 1.12 and 0.89 percentage points, respectively; together, they achieve 40.33% mAP, a 2.36-point gain. Adding SR to TC primarily improves discontinuity and onset AP (+1.73 and +1.95 points), with little change in continuity (+0.05 points), indicating that its additional benefit lies mainly in forecasting state changes.

Temporal and semantic factors within TransitionCast. To examine the source of TC's gains, we evaluate temporal modeling through magnitude and semantic modeling through direction, both individually and jointly, before StateRecast (Table 5). Learning magnitude alone improves mAP by 0.87 points, whereas learning direction alone reduces it by 0.76 points. Joint learning achieves the highest mAP of 39.09%, exceeding magnitude alone by 0.25 points. This pattern suggests that semantic predictions become more useful when the State Composer combines them with temporal evidence of whether a change occurs within the forecast window, supporting the separate parameterization and joint composition of the two factors.

5 Conclusion

We presented SurgCast, a structured framework for surgical state-change forecasting. TransitionCast models transition timing and semantic effects through the magnitude and direction of a learned polar representation, and composes these predictions with current-state evidence. StateRecast further refines the forecasts through additive logit corrections. Experiments on CholecT50 and GraSP demonstrate state-of-the-art performance for surgical actions and procedural steps, with consistent gains across the evaluated backbones on CholecT50. These findings support structured transition decoding as an effective approach to surgical forecasting with frozen visual representations.

AI use statement

We used large language models to assist with manuscript editing and language polishing, the identification of relevant literature, and research implementation through code generation. The authors take full responsibility for the final manuscript, implementation, and reported findings, including all content produced with AI assistance.

Reproducibility statement

The training and evaluation code and configuration files will be made publicly available upon publication.

References

Mahmoud Assran, Adrien Bardes, David Fan, Quentin Garrido, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Nicolás Ayobi, Santiago Rodríguez, Alejandra Pérez, Isabela Hernández, Nicolás Aparicio, Eugénie Dessevres, Sebastián Peña, Jessica Santander, Juan Ignacio Caicedo, Nicolás Fernández, and Pablo Arbeláez. Pixel-wise recognition for holistic surgical scene understanding. *Medical Image Analysis*, 2025.

Yutong Ban, Guy Rosman, Jennifer A. Eckhoff, Thomas M. Ward, Daniel A. Hashimoto, Taisei Kondo, Hidekazu Iwaki, Ozanan R. Meireles, and Daniela Rus. SUPR-GAN: SUrgical PRediction GAN for event anticipation in laparoscopic and robotic surgery. *IEEE Robotics and Automation Letters*, 7(2):5741–5748, 2022.

Dominik Batić, Felix Holm, Ege Özsoy, Tobias Czempiel, and Nassir Navab. EndoViT: Pretraining vision transformers on a large collection of endoscopic images. *International Journal of Computer Assisted Radiology and Surgery*, 19(6):1085–1091, 2024. doi: 10.1007/s11548-024-03091-5.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9650–9660, 2021. doi: 10.1109/ICCV48922.2021.00951.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. doi: 10.48550/arXiv.2003.04297.

Tobias Czempiel, Magdalini Paschali, Matthias Keicher, Walter Simson, Hubertus Feussner, Seong Tae Kim, and Nassir Navab. TeCNO: Surgical phase recognition with multi-stage temporal convolutional networks. In *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pp. 343–352, 2020.

Daryl J. Daley and David Vere-Jones. *An Introduction to the Theory of Point Processes, Volume I: Elementary Theory and Methods*. Springer, 2 edition, 2003.

Xiaojie Gao, Yueming Jin, Yonghao Long, Qi Dou, and Pheng-Ann Heng. Trans-SVNet: Accurate phase recognition from surgical videos via hybrid embedding aggregation transformer. In *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pp. 593–603, 2021.

