# FUNCTIONAL AGREEMENT OF COMPUTATIONALLY EQUIVALENT STATES

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Learned stateful sequential models can implement the same semantic computation through different histories but arrive at different internal states. The behavior of these states may diverge during subsequent model rollouts. We introduce Functional Agreement of Computationally Equivalent States (FACES), an auxiliary loss that compares their future behavior under the same continuation. This objective encourages future behavioral agreement without forcing the internal states themselves to be equal. We introduce several FACES variants that target agreement at different future depths. Geometrically, we show that the objective induces a local quadratic form that weights state-perturbation directions according to their effects on future behavioral readouts. We first perform a mechanistic study on simple affine programs and show that learned actions can support much more accurate long-horizon execution under exact affine dynamics. We also show that route-induced state differences can have strongly direction-dependent effects on future behavior. We then show that FACES reduces long-horizon prediction error relative to ordinary supervised training and immediate agreement on controlled visual $\mathrm{SE}(2)$ transformation sequences. Using real-world data, we show that multiscale AFACES achieves the lowest mean test error among the evaluated methods across temporal factorizations of the EuRoC IMU trajectories. We also show that continuation agreement improves over immediate agreement on a discrete, noninvertible problem. These results show that continuation-based behavioral regularization can improve both the interchangeability of states representing the same semantic computation and task performance.

## 1 INTRODUCTION

A stateful sequential model can execute two semantically equivalent histories correctly and still fail to treat their resulting states as interchangeable. The difference may be invisible in the current output, then emerge after computation continues. This exposes a gap in ordinary task supervision: it trains predictions along observed histories, but does not require histories that represent the same computation to support the same later behavior. This problem arises across very different sequential computations. For example, transformations can be decomposed or composed, and the same dynamical evolution can be factorized at different temporal scales. These different histories represent the same semantic computation and should therefore support the same future behavior.

We turn these known semantic equivalences into a source of supervision. We introduce *Functional Agreement of Computationally Equivalent States* (FACES). For equivalent histories $p \simeq q$ starting from the same state $h$, we apply the same continuation $u$ to both resulting states and train their task-relevant readouts to agree:

$$B_u(H_\theta(p;h)) \approx B_u(H_\theta(q;h)), \qquad B_u(h) = Q_\omega(H_\theta(u;h)). \tag{1}$$

Rather than forcing internal states to be equal, we ask whether one can replace the other without changing future task behavior. The continuation acts as a probe to expose route-dependent differences that are invisible at the immediate task readout but consequential after later transitions.

Prior work studies path sensitivity in learned trajectories (Gu, 2026), prediction invariance across valid computation paths (Li et al., 2026a), and consistency between direct and composed transitions (Shikhman, 2026; Tang et al., 2026). Predictive state representations also motivate defining a state



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.

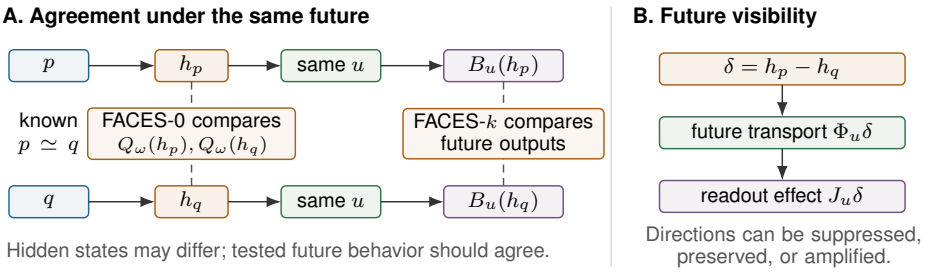



Figure 1: **Overview of FACES. (A)** Semantically equivalent computation histories are compared under the same future continuation. **(B)** Their functional importance depends on how route differences propagate through future dynamics before reaching the task readout.

through its future observable behavior (Littman & Sutton, 2001; Hefny et al., 2018). FACES builds on this view by turning known equivalences between histories into an objective for future behavioral interchangeability. Our local analysis shows how the objective works: FACES weights state-difference directions by their effects on future readouts, rather than by hidden-state distance alone.

We first study this mechanism on affine programs, where we can separate the learned primitive actions from recurrent model error and examine how route-induced perturbations affect later outputs. We then show on a more complex but still controlled visual $\mathrm{SE}(2)$ task that continuation agreement improves long-horizon execution beyond immediate agreement and additional task supervision. Next, on real EuRoC inertial data, we find that the useful continuation horizon changes across temporal factorizations of the same motion; this motivates MS-AFACES, which improves mean performance across these factorizations. We also show that the benefit extends to discrete, noninvertible computation.

Our contributions are: **(1)** We introduce FACES, which turns known equivalences between computation histories into supervision for future behavioral interchangeability, together with exact-depth, cumulative, anchored, and multiscale variants; **(2)** We show that FACES induces a continuation-dependent local geometry that weights state differences by their effects on future readouts, and characterize when agreement under shared continuations provides supervision beyond immediate agreement; and **(3)** We show mechanistically that small route-induced state differences can have strongly direction-dependent future effects, then demonstrate that continuation agreement improves long-horizon execution across visual $\mathrm{SE}(2)$ sequences, real-world EuRoC inertial data, and discrete noninvertible computation.

## 2 Functional Agreement of Computationally Equivalent States

Let $\mathcal{H} \subseteq \mathbb{R}^d$ be an internal state space and $\mathcal{A}$ a control space. A learned transition $F_\theta : \mathcal{H} \times \mathcal{A} \to \mathcal{H}$ updates $h_{t+1} = F_\theta(h_t, a_t)$, and a task readout $Q_\omega : \mathcal{H} \to \mathbb{R}^m$ produces $y_t = Q_\omega(h_t)$.

We call a sequence of controls $p = (a_1, \ldots, a_T)$ that are applied sequentially to an initial state $h$ a computation history or program. Starting from $h$, the resulting rollout state is:

$$H_\theta(p; h) = F_\theta(\cdots F_\theta(F_\theta(h, a_1), a_2) \cdots, a_T). \tag{2}$$

A *semantic primitive* $g$ is a known operation representable by one model step; $a(g) \in \mathcal{A}$ denotes its model-facing control encoding. When primitives admit a known composition law and their composition is also representable in one step, they yield exact equivalent routes: $(g_1, g_2) \simeq (g_2 \circ g_1)$.

More generally, $p \simeq q$ denotes an externally specified equivalence between histories implementing the same semantic computation. Such equivalences can come from composition, temporal refactorization, or reorderings known to preserve the result; they need not arise from semantic primitives.

Let $u = (u_1, \ldots, u_j)$ be an admissible future computation, where $u_k \in \mathcal{A}$ are future controls. We assume that this semantic equivalence is preserved when the same future controls are appended:

$$p \simeq q \quad \Longrightarrow \quad pu \simeq qu, \tag{3}$$

where $pu$ denotes applying computation history $p$ followed by $u$.

The states $H_\theta(p;h)$ and $H_\theta(q;h)$ may retain route-dependent information not reflected in the current task readout. We require *functional interchangeability* under shared futures, allowing their hidden states to differ.

### 2.1 Agreement under a shared continuation

For an admissible continuation $u$, we define the behavior induced by $u$ from an internal state $h$ as

$$B_u(h) = Q_\omega(H_\theta(u;h)), \tag{4}$$

where the model parameters are left implicit in $B_u$ for notational simplicity. For the empty continuation $\varnothing$, we take $H_\theta(\varnothing;h) = h$, so that $B_\varnothing(h) = Q_\omega(h)$.

For an equivalent pair $p \simeq q$ starting from the same initial state $h$, define the exact-depth functional disagreement after $j$ shared future steps as

$$\ell_j(p,q;h,u_{1:j}) = d\big(B_{u_{1:j}}(H_\theta(p;h)), B_{u_{1:j}}(H_\theta(q;h))\big), \tag{5}$$

where $d(\cdot,\cdot)$ is an application-appropriate discrepancy in the task-relevant behavioral space, such as squared error or a divergence between predictive distributions.

### 2.2 FACES objectives

Let $\mathcal{L}_j = \mathbb{E}_{(h,p,q),\,u_{1:j}}[\ell_j(p,q;h,u_{1:j})]$ be the expected disagreement after exactly $j$ shared future steps. FACES-$K$ uses $\mathcal{L}_{\text{FACES-}K} = \mathcal{L}_K$, with FACES-0 giving immediate agreement. For depth weights $\boldsymbol{\alpha} = (\alpha_0, \ldots, \alpha_K)$, define

$$\mathcal{L}_{\text{FACES}}(\boldsymbol{\alpha}) = \sum_{j=0}^{K} \alpha_j \mathcal{L}_j, \qquad \alpha_j \geq 0, \qquad \sum_{j=0}^{K} \alpha_j = 1. \tag{6}$$

A common rescaling of the weights is absorbed into the training coefficient. FACES-$K$ uses $\alpha_j = \mathbb{1}[j = K]$.

**Cumulative FACES.** When agreement is enforced uniformly from depth 0 through $K$, we obtain FACES-$\leq K$, with $\mathcal{L}_{\text{FACES-}\leq K} = \frac{1}{K+1}\sum_{j=0}^{K} \mathcal{L}_j$.

**Multiscale FACES.** When no single cumulative horizon is appropriate for all route defects, MS-FACES combines FACES-$\leq k$ objectives according to a distribution $\pi$ over $k \in \{0,\ldots,K\}$: $\mathcal{L}_{\text{MS-FACES}} = \mathbb{E}_{k\sim\pi}[\mathcal{L}_{\text{FACES-}\leq k}]$. For uniformly weighted cumulative objectives, this is equivalent to the exact-depth weights

$$\alpha_j^{\text{MS}} = \sum_{k=j}^{K} \frac{\pi(k)}{k+1}. \tag{7}$$

Thus, FACES-$K$, FACES-$\leq K$, and MS-FACES differ only in how agreement is weighted across continuation depths.

**Anchored FACES.** When one route is more reliable, symmetric agreement can move it toward the weaker route. Anchored FACES (AFACES) instead uses a reference route $p_T$ and an equivalent student route $p_S$, with $p_T \simeq p_S$, and defines

$$\ell_j^{\text{A}}(p_T,p_S;h,u_{1:j}) = d\big(\text{sg}\big[B_{u_{1:j}}(H_\theta(p_T;h))\big], B_{u_{1:j}}(H_\theta(p_S;h))\big), \tag{8}$$

where $\text{sg}[\cdot]$ denotes stop-gradient. Defining $\mathcal{L}_j^{\text{A}} = \mathbb{E}_{(h,p_T,p_S),\,u_{1:j}}[\ell_j^{\text{A}}(p_T,p_S;h,u_{1:j})]$, the general anchored objective is $\mathcal{L}_{\text{AFACES}}(\boldsymbol{\alpha}) = \sum_{j=0}^{K} \alpha_j \mathcal{L}_j^{\text{A}}$. AFACES-$K$, AFACES-$\leq K$, and MS-AFACES use the same depth weights as their symmetric counterparts. Stop-gradient blocks the auxiliary gradient through the reference branch; shared parameters still receive updates through the student branch and task supervision.

**Training objective.** FACES is used as an auxiliary objective alongside ordinary task supervision. For a symmetric variant, $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \eta_{\text{FACES}}\mathcal{L}_{\text{FACES}}(\boldsymbol{\alpha})$, where $\eta_{\text{FACES}}$ controls the auxiliary loss strength. For AFACES, $\mathcal{L}_{\text{FACES}}$ is replaced by $\mathcal{L}_{\text{AFACES}}$.

# 3 When Shared-Future Agreement Matters

For a fixed model, we examine what immediate agreement can miss, which state differences shared futures penalize, and how depth weighting and task supervision shape the objective. Proofs and extensions are in Appendix A.

**Why current agreement can miss future differences.** If $Q(x, z) = x$ and $F(x, z) = (z, x)$, the states $(0, 0)$ and $(0, \epsilon)$ have equal readouts initially but readouts differing by $\epsilon$ after one shared step. Conversely, a state difference $(\epsilon, 0)$ gives equal readouts after that step. Agreement at one depth need not imply agreement at another.

On a forward-invariant reachable set with common admissible actions, immediate readout equality is preserved under every shared future exactly when the next readout depends only on the current readout and action: $Q \circ F_a = f_a \circ Q$, where $F_a(h) = F_\theta(h, a)$.

**Which state differences shared futures penalize.** For $p \simeq q$, write $h = H_\theta(q; h_0)$ and $\delta = H_\theta(p; h_0) - h$, the route defect between their internal states. Holding the conditional future distribution fixed, define

$$J_u(h) = D_h B_u(h), \qquad \Gamma_j(h) = \mathbb{E}_{u\sim\mathcal{U}_j}[J_u(h)^\top J_u(h)]. \tag{9}$$

$J_u(h)$ is the Jacobian of $B_u$ at $h$ and $D_h$ is differentiation with respect to the hidden-state argument.

**Proposition 1 (Expected quadratic discrepancy)** *Suppose each supported $B_u$ is $C^2$ on a common convex neighborhood, with uniformly bounded first and second derivatives. Then, for small $\delta$,*

$$\mathcal{J}_j^h(\delta) := \mathbb{E}_u\|B_u(h + \delta) - B_u(h)\|^2 = \delta^\top \Gamma_j(h)\delta + O(\|\delta\|^3). \tag{10}$$

*If $u$ and $\delta$ are conditionally independent given $h$, let $M_{\text{route}}(h) = \mathbb{E}[\delta\delta^\top \mid h]$ be the conditional second-moment matrix. For $\delta$ supported in a sufficiently small neighborhood of zero,*

$$\mathbb{E}_\delta[\mathcal{J}_j^h(\delta) \mid h] = \operatorname{tr}(\Gamma_j(h)M_{\text{route}}(h)) + O(\mathbb{E}_\delta[\|\delta\|^3 \mid h]). \tag{11}$$

For route differences $\delta = \epsilon v$, $v \in \ker \Gamma_0$ and $v^\top \Gamma_j v > 0$ identify a direction penalized only by future agreement to quadratic order. Futures can also reweight directions outside $\ker \Gamma_0$. Thus the penalty depends on how the actual route differences align with future sensitivity, rather than on state distance alone. The affine diagnostic examines this distinction using a behavioral-Jacobian decomposition of observed route defects (Section 4.1).

**Depth weighting and task supervision.** Mixtures $\Gamma_\alpha = \sum_j \alpha_j \Gamma_j$ with positive weights on the same depths have the same null space under matching future distributions. Thus, MS-AFACES and cumulative-4 differ in how they weight those depths, not in which directions they can penalize. To see when agreement provides supervision beyond the ordinary task loss, consider predictions $y_p, y_q$ with common target $y^\star$:

$$\frac{\|y_p - y^\star\|^2 + \|y_q - y^\star\|^2}{2} = \left\|\frac{y_p + y_q}{2} - y^\star\right\|^2 + \frac{\|y_p - y_q\|^2}{4}. \tag{12}$$

Thus, task supervision of both routes already penalizes their disagreement; FACES reweights this term and can extend agreement supervision to future pairs not directly constrained by the task loss.

# 4 Experiments

**Continued training from a competent common checkpoint.** We study whether FACES can improve task execution in a partially trained but sufficiently competent model. In an untrained model, the encoded states and transitions may not yet represent useful behavior, allowing trivial agreement without solving the task. We therefore first train with ordinary task supervision to obtain a competent checkpoint, then branch from it to train all methods. Within each comparison, every method starts from the same checkpoint and retains task supervision. The ordinary baseline receives only further task updates, while the FACES models also receive the agreement loss. This controls the starting model and the benefit of additional task training.