Tim J. M. Jaspers, Ronald L. P. D. de Jong, Yiping Li, Carolus H. J. Kusters, Franciscus H. A. Bakker, Romy C. van Jaarsveld, Gino M. Kuiper, Richard van Hillegersberg, Jelle P. Ruurda, Willem M. Brinkman, Josien P. W. Pluim, Peter H. N. de With, Marcel Breeuwer, Yasmina Al Khalil, and Fons van der Sommen. Scaling up self-supervised learning for improved surgical foundation models. *Medical Image Analysis*, 108:103873, February 2026. doi: 10.1016/j.media.2025.103873.

Efstathios Karypidis, Ioannis Kakogeorgiou, Spyridon Gidaris, and Nikos Komodakis. Dinoforesight: Looking into the future with dino. *Advances in Neural Information Processing Systems*, 38:163779–163811, 2025.

Weiyang Liu, Zhen Liu, Zhiding Yu, Bo Dai, Rongmei Lin, Yisen Wang, James M. Rehg, and Le Song. Decoupled networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2771–2779, 2018.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Lena Maier-Hein, Swaroop S. Vedula, Stefanie Speidel, Nassir Navab, Ron Kikinis, Adrian Park, Matthias Eisenmann, Hubertus Feussner, Germain Forestier, Stamatia Giannarou, Makoto Hashizume, Darko Katic, Hannes Kenngott, Michael Kranzfelder, Anand Malpani, Keno März, Thomas Neumuth, Nicolas Padoy, Carla Pugh, Nicolai Schoch, Danail Stoyanov, Russell Taylor, Martin Wagner, Gregory D. Hager, and Pierre Jannin. Surgical data science for next-generation interventions. *Nature Biomedical Engineering*, 1(9):691–696, 2017. doi: 10.1038/s41551-017-0132-7.

Chinedu Innocent Nwoye, Tong Yu, Cristians Gonzalez, Barbara Seeliger, Pietro Mascagni, Didier Mutter, Jacques Marescaux, and Nicolas Padoy. Rendezvous: Attention mechanisms for the recognition of surgical action triplets in endoscopic videos. *Medical Image Analysis*, 78:102433, 2022.

Dominik Rivoir, Sebastian Bodenstedt, Isabel Funke, Felix von Bechtolsheim, Marius Distler, Jürgen Weitz, and Stefanie Speidel. Rethinking anticipation tasks: Uncertainty-aware anticipation of sparse surgical instrument usage for context-aware assistance. In *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pp. 752–762, 2020.

Saurav Sharma, Chinedu Innocent Nwoye, Didier Mutter, and Nicolas Padoy. State-change learning for prediction of future events in endoscopic videos. *arXiv preprint arXiv:2510.12904*, 2025.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=2NlGyqNjns. Featured Certification.

Yujee Song, Donghyun Lee, Rui Meng, and Won Hwa Kim. Decoupled marked temporal point process using neural ordinary differential equations. In *International Conference on Learning Representations (ICLR)*, 2024.

Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. VideoMAE V2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14549–14560, 2023.

Lianhao Yin, Yutong Ban, Jennifer Eckhoff, Ozanan Meireles, Daniela Rus, and Guy Rosman. Hypergraph-transformer (HGT) for interaction event prediction in laparoscopic and robotic surgery. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6846–6853, 2025. doi: 10.1109/ICRA55743.2025.11127325.

Kun Yuan, Matthew Holden, Shijian Gao, and Won-Sook Lee. Anticipation for surgical workflow through instrument interaction and recognized signals. *Medical Image Analysis*, 82:102611, 2022.

A Related Work

Surgical action and workflow recognition. Surgical video recognition spans fine-grained actions and broader workflow stages. For phase recognition, TeCNO models temporal dependencies with multi-stage temporal convolutions (Czempiel et al., 2020), and Trans-SVNet aggregates spatial and temporal embeddings with a Transformer (Gao et al., 2021). At the action level, Rendezvous recognizes instrument–verb–target triplets using instrument-guided spatial attention and semantic attention across triplet components (Nwoye et al., 2022). These methods infer activity labels for observed video. In SurgCast, current-state recognition serves as evidence for forecasting how activity presence changes between the observed and future windows.