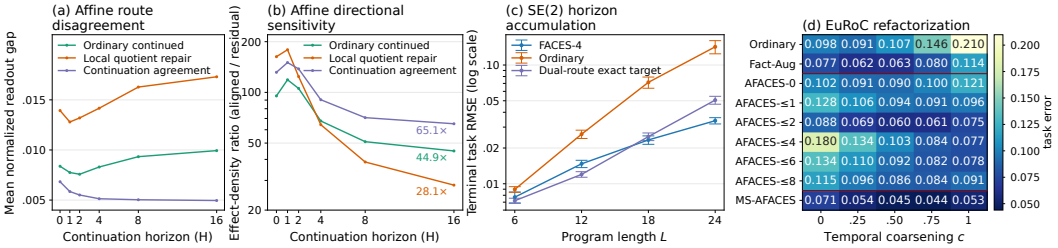



Figure 2: **Route effects, directional sensitivity, and continuation scale.** (a) Normalized readout disagreement after equivalent affine routes and a shared length-$H$ future. (b) Aligned-to-residual effect-density ratio (Appendix B.5); larger values indicate greater directional sensitivity. (c) Range-matched terminal RMSE for visual $\mathrm{SE}(2)$ sequences, shown as mean $\pm$ sample standard deviation over five seeds on a log scale. (d) EuRoC test error across temporal coarsening $c$, averaged over five seeds. Fact-Aug directly supervises factorized trajectories. Lower is better in (a), (c), and (d).

### 4.1 Affine programs: long-horizon error and directional sensitivity

We first conduct a controlled experiment on affine programs to investigate long-horizon error in the recurrent model. Each primitive $g_{\alpha,\beta}(x) = \alpha x + \beta$ is an affine transformation. For affine transformations, we can construct equivalent routes by replacing two successive primitives with their composition. We use a controller to infer a two-dimensional action from an embedding of the current primitive, and a gated recurrent unit (GRU) (Cho et al., 2014) as the dynamical model. The readout predicts the accumulated affine map after each primitive. We train on trajectories of length 2-6, then test on trajectories up to length 96. Appendix B gives the complete task construction, model, training protocol, diagnostics, and results.

**Diagnosing length-96 error.** We first train an ordinary model, then ask whether its higher length-96 error relative to the training horizons comes from the learned actions. We freeze the controller and train a small nonlinear probe to predict $(\alpha, \beta)$ from these actions using groundtruth supervision. We then replace the GRU with the exact affine dynamics. Length-96 map RMSE falls from $0.23137$ to $0.02815$, an $87.83\%$ reduction. Since the learned actions support much more accurate execution, we next examine how differences produced by the recurrent model affect future behavior.

**Behavioral sensitivity to route perturbations.** Applying two primitives sequentially and applying their composition directly represent the same affine transformation but produce different hidden states. Since hidden states have no unique groundtruth target, we study the route-induced perturbation $\delta = h_{\mathrm{seq}} - h_{\mathrm{dir}}$ and examine its effect on future outputs. Using the immediate and four one-step readouts, we split $\delta$ into a Jacobian-aligned component and a residual, then propagate both through the same future. The aligned component contains only a small share of the state-difference energy but has a much larger behavioral effect per unit energy. Thus, the effect of a route perturbation depends strongly on its direction, rather than only on its magnitude.

**Reducing the effect of route perturbations.** This motivates us to train directly against the future behavioral effect of route perturbations. We compare ordinary continued training, local quotient repair, and continuation agreement in a separate seed-42 diagnostic. Local quotient repair targets transitions that make weakly observable perturbations observable, while continuation agreement penalizes the effect of the actual route perturbations after shared futures. At $H = 16$, the aligned-to-residual effect-density ratios are $44.9$, $28.1$, and $65.1$, respectively, showing that the behavioral effect remains strongly direction-dependent (Fig. 2b). Continuation agreement produces the smallest overall route discrepancy across the tested horizons (Fig. 2a).

**Improving length-96 performance.** We finally test whether training against route disagreement under shared futures improves long-horizon task performance. Across five seeds, Continuation FACES reduces mean length-96 map RMSE from $0.10626$ under ordinary training and $0.07724$ under FACES-0 to $0.05794$. It improves over ordinary training on all five seeds and over FACES-0 on four of five. FACES-0 applied after a random history obtains $0.09888$, showing that the improvement does

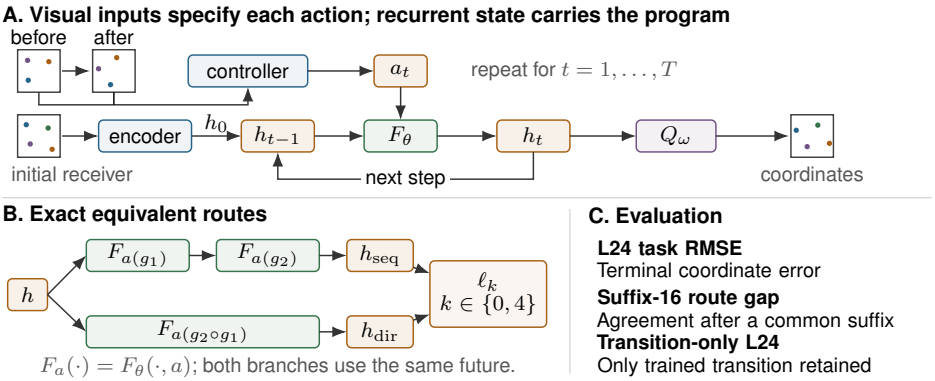



Figure 3: **Visual** $\mathrm{SE}(2)$ **transformation sequences. (A)** Visual pairs specify primitives; recurrent state carries their accumulated effect on a separate receiver. **(B)** Sequential and composed routes implement the same transformation; FACES compares their readouts immediately or after a shared future. **(C)** Evaluation measures long-sequence accuracy, future route agreement, and performance retaining only the trained transition module.

not come simply from applying immediate agreement at deeper recurrent states. For this controlled comparison, we restrict the FACES gradient to the recurrent model.

### 4.2 Visual $\mathrm{SE}(2)$ transformations: immediate versus continuation agreement

After the affine study, we increase the task complexity by adding visual perception and a separate receiver scene while retaining exact control over the underlying transformations. In our visual $\mathrm{SE}(2)$ task, a before/after image pair demonstrates a transformation that must be applied to a new receiver scene. This setup resembles Edit Transfer, which infers an edit from source-target image pairs and applies it to a new query image (Chen et al., 2025). Because planar $\mathrm{SE}(2)$ transformations compose exactly, we still know the equivalent routes and groundtruth future targets. This allows us to compare continuation agreement, immediate agreement, and additional task supervision across five seeds.

**Task and equivalent routes.** Each primitive $g \in \mathrm{SE}(2)$ is a planar rotation and translation. We render it as a before/after image pair, from which a controller infers a three-dimensional action $a(g)$ without receiving the transformation parameters. The demonstration and receiver scenes are generated independently, so the controller must infer the transformation and transfer it to new content. The receiver scene, represented by coordinates and occupancy masks, is encoded once to initialize a GRU state (Cho et al., 2014). Subsequent transitions receive only the inferred actions and previous state, and a readout predicts the receiver coordinates after each primitive (Fig. 3).

Independently rendered demonstrations of $g_1$, $g_2$, and $g_2 \circ g_1$ give two routes that implement exactly the same receiver transformation, producing $h_{\text{seq}} = F_\theta(F_\theta(h, a(g_1)), a(g_2))$ and $h_{\text{dir}} = F_\theta(h, a(g_2 \circ g_1))$, respectively. Training sequences have lengths 2-6; evaluation uses lengths 6, 8, 12, 18, and 24. We retain evaluation sequences whose exact trajectories stay within the training coordinate range to isolate accumulation error from range extrapolation. For each seed, ordinary continued training, FACES-0, and FACES-4 start from the same pretrained checkpoint and use the same task stream. FACES-4 compares receiver predictions after four shared future primitives.

We report terminal coordinate root mean squared error (RMSE) at length 24 (*L24 task RMSE*), the coordinate discrepancy between the two route predictions after 16 shared future primitives (*Suffix-16 route gap*), and *Transition-only L24*. The last retains each trained transition module while restoring the controller, receiver encoder, and readout from the common pretrained checkpoint. Appendix C gives the complete task construction, architecture, sampling, optimization, controls, and results.

**Continuation benefit.** Across five seeds, FACES-0 reduces mean L24 error by $51\%$ relative to ordinary training; FACES-4 reduces it by a further $52\%$ (Table 1). FACES-4 improves all three metrics over FACES-0 on every seed. Its smaller Suffix-16 route gap shows that the route predictions

Table 1: Visual SE(2) results averaged over five seeds. Transition-only restores all modules except the trained transition. Lower is better.

| Method | L24 task RMSE | Suffix-16 gap | Transition-only L24 |
|---|---|---|---|
| Ordinary continued training | 0.1424 | 0.01483 | 0.2064 |
| FACES-0 | 0.0702 | 0.00806 | 0.1822 |
| **FACES-4** | **0.0340** | **0.00362** | **0.1095** |
| Single-route exact target | 0.0516 | 0.00672 | 0.1399 |
| Dual-route exact target | 0.0506 | 0.00662 | 0.1430 |
| Additive-action control | 0.1561 | 0.01524 | 0.2789 |

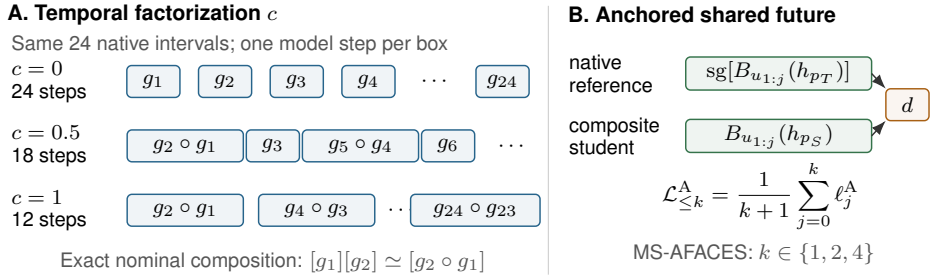



Figure 4: **EuRoC temporal factorization and continuation.** Left: grouping adjacent pairs among 24 native intervals preserves the nominal motion while reducing recurrent steps from 24 ($c = 0$) to 12 ($c = 1$). Right: AFACES-$\leq k$ averages depths $0, \ldots, k$; MS-AFACES mixes cumulative horizons $k \in \{1, 2, 4\}$ independently of $c$.

remain closer after a longer shared future, while its lower Transition-only L24 error shows that the gain remains when only the trained transition module is retained. Figure 2c shows the error across sequence lengths for ordinary training, FACES-4, and the dual-route exact-target control.

**Controls for supervision depth.** The single-route and dual-route exact-target controls use the same four future transitions as FACES-4, supervising one or both routes independently against their exact future target. Both improve over ordinary training. FACES-4 has approximately $34\%$ and $33\%$ lower mean L24 error than the single-route and dual-route controls, respectively, and improves all three metrics over both controls on every seed. Its advantage is specific to longer sequences: FACES-0 and the exact-target controls have lower mean error at lengths 6-12.

**Control for action composition.** We also test whether directly regularizing the action representation is sufficient. An ALAM-inspired control (Tang et al., 2026) encourages $a(g_2 \circ g_1) \approx a(g_1) + a(g_2)$ and $a(g_1) + a(g_1^{-1}) \approx 0$. All three metrics are worse on average than under ordinary training. Thus, enforcing this additive action structure does not reproduce the shared-future agreement benefit.

### 4.3 EuRoC inertial odometry across temporal factorizations

After the two controlled synthetic studies, we next test whether FACES can provide a benefit on real-world data. We use IMU trajectories from EuRoC (Burri et al., 2016). Although the measurements are noisy, their preintegrated IMU increments admit a known nominal composition rule that lets us construct equivalent routes (Forster et al., 2017). In this setting, we find that AFACES improves performance when the same physical trajectory is processed through different temporal factorizations. Such temporal refactorization commonly arises in robotics because sampling rates, integration windows, batching, or keyframe selection can change the number and scale of recurrent updates. We also find that the preferred fixed continuation depth changes across these factorizations. This leads us to develop MS-AFACES, which combines several continuation horizons.

**Task and equivalent routes.** We study inertial odometry from EuRoC IMU sequences. The task is to estimate the navigation state after each approximately $50\,\mathrm{ms}$ native interval. Each interval is

Table 2: EuRoC validation error. *All* averages over the five factorizations; *Coarse* averages over $c = .75, 1$. Lower is better.

| Method | All | $c = 0$ | Coarse |
|---|---|---|---|
| Ordinary | 0.10143 | 0.07443 | 0.14620 |
| Fact-Aug | 0.07950 | **0.07364** | 0.09811 |
| AFACES-0 | 0.09214 | 0.08894 | 0.10597 |
| AFACES-$\leq 1$ | **0.07082** | 0.08309 | 0.07044 |
| AFACES-$\leq 2$ | 0.08833 | 0.11411 | 0.07880 |
| AFACES-$\leq 4$ | 0.07263 | 0.11500 | **0.05352** |
| MS-AFACES | **0.07006** | 0.08032 | 0.06843 |

Table 3: EuRoC test error, averaged over the five factorizations. Lower is better.

| Method | Mean |
|---|---|
| Ordinary | 0.13034 |
| Fact-Aug | 0.07902 |
| AFACES-0 | 0.10103 |
| AFACES-$\leq 1$ | 0.10312 |
| AFACES-$\leq 2$ | **0.07049** |
| AFACES-$\leq 4$ | 0.11556 |
| MS-AFACES | **0.05330** |

represented by a deterministic preintegrated IMU increment $(\Delta R, \Delta v, \Delta p, \Delta t)$ constructed from the standard discrete IMU kinematics (Forster et al., 2017). Our recurrent estimator applies nominal IMU propagation, then learns a bounded residual correction. An MLP embeds the increment and current navigation state, and a GRU predicts the correction. We supervise the navigation state after every native interval. We use six EuRoC sequences for training, two for validation, and three difficult sequences for testing. We report $E = (\mathrm{RMSE}_p/s_p)^2 + (\mathrm{RMSE}_v/s_v)^2$, where $s_p$ and $s_v$ are fixed dataset scales, and average results over five training seeds. Figure 2d shows the test surface across continuation horizons and factorizations; Table 2 reports validation summaries. Appendix D gives the complete data processing, model, route construction, gradient treatment, training protocol, and results.

To construct equivalent routes, we exactly compose two adjacent native increments as $g_{1,2} = g_2 \circ g_1$. The sequential route applies $[g_1][g_2]$, while the composed route applies $[g_{1,2}]$ in one recurrent step. AFACES uses the sequential route as the stop-gradient reference because it is directly supervised during ordinary task training, and uses the composed route as the student. We then append the same native continuation to both routes. For cumulative horizon $k$, $\mathcal{L}_{\text{AFACES-}\leq k} = (k+1)^{-1} \sum_{j=0}^{k} \mathcal{L}_j^{\mathrm{A}}$. During the auxiliary update, the future rollout passes gradients to the student endpoint, but not to the transition parameters used inside that future rollout.

**Continuation depth depends on temporal factorization.** To change temporal factorization without changing the physical motion, we start from the same 24 native intervals and group a fraction $c \in \{0, .25, .5, .75, 1\}$ of adjacent pairs. Each grouped pair is replaced by its exact composition. Thus, $c = 0$ executes the trajectory using 24 recurrent steps, while $c = 1$ executes the same trajectory using 12 larger steps (Fig. 4). As a control for exposure to coarsened inputs, Fact-Aug continues training with equal weight on native trajectories and randomly factorized versions of the same trajectories. Both receive direct task supervision, and no agreement loss is used. The useful continuation depth changes across these factorizations. Among the fixed horizons in Table 2, AFACES-$\leq 1$ has the lowest validation error averaged over $c$, while AFACES-$\leq 4$ performs best at the two coarsest factorizations. Fact-Aug performs best at $c = 0$ among the non-multiscale methods but does not match the fixed AFACES models on coarser execution. We test $k \in \{0, 1, 2, 4, 6, 8\}$; the table shows $k \leq 4$.