Surgical action and workflow forecasting. Existing forecasting methods address complementary targets. Rivoir et al. (2020) formulate sparse instrument anticipation as uncertainty-aware regression of the remaining time until use. IIA-Net combines instrument interactions with recognition signals to anticipate the timing of instrument and phase occurrences (Yuan et al., 2022). SUPR-GAN predicts future surgical phase trajectories through adversarial learning (Ban et al., 2022), and HGT uses a hypergraph-transformer to anticipate action triplets and safety-related events (Yin et al., 2025). Most closely related to our setting, SurgFUTR introduces state-change learning, which assigns each activity to continuity, discontinuity, onset, or background according to its presence in the observed and future windows (Sharma et al., 2025). Its teacher–student variant, SurgFUTR-TS, learns predictive representations through action dynamics modeling and distillation, then decodes them into categorical state-change predictions. SurgCast adopts this formulation and complements direct categorical decoding with a structured decoder built on frozen visual representations, explicitly linking transition timing and semantic effects to the predicted state changes.

Structured parameterizations. Explicit parameterizations can encode inductive biases about the distinct factors underlying a prediction task. The Decoupled Networks framework reparameterizes convolutional inner products into magnitude and angular functions to model intra-class variation and semantic differences separately (Liu et al., 2018). In marked temporal point processes, the conditional intensity can be factorized into a ground intensity governing event timing and a conditional mark distribution (Daley & Vere-Jones, 2003). Dec-ODE builds on this formulation by modeling the evolving influences of individual past events with neural ordinary differential equations (Song et al., 2024). TransitionCast realizes a related factorization through the polar form of a learned transition representation, whose magnitude parameterizes the total transition intensity over the full horizon of the temporal decoder and whose direction predicts activity begin or end. Whereas Dec-ODE conditions on event histories, TransitionCast derives both factors from a learned temporal representation constructed from frozen visual features. The resulting temporal and semantic predictions receive separate supervision and are combined with current-state evidence by the State Composer to produce window-level four-state probabilities.

B Additional Experimental Analysis

We examine the sensitivity of SurgCast to two design choices in TransitionCast: the strength of semantic supervision and the number of subspaces K. We use the same CholecT50 setup and frozen DINOv3 ViT-L/16 backbone as in the main experiments. We denote the model with TransitionCast but without StateRecast by +TC, and the full model by +TC+SR.

Semantic supervision weight. The factor ablation (Table 5) shows that training the direction alone reduces mAP relative to the baseline, whereas training it jointly with the magnitude provides an additional gain over magnitude-only training. This suggests that the semantic changes encoded by the direction are more useful when combined with the temporal evidence provided by the magnitude. We therefore examine how semantic supervision strength affects the direction's standalone contribution and its interaction with the magnitude.

In the first-stage objective, the temporal and semantic losses are combined as

$$\mathcal{L}_{\text{polar}} = \frac{1}{2}\left(\mathcal{L}_{\text{time}} + \beta_{\text{dir}} M^{*} \mathcal{L}_{\text{sem}}\right), \tag{11}$$

Table 6: **Sensitivity to semantic supervision strength on CholecT50** ($K = 4$). Test mAP (%) is averaged over three seeds, with gains over the DINOv3-L baseline in percentage points. The shaded row marks the setting used in the main experiments.

	Validation	Test			
β_{dir}	+TC+SR	Direction only	+TC	+TC+SR	Synergy
0.5	39.15	38.07 +0.10	38.95 +0.98	40.02 +2.05	+0.01
1	39.33	37.98 +0.01	39.06 +1.09	40.34 +2.37	+0.21
2	39.43	37.23 −0.74	39.01 +1.04	40.36 +2.39	+0.91
2.5	39.45	37.21 −0.76	39.09 +1.12	40.33 +2.36	+1.01

Table 7: **Sensitivity to the number of subspaces on CholecT50.** mAP (%) is reported as mean ± sample standard deviation over three seeds. The shaded row marks $K = 4$, the setting used in the main experiments. The best mAP in each column is shown in **bold**.