**Combining continuation horizons.** This tradeoff motivates MS-AFACES. We combine cumulative horizons $k \in \{1, 2, 4\}$ using $\pi = \mathrm{Uniform}\{1, 2, 4\}$, giving $\mathcal{L}_{\mathrm{MS}} = \frac{1}{3}(\mathcal{L}_{\text{AFACES-}\leq 1} + \mathcal{L}_{\text{AFACES-}\leq 2} + \mathcal{L}_{\text{AFACES-}\leq 4})$. MS-AFACES has the lowest validation error averaged over $c$ (Table 2). AFACES-$\leq 4$ remains better at the two coarsest settings, but has substantially higher error at $c = 0$. Fact-Aug reduces the test mean from $0.13034$ under ordinary training to $0.07902$, but remains above MS-AFACES at $0.05330$. On the test set, MS-AFACES reduces mean error across $c$ from $0.07049$ for AFACES-$\leq 2$, the lowest-error fixed horizon in the test grid, to $0.05330$, a $24.4\%$ reduction (Table 3). It also outperforms, on all five seeds, a per-seed oracle that retrospectively selects the best $k \in \{1, 2, 4\}$. Combining horizons therefore performs well across temporal factorizations whose preferred fixed horizon differs.

### 4.4 Discrete computation and task qualification

**Discrete noninvertible computation.** We use SparseWriter to test whether continuation agreement improves over immediate agreement for discrete and noninvertible operations. We construct a simple memory-writing task whose primitives overwrite selected locations. Exact composition provides equivalent histories even though overwritten values cannot be recovered. Across five seeds, FACES-4 improves both route agreement and task error over FACES-0, reducing length-24 bit-error rate from $0.39679$ to $0.34115$ ($14.0\%$) and the suffix-16 route gap from $0.03965$ to $0.03369$ ($15.1\%$). Appendix E gives the construction and full results.

**Task qualification and negative findings.** To identify applicable settings for FACES, we searched prior work for stateful models with plausible equivalent routes. For each candidate, we constructed a route pair, shared continuation, and behavioral readout, then used the analysis in Section 3 to guide practical prescreening criteria. We reject candidates when future computation removes route differences, agreement largely duplicates existing supervision, or improved agreement does not yield consistent task gains. Appendix F gives the constructions and results.

## 5 Related Work

**Equivalent computations and structured transitions.** The *Propagation Field* studies path sensitivity of internal trajectories and outputs (Gu, 2026). PCEvo enforces prediction invariance across chemically valid paths with common endpoints (Li et al., 2026a), while semigroup consistency compares direct and composed temporal evolution (Shikhman, 2026). Related approaches study invariance to asynchronous algorithm execution and contract-equivalent specification histories (Dudzik et al., 2024; Monteagudo-Lago et al., 2024; Wu et al., 2026). AC-LAM, ALAM, and group-action methods instead encourage algebraic structure in learned actions or transitions (Wei et al., 2026; Tang et al., 2026; Keurti et al., 2023; Wang et al., 2026).

**Future behavior and multi-step consistency.** Predictive state representations describe histories through action-conditioned future predictions (Littman & Sutton, 2001; Hefny et al., 2018). Methods based on testing equivalence, bisimulation, and invariant representation learning compare states through future observable or task-relevant consequences (De Nicola & Hennessy, 1984; Ferns et al., 2011; Gelada et al., 2019; Zhang et al., 2021). Behavior Consistency evaluates predicted states through a fixed agent, while Hidden APIs compares language-model states through response distributions under future operations (Huang et al., 2026; Ma et al., 2026). Multi-step latent prediction and consistency objectives also shape representations over future horizons (Hafner et al., 2019; Schwarzer et al., 2021; Zhong et al., 2026; Bhand & Joshi, 2026), and FutureBridge uses a fixed shared future to evaluate alternative intermediate choices (Li et al., 2026b). Unlike objectives that match latent predictions, impose algebraic structure, or test equivalence under all possible futures, FACES uses externally known equivalent histories to supervise the task behavior of their resulting states under sampled shared continuations.

## 6 Discussion and Conclusion

We asked whether a model can continue interchangeably from states reached by equivalent histories. The results show why immediate agreement is not enough: small route-induced differences can have strongly directional future effects, and training against these effects improves long-horizon execution on visual $\mathrm{SE}(2)$ and SparseWriter. On EuRoC, the relevant continuation depth changes with temporal factorization, motivating multiscale agreement. The local geometry connects these findings by measuring how future computation transforms a state difference into output disagreement.

FACES requires known equivalent histories, persistent state, and shared continuations exposing relevant route differences. Our qualification checks reject candidates where futures erase those differences, supervision already constrains them, or agreement does not track correctness. Otherwise, FACES turns semantic structure into future-behavior supervision without forcing a particular hidden-state representation.

### AI USE STATEMENT

We used generative AI tools throughout this project as research and writing assistants. We used them to help develop and critique our initial ideas and conceptual formulations, refine hypotheses and experimental designs, generate synthetic datasets used in our experiments through AI-assisted code, assist with mathematical derivations and proofs, implement and review experimental code, interpret experimental results, maintain and organize experimental artifacts, and assist with translation. We also used them for literature search and summarization, brainstorming, editing and restructuring manuscript text, preparing figures and other research artifacts, identifying potential candidate tasks for FACES, and refining task-specific experimental settings. We additionally used generative AI for extensive literature searches aimed at challenging and assessing the novelty of the research ideas developed in this project. We did not use generative AI to clean or reformat datasets or to support qualitative or thematic data analysis.

We reviewed and verified all AI-assisted work in this project. We manually checked all AI-assisted mathematical arguments and proofs, including repeated verification of the arguments used in the final manuscript. We manually inspected AI-assisted code before execution. For the major experiments reported in the paper (Affine, SE(2), EuRoC, and SparseWriter), we reviewed the corresponding code multiple times and manually debugged it step by step. We manually inspected experimental artifacts, experimental data, and reported results, and cross-checked them against the implementation and experimental records. We reviewed experimental records maintained or logged with AI assistance before incorporating them into the project records. We also reviewed and evaluated experimental procedures suggested with AI assistance before adopting them. We manually checked all experimental details, mathematical statements, manuscript claims, citations, and quotations against the corresponding experimental records, derivations, implementations, and original sources. We also reviewed all AI-assisted manuscript text and verified its technical content before incorporating it into the final paper.

We take full responsibility for the final content of the paper, including all material produced with the assistance of generative AI.

### REPRODUCIBILITY STATEMENT

We provide the assumptions, derivations, and proofs underlying FACES in Appendix A. Detailed experimental specifications are provided for the Affine experiments in Appendix B, Visual $\mathrm{SE}(2)$ in Appendix C, EuRoC in Appendix D, and SparseWriter in Appendix E, including the task constructions, model architectures, training procedures, hyperparameters, data processing, and evaluation protocols needed to reproduce the reported results. Major experimental comparisons are evaluated across five random seeds using matched data and evaluation protocols. We also independently reran the major experiments and obtained the same qualitative conclusions. We will release the experimental code and configurations upon publication.

## REFERENCES

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. URL https://arxiv.org/abs/1607.06450.

Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5173–5182, 2017.

Dániel Baráth, Jana Noskova, Maksym Ivashechkin, and Jiří Matas. MAGSAC++, a fast, reliable and accurate robust estimator. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1301–1309, 2020. doi: 10.1109/CVPR42600.2020.00138.

Kavya Bhand and Aadi Joshi. Multi-horizon consistency as geometry: When latent dynamics contract, and when they do not. *arXiv preprint arXiv:2607.21645*, 2026.

Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The EuRoC micro aerial vehicle datasets. *The international journal of robotics research*, 35(10):1157–1163, 2016.

Lan Chen, Qi Mao, Yuchao Gu, and Mike Zheng Shou. Edit transfer: Learning image editing via vision in-context relations. *arXiv preprint arXiv:2503.13327*, 2025.