	Validation		Test	
K	+TC	+TC+SR	+TC	+TC+SR
1	37.77±0.50	39.22±0.52	38.94±0.37	40.27±0.25
2	37.48±0.20	39.17±0.07	38.80±0.36	40.13±0.26
4	**37.82**±0.31	**39.45**±0.14	**39.09**±0.40	40.33±0.19
8	37.57±0.38	39.33±0.32	39.07±0.50	**40.40**±0.51
16	37.72±0.13	39.17±0.31	39.08±0.39	40.38±0.38

β_{dir} controls the relative strength of semantic supervision. With $K = 4$ and all other settings fixed, we increase β_{dir} and compare direction-only training with joint training of the direction and magnitude (Table 6).

The direction-only variant trains the semantic decoder using only the semantic loss, while keeping the temporal weights fixed at their initialization from the training event histogram. We define synergy as $\Delta_{\text{TC}} - \Delta_{\text{Dir.}} - \Delta_{\text{Mag.}}$, where each Δ denotes the mAP gain over the baseline. This measures the gain from joint training beyond the sum of the two standalone gains.

As semantic supervision increases, direction-only test mAP decreases from 38.07% to 37.21%, while +TC varies by only 0.14 percentage points and the full model remains between 40.02% and 40.36%. Synergy consequently increases from 0.01 to 1.01 percentage points, primarily reflecting the decline in direction-only performance rather than an increase in the joint gain. This pattern is consistent with the direction contributing through its interaction with temporal evidence in the State Composer.

The main experiments use $\beta_{\text{dir}} = 2.5$, for which the full model achieves validation and test mAP of 39.45% and 40.33%, respectively. Across these settings, the small variation in test mAP further indicates that the full model is robust to the strength of semantic supervision.

Number of subspaces. TransitionCast partitions the transition representation space into K orthogonal subspaces, each with separate temporal weights and a semantic decoder. This allows different transition patterns to have distinct temporal profiles and semantic effects. With the total representation dimension d fixed, increasing K reduces the dimension of each subspace to d/K, while $K = 1$ leaves the space unpartitioned. We examine how this partitioning granularity affects forecasting performance.

We compare $K \in \{1, 2, 4, 8, 16\}$, using the same orthogonal basis estimated on the training split (Table 7). Each subspace uses the same semantic decoder architecture, which maps its direction to a four-dimensional intermediate representation before predicting activity onsets and offsets.

Full-model test mAP varies modestly with K. It decreases from 40.27% at $K = 1$ to 40.13% at $K = 2$, then increases to 40.40% at $K = 8$ and remains similar at $K = 16$ (40.38%). For +TC, test mAP is nearly unchanged once $K \geq 4$, ranging from 39.07% to 39.09%. Thus, finer partitioning provides limited gains, and increasing the number of subspaces does not consistently improve performance.

On the validation split, $K = 4$ achieves the highest mAP for both +TC and +TC+SR. This is also the setting used in the main experiments.

C State Composer

The temporal and semantic decoders predict when transitions may occur and which activities may begin or end. However, the four state-change categories depend on whether an activity occurs in the observation and future windows. The State Composer connects these two descriptions by combining transition predictions with recognition evidence to construct a four-state distribution.

Recognition evidence and calibration. We use the current recognition probability $p^{\mathrm{rec}}_{t,c}$ and the observation-window maximum $p^{\mathrm{obs}}_{t,c} = \max_{\tau \in \mathcal{O}_t} p^{\mathrm{rec}}_{\tau,c}$ defined in the main text. The latter summarizes evidence of activity presence across the observation window, including before t. To adjust prediction confidence before composition, we apply temperature scaling, $g(p;T) = \mathrm{sigmoid}(\mathrm{logit}(p)/T)$, using a shared positive temperature for the two recognition probabilities and a separate shared temperature for the semantic probabilities $m^{\pm}_{t,c,\ell}$ in Eq. 4. Because g is increasing, calibrating the observation-window maximum is equivalent to taking the maximum of the calibrated recognition probabilities. We then recompute the difference to obtain $\widetilde{\boldsymbol{q}}_{t,c} = [\widetilde{p}^{\mathrm{rec}}_{t,c}, \widetilde{p}^{\mathrm{obs}}_{t,c}, \widetilde{p}^{\mathrm{obs}}_{t,c} - \widetilde{p}^{\mathrm{rec}}_{t,c}]^{\top}$. Positive activity- and second-specific scale factors adapt the shared transition intensities to each activity. All temperatures and scale factors are initialized to one and learned during the first training stage. Below, we omit the indices t, c for clarity.

Absence over the future window. An activity occurs in the future window if it is present at any future sampling time. We therefore compute the complementary probability that it is absent at all four future sampling times. For each future second $\ell \in \{1, \ldots, 4\}$, ending at $t + \ell$, we use the exponential mapping in Eq. 3 to obtain the calibrated transition occurrence probability $a_\ell = 1 - \exp(-s_\ell \lambda_\ell)$, where $s_\ell > 0$ is the learned intensity scale. Multiplying a_ℓ by the calibrated semantic probabilities $\widetilde{m}^{+}_{\ell}$ and $\widetilde{m}^{-}_{\ell}$ gives the absent-to-present and present-to-absent transition probabilities, respectively.

We model activity presence at successive sampling times as a two-state Markov process with these transition probabilities. Let A_0 and A_1 denote the probabilities of absence at all future sampling times, conditional on absence and presence at t, respectively:

$$\begin{aligned} A_0 &= \prod_{\ell=1}^{4} \big(1 - a_\ell \widetilde{m}^{+}_{\ell}\big), \\ A_1 &= a_1 \widetilde{m}^{-}_{1} \prod_{\ell=2}^{4} \big(1 - a_\ell \widetilde{m}^{+}_{\ell}\big). \end{aligned} \tag{12}$$

The first expression multiplies the absent-to-absent transition probabilities across all four future seconds. The second combines a present-to-absent transition in the first second with absent-to-absent transitions in the remaining three seconds.

Four-state composition. We use the calibrated recognition probabilities to assign mass to three cases: $1 - \widetilde{p}^{\mathrm{obs}}$ for absence at all observed sampling times, $\widetilde{p}^{\mathrm{obs}} - \widetilde{p}^{\mathrm{rec}}$ for presence at an earlier sampling time but absence at t, and $\widetilde{p}^{\mathrm{rec}}$ for presence at t. Combining these three cases with the conditional absence probabilities gives

$$\begin{aligned} P^{\mathrm{bg}} &= (1 - \widetilde{p}^{\mathrm{obs}}) A_0, \\ P^{\mathrm{on}} &= (1 - \widetilde{p}^{\mathrm{obs}})(1 - A_0), \\ P^{\mathrm{dis}} &= \widetilde{p}^{\mathrm{rec}} A_1 + (\widetilde{p}^{\mathrm{obs}} - \widetilde{p}^{\mathrm{rec}}) A_0, \\ P^{\mathrm{con}} &= \widetilde{p}^{\mathrm{obs}} - P^{\mathrm{dis}}. \end{aligned} \tag{13}$$

In particular, discontinuity includes an activity that is present at t and one that occurred only earlier in the observation window, provided that it is absent at every future sampling time in either case.

Continuity includes the remaining cases in which the activity occurs in both windows, even if it is absent at t.

Since $0 \leq \widetilde{p}^{\mathrm{rec}} \leq \widetilde{p}^{\mathrm{obs}} \leq 1$ and $A_0, A_1 \in [0, 1]$, the four probabilities are non-negative and sum to one, forming $\boldsymbol{p}^{\mathrm{comp}} = [P^{\mathrm{con}}, P^{\mathrm{dis}}, P^{\mathrm{on}}, P^{\mathrm{bg}}]^\top$. This distribution corrects the base logits through Eq. 5, after which softmax yields the four-state probabilities predicted by TransitionCast.