Liang CHEN, Li Shen, Yang Deng, Xiaoyan Zhao, Bin Liang, and Kam-Fai Wong. PEARL: Towards permutation-resilient LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=txoJvjfI9w.

Kyunghyun Cho, Bart Van Merriënboer, Çağlar Gulçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1724–1734, 2014.

Christopher B. Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3D-R2N2: A unified approach for single and multi-view 3d object reconstruction. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (eds.), *Computer Vision – ECCV 2016*, pp. 628–644, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46484-8.

Rocco De Nicola and Matthew CB Hennessy. Testing equivalences for processes. *Theoretical computer science*, 34(1-2):83–133, 1984.

Andrew Joseph Dudzik, Tamara von Glehn, Razvan Pascanu, and Petar Veličković. Asynchronous algorithmic alignment with cocycles. In *Proceedings of the Second Learning on Graphs Conference*, volume 231 of *Proceedings of Machine Learning Research*, pp. 3:1–3:17. PMLR, 2024.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning, 2017. URL https://arxiv.org/abs/1702.03118.

Norm Ferns, Prakash Panangaden, and Doina Precup. Bisimulation metrics for continuous Markov decision processes. *SIAM Journal on Computing*, 40(6):1662–1714, 2011.

Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-manifold preintegration for real-time visual-inertial odometry. *IEEE Trans. Robotics*, 33(1):1–21, 2017. doi: 10.1109/TRO.2016.2597321. URL https://doi.org/10.1109/TRO.2016.2597321.

Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G Bellemare. DeepMDP: Learning continuous latent space models for representation learning. In *International conference on machine learning*, pp. 2170–2179. PMLR, 2019.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*, 2024.

Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In William W. Cohen and Andrew W. Moore (eds.), *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, volume 148 of *ACM International Conference Proceeding Series*, pp. 369–376. ACM, 2006. doi: 10.1145/1143844.1143891. URL https://doi.org/10.1145/1143844.1143891.

Xingrui Gu. The propagation field: A geometric substrate theory of deep learning. *arXiv preprint arXiv:2605.08529*, 2026.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pp. 2555–2565. PMLR, 2019.

Ahmed Hefny, Zita Marinho, Wen Sun, Siddhartha Srinivasa, and Geoffrey Gordon. Recurrent predictive state policy networks. In *International Conference on Machine Learning*, pp. 1949–1958. PMLR, 2018.

Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units, 2017. URL https://openreview.net/forum?id=Bk0MRI5lg.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Youling Huang, Guanqiao Chen, Junchi Yao, Lu Wang, Fangkai Yang, Chao Du, ChenZhuo Zhao, Pu Zhao, Qingwei Lin, Saravan Rajmohan, et al. Beyond state consistency: Behavior consistency in text-based world models. *arXiv preprint arXiv:2604.13824*, 2026.

Hamza Keurti, Hsiao-Ru Pan, Michel Besserve, Benjamin F Grewe, and Bernhard Schölkopf. Homomorphism autoencoder–learning group structured representations from observed transitions. In *International Conference on Machine Learning*, pp. 16190–16215. PMLR, 2023.

Yash Kumar Lal, Nathanael Chambers, Raymond Mooney, and Niranjan Balasubramanian. TellMeWhy: A dataset for answering why-questions in narratives. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 596–610, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.53. URL https://aclanthology.org/2021.findings-acl.53/.

Kun Li, Longtao Hu, Jiajun Yu, Yida Xiong, Hongzhi Zhang, Jiameng Chen, Xiantao Cai, Jia Wu, and Wenbin Hu. PCEvo: Path-consistent molecular representation via virtual evolutionary. In Diego Calvanese (ed.), *Proceedings of the Thirty-Fifth International Joint Conference on Artificial Intelligence, IJCAI-26*, pp. 5558–5566. International Joint Conferences on Artificial Intelligence Organization, 8 2026a. doi: 10.24963/ijcai.2026/619. URL https://doi.org/10.24963/ijcai.2026/619. Main Track.

Quanquan Li, Hongbo Zhang, Yihe Chi, Jingyu Li, Xidong Xi, Liuyang Song, Hongzhen Zhang, Yuxiang Huang, Jing Ke, Siyuan Ma, et al. FutureBridge: Token selection beyond local preference in collaborative decoding. *arXiv preprint arXiv:2608.06819*, 2026b.

Michael Littman and Richard S Sutton. Predictive representations of state. *Advances in neural information processing systems*, 14, 2001.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

SiYuan Ma, Yiqin Luo, Zhangji, Canran Xiao, Albert Gao, Wei-Hsing Huang, Wei Wang, Qiwei Wu, Xinran Li, Jinfeng Wei, and Qixin Zhang. Hidden APIs in language models: Discovering reusable causal interfaces from forked futures, 2026. URL https://arxiv.org/abs/2607.27617.

Pablo Monteagudo-Lago, Arielle Rosinski, Andrew Joseph Dudzik, and Petar Veličković. Asynchrony invariance loss functions for graph neural networks. In Sharvaree Vadgama, Erik Bekkers, Alison Pouplin, Sekou-Oumar Kaba, Robin Walters, Hannah Lawrence, Tegan Emerson, Henry Kvinge, Jakub Tomczak, and Stephanie Jegelka (eds.), *Proceedings of the Geometry-grounded Representation Learning and Generative Modeling Workshop (GRaM)*, volume 251 of *Proceedings of Machine Learning Research*, pp. 65–77. PMLR, 29 Jul 2024. URL https://proceedings.mlr.press/v251/monteagudo-lago24a.html.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 5206–5210. IEEE, 2015.

Rahul Raguram, Ondrej Chum, Marc Pollefeys, Jiri Matas, and Jan-Michael Frahm. USAC: A universal framework for random sample consensus. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35 (8):2022–2038, 2013. doi: 10.1109/TPAMI.2012.257. URL https://doi.org/10.1109/TPAMI.2012.257.

Dima Rekesh, Nithin Rao Koluguri, Samuel Kriman, Somshubra Majumdar, Vahid Noroozi, He Huang, Oleksii Hrinchuk, Krishna Puvvada, Ankur Kumar, Jagadeesh Balam, et al. Fast conformer with linearly scalable attention for efficient speech recognition. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 1–8. IEEE, 2023.

Pedro Rodriguez, Paul A. Crook, Seungwhan Moon, and Zhiguang Wang. Information seeking in the spirit of learning: A dataset for conversational curiosity. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pp. 8153–8172. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.EMNLP-MAIN.655. URL https://doi.org/10.18653/v1/2020.emnlp-main.655.

Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. In *International Conference on Learning Representations*, 2021.

Lennon J Shikhman. Semigroup consistency as a diagnostic for learned physics simulators. *arXiv preprint arXiv:2605.26324*, 2026.

Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 8922–8931. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.00881. URL https://openaccess.thecvf.com/content/CVPR2021/html/Sun_LoFTR_Detector-Free_Local_Feature_Matching_With_Transformers_CVPR_2021_paper.html.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4149–4158. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1421. URL https://doi.org/10.18653/v1/n19-1421.

Zuojin Tang, Haoyun Liu, Xinyuan Chang, Changjie Wu, Dongjie Huo, Yandan Yang, Bin Liu, Zhejia Cai, Feng Xiong, Mu Xu, et al. ALAM: Algebraically consistent latent action model for vision-language-action models. *arXiv preprint arXiv:2605.10819*, 2026.

Petar Veličković, Adrià Puigdomènech Badia, David Budden, Razvan Pascanu, Andrea Banino, Misha Dashevskiy, Raia Hadsell, and Charles Blundell. The CLRS algorithmic reasoning benchmark. In *International Conference on Machine Learning*, pp. 22084–22102. PMLR, 2022.

Zijie Wang, Wei Zhang, Weiming Zhang, Fanqi Zhang, Xiao Tan, Yipeng Qin, and Guanbin Li. World models as group actions. *arXiv preprint arXiv:2605.24578*, 2026.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019. doi: 10.1162/tacl_a_00290. URL https://aclanthology.org/Q19-1040/.

Hangxing Wei, Xiaoyu Chen, Chuheng Zhang, Tim Pearce, Jianyu Chen, Alex Lamb, Li Zhao, and Jiang Bian. Learning additively compositional latent actions for embodied ai. *arXiv preprint arXiv:2604.03340*, 2026.

Yangfan Wu, Haozhe Wang, Huanyu Yang, Jianmin Ji, and Fangzhen Lin. SpecPath: Testing coding agents across contract-equivalent specification histories. *arXiv preprint arXiv:2608.09799*, 2026.

Kaijia Xu and Petar Veličković. Recurrent aggregators in neural algorithmic reasoning. In *The Third Learning on Graphs Conference*, 2024. URL https://openreview.net/forum?id=TcQZGWWEeg.

Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=-2FCwDKRREu.

Qimin Zhong, Hao Liao, Haiming Qin, Mingyang Zhou, Rui Mao, Wei Chen, and Naipeng Chao. Toward consistent world models with multi-token prediction and latent semantic enhancement. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13582–13602, 2026.

# A SHARED-FUTURE AGREEMENT: SETUP, PROOFS, AND EXTENSIONS

## A.1 SAMPLING DISTRIBUTIONS AND OBJECTIVES

Let $\mathcal{E}$ denote a distribution over triples $(h, p, q)$ for which the problem structure certifies

$$p \simeq q. \tag{13}$$

The equivalence relation is external to the learned model. Starting from the same initial state $h$, the two route states are $H_\theta(p; h)$ and $H_\theta(q; h)$.

For each depth $j$, let $\mathcal{U}_j$ denote the distribution of admissible shared continuations of exactly $j$ controls. In applications where the set of valid futures depends on the current example or route pair, this notation stands for the corresponding conditional distribution $\mathcal{U}_j(\cdot \mid h, p, q)$; we suppress the conditioning notation in the main paper. The same sampled continuation is always applied to both branches.

For $u = (u_1, \ldots, u_j)$, the per-sample exact-depth disagreement is

$$\ell_j(p, q; h, u) = d(B_u(H_\theta(p; h)), B_u(H_\theta(q; h))) \,. \tag{14}$$

Its expectation is

$$\mathcal{L}_j = \mathbb{E}_{(h,p,q)\sim\mathcal{E}} \mathbb{E}_{u\sim\mathcal{U}_j} \left[\ell_j(p, q; h, u)\right] . \tag{15}$$

Thus FACES-$j$ is simply $\mathcal{L}_j$, including $\mathcal{L}_{\text{FACES-0}} = \mathcal{L}_0$.

For nonnegative depth weights $\widetilde{\alpha}_j$ with $s = \sum_{j=0}^{K} \widetilde{\alpha}_j > 0$, we use normalized weights $\alpha_j = \widetilde{\alpha}_j / s$ and absorb $s$ into the auxiliary-loss coefficient. The resulting objective is $\mathcal{L}_{\text{FACES}}(\boldsymbol{\alpha}) = \sum_{j=0}^{K} \alpha_j \mathcal{L}_j$. Exact-depth FACES-$K$ uses $\alpha_j = \mathbb{1}[j = K]$, while uniform cumulative FACES-$\leq K$ uses $\alpha_j = 1/(K+1)$.

Throughout, the model is fixed and semantic equivalence $p \simeq q$ is preserved under common admissible futures. The local expansion applies to route states represented in a common Euclidean coordinate space. Vector norms are Euclidean, and derivative norms are the corresponding induced operator norms. For symmetric matrices, $A \preceq B$ means that $B - A$ is positive semidefinite; $\ker A$ denotes the null space of $A$.

## A.2 SHARED FUTURES AND PRESERVATION OF READOUT EQUIVALENCE

FACES uses the same continuation on both branches so that the comparison isolates route-dependent state differences. If the two branches instead received independent continuations $u$ and $v$ drawn from the same distribution, even identical entering states could produce nonzero expected disagreement purely because different futures induce different outputs:

$$\begin{aligned}
\mathbb{E}_{u,v}\|B_u(h) - B_v(h)\|^2 &= \mathbb{E}_u\|B_u(h)\|^2 + \mathbb{E}_v\|B_v(h)\|^2 - 2\mathbb{E}_{u,v}\left[B_u(h)^\top B_v(h)\right] \\
&= 2\mathbb{E}_u\|B_u(h)\|^2 - 2\|\mathbb{E}_u B_u(h)\|^2 \\
&= 2\mathbb{E}_u\|B_u(h) - \mathbb{E}_u B_u(h)\|^2 \\
&= 2\,\mathrm{tr}\,\mathrm{Cov}_u[B_u(h)].
\end{aligned} \tag{16}$$

The equalities follow by expanding the square, using that $u$ and $v$ are i.i.d., and applying the definition of covariance. Thus independent futures introduce a variance term unrelated to route dependence. Sharing the continuation removes this confound: when the entering states are identical, their behavioral disagreement is exactly zero for every shared future.

**Proposition 2 (Preservation of readout equivalence)** *On a forward-invariant reachable set with a common admissible action set, $Q(h) = Q(h')$ implies $B_u(h) = B_u(h')$ for every finite common future $u$ if and only if, for every action $a$, there exists a map $f_a$ such that*

$$Q \circ F_a = f_a \circ Q, \qquad F_a(h) = F_\theta(h, a). \tag{17}$$

**Proof of Proposition 2.** Suppose readout agreement is preserved under every finite common future. For $y$ in the image of $Q$, define $f_a(y) = Q(F_a(h))$ using any state with $Q(h) = y$. If two states $h, h'$ have the same readout $y$, preservation under the one-step future $a$ gives $Q(F_a(h)) = Q(F_a(h'))$. Thus either choice gives the same value of $f_a(y)$, so $f_a$ is well-defined and $Q \circ F_a = f_a \circ Q$.

Conversely, suppose these maps $f_a$ exist. For a common future $u = (a_1, \dots, a_k)$, repeated substitution gives

$$B_u(h) = Q(F_{a_k} \circ \cdots \circ F_{a_1}(h)) = f_{a_k} \circ \cdots \circ f_{a_1}(Q(h)). \tag{18}$$

Thus $Q(h) = Q(h')$ implies $B_u(h) = B_u(h')$ for every such future.

The following examples make the factorization condition concrete for affine dynamics and illustrate the stronger preservation of the full readout difference under additive dynamics.

For $Q(h) = Ch$ and $F_a(h) = A_a h + b_a$ on $\mathbb{R}^d$, the factorization condition is equivalent to $A_a \ker C \subseteq \ker C$ for every $a$: differences of two states with equal readout are exactly the vectors in $\ker C$. This condition means that state differences invisible to the readout remain invisible after each action.

In the additive case $F_a(h) = h + \psi(a)$, where $\psi(a)$ is a state increment depending only on the action, a shared future adds the same $\sum_{t=1}^{k} \psi(a_t)$ to both states, so $B_u(h) - B_u(h') = C(h - h')$. With this linear readout, additive transitions preserve the entire readout difference under shared continuations, so squared Euclidean disagreement remains constant across continuation depths.

### A.3 Local quadratic approximation and route differences

Fix $h$, a convex neighborhood containing $h$ and $h + \delta$, and a future law that is held fixed in state derivatives. Suppose $B_u$ is $C^2$ and $\|D_h B_u(x)\| \le G_j$, $\|D_h^2 B_u(x)\| \le M_j$ uniformly over that neighborhood and supported length-$j$ futures. Here $D_h^2 B_u(x)$ denotes the second derivative with respect to the hidden-state argument at $x$, viewed as a bilinear map.

**Proof of Proposition 1.** Taylor's formula gives

$$B_u(h + \delta) - B_u(h) = J_u(h)\delta + r_u(\delta), \tag{19}$$

$$r_u(\delta) = \int_0^1 (1 - t) D_h^2 B_u(h + t\delta)[\delta, \delta]\, dt. \tag{20}$$

The triangle inequality and the bilinear operator norm give

$$\begin{aligned} \|r_u(\delta)\| &\le \int_0^1 (1 - t) \|D_h^2 B_u(h + t\delta)[\delta, \delta]\|\, dt \\ &\le \int_0^1 (1 - t) \|D_h^2 B_u(h + t\delta)\|\, \|\delta\|^2\, dt \\ &\le M_j \|\delta\|^2 \int_0^1 (1 - t)\, dt = \tfrac{1}{2} M_j \|\delta\|^2. \end{aligned} \tag{21}$$

Similarly, the induced matrix norm and $J_u(h) = D_h B_u(h)$ give

$$\|J_u(h)\delta\| \le \|J_u(h)\|\, \|\delta\| = \|D_h B_u(h)\|\, \|\delta\| \le G_j \|\delta\|. \tag{22}$$

Expanding the squared norm gives

$$\|J_u\delta + r_u\|^2 = \delta^\top J_u^\top J_u \delta + 2\langle J_u\delta, r_u\rangle + \|r_u\|^2. \tag{23}$$

Cauchy-Schwarz and these bounds yield

$$\left| \|B_u(h + \delta) - B_u(h)\|^2 - \delta^\top J_u^\top J_u \delta \right| \le G_j M_j \|\delta\|^3 + \tfrac{1}{4} M_j^2 \|\delta\|^4. \tag{24}$$

The bound is uniform in $u$, so taking expectation preserves the cubic remainder:

$$\begin{aligned} \mathcal{J}_j^h(\delta) := \mathbb{E}_u \|B_u(h + \delta) - B_u(h)\|^2 &= \delta^\top \mathbb{E}_u[J_u^\top J_u]\delta + O(\|\delta\|^3) \\ &= \delta^\top \Gamma_j(h)\delta + O(\|\delta\|^3). \end{aligned} \tag{25}$$

Conditional independence of $u$ and $\delta$ given $h$ makes the same future law, and hence the same $\Gamma_j(h)$, apply to every sampled route difference. For $M_{\mathrm{route}}(h) = \mathbb{E}_\delta[\delta\delta^\top \mid h]$,

$$\begin{aligned}\mathbb{E}_\delta[\delta^\top \Gamma_j(h)\delta \mid h] &= \mathbb{E}_\delta[\mathrm{tr}(\Gamma_j(h)\delta\delta^\top) \mid h] \\ &= \mathrm{tr}(\Gamma_j(h)M_{\mathrm{route}}(h)).\end{aligned} \tag{26}$$

Averaging the uniform remainder over sufficiently small route differences gives $\mathbb{E}_\delta[\mathcal{J}_j^h(\delta) \mid h] = \mathrm{tr}(\Gamma_j(h)M_{\mathrm{route}}(h)) + O(\mathbb{E}_\delta[\|\delta\|^3 \mid h])$. Here $M_{\mathrm{route}}$ is an uncentered second-moment matrix.

We next interpret the quadratic approximation, connect it to route-dependent sampling, and examine its behavior under changes of hidden-state coordinates.

**Local geometry.** We examine the expected behavioral disagreement $\mathcal{J}_j^h(\delta)$ as a function of the state perturbation $\delta$, holding the reference state $h$, the model, and the future law fixed. At $\delta = 0$,

$$\mathcal{J}_j^h(0) = 0, \qquad \nabla_\delta \mathcal{J}_j^h(0) = 0, \qquad \nabla_\delta^2 \mathcal{J}_j^h(0) = 2\Gamma_j(h). \tag{27}$$

The loss has no linear term because it squares the readout difference, whose first-order term is $J_u(h)\delta$. For $\delta = \epsilon v$, the quadratic penalty is $\epsilon^2 v^\top \Gamma_j(h) v$, where $v^\top \Gamma_j(h) v = \mathbb{E}_u \|J_u(h)v\|^2 \geq 0$. Thus $\Gamma_j(h)$ is positive semidefinite, and its quadratic form measures the sensitivity of future readouts to the perturbation direction. If $v \in \ker \Gamma_j(h)$, then $J_u(h)v = 0$ almost surely: the first-order readout change vanishes, and the remainder bound gives $\mathcal{J}_j^h(\epsilon v) = O(\epsilon^4)$.

For comparison, hidden-state matching minimizes $\|(h+\delta) - h\|^2 = \|\delta\|^2$, whose Hessian with respect to $\delta$ is $2I$. It therefore penalizes all unit directions equally in the chosen coordinates, whereas FACES weights them according to their effect on future readouts.

**Conditional route sampling.** The expected training penalty depends jointly on the sampled route differences and the readout sensitivity under their corresponding future laws. If $\xi = (h_0, p, q)$ determines the future law, define

$$\Gamma_j(h_q; \xi) = \mathbb{E}_{u|\xi}[J_u(h_q)^\top J_u(h_q)], \qquad \mathcal{E}_j = \mathbb{E}_\xi[\delta_\xi^\top \Gamma_j(h_q;\xi)\delta_\xi], \quad \delta_\xi = h_p - h_q. \tag{28}$$

Then $\mathcal{L}_j = \mathcal{E}_j + O(\mathbb{E}_\xi \|\delta_\xi\|^3)$ under the uniform local bounds. In general, $\mathcal{E}_j$ cannot be replaced by $\mathrm{tr}(\mathbb{E}\Gamma_j\, \mathbb{E}[\delta\delta^\top])$. For one nonzero route difference, the Rayleigh quotient $\rho_j = \delta^\top \Gamma_j \delta / \|\delta\|^2$ measures the quadratic discrepancy per unit squared state difference.

For a route difference $\delta = \epsilon v$ with $\Gamma_0 v = 0$, the immediate output difference is $O(\epsilon^2)$ and its squared loss is $O(\epsilon^4)$. If $v^\top \Gamma_j v > 0$, the future loss instead has leading term $\epsilon^2 v^\top \Gamma_j v$. This is an additional local constraint. Its contribution to expected loss depends on the distribution of route differences. Matrices with the same null space can still assign different quadratic penalties.

**Change of coordinates.** A change of hidden-state coordinates re-expresses the same behavioral predictions. Let $\widetilde{h} = \varphi(h)$ be a diffeomorphism and $T = D\varphi(h)$ its Jacobian at the reference state. In the new coordinates, the behavioral map is $\widetilde{B}_u = B_u \circ \varphi^{-1}$, so $\widetilde{B}_u(\varphi(h)) = B_u(h)$. The chain rule and $D\varphi^{-1}(\varphi(h)) = T^{-1}$ give

$$\widetilde{J}_u(\widetilde{h}) = D_{\widetilde{h}} \widetilde{B}_u(\widetilde{h}) = J_u(h) T^{-1}. \tag{29}$$

Using the same future law and holding the reference state fixed, $T$ is independent of $u$. Hence the expected Gram matrix transforms as

$$\begin{aligned}\widetilde{\Gamma}_j(\widetilde{h}) &= \mathbb{E}_u[\widetilde{J}_u(\widetilde{h})^\top \widetilde{J}_u(\widetilde{h})] \\ &= \mathbb{E}_u[T^{-\top} J_u(h)^\top J_u(h) T^{-1}] \\ &= T^{-\top} \Gamma_j(h) T^{-1},\end{aligned} \tag{30}$$

where $T^{-\top} = (T^{-1})^\top$. A perturbation direction transforms as $\widetilde{v} = Tv$, and therefore

$$\widetilde{v}^\top \widetilde{\Gamma}_j \widetilde{v} = (Tv)^\top (T^{-\top} \Gamma_j T^{-1})(Tv) = v^\top \Gamma_j v. \tag{31}$$

Thus the quadratic penalty on a direction is independent of its coordinate representation.

For a finite displacement, $\widetilde{\delta} = \varphi(h+\delta) - \varphi(h) = T\delta + O(\|\delta\|^2)$, so $\widetilde{\delta}^\top \widetilde{\Gamma}_j \widetilde{\delta} = \delta^\top \Gamma_j \delta + O(\|\delta\|^3)$. The quadratic approximations therefore agree to second order, while the behavioral discrepancy itself is unchanged exactly. For a linear coordinate change, $\widetilde{\delta} = T\delta$ exactly and $\widetilde{M}_{\text{route}} = \mathbb{E}[T\delta\delta^\top T^\top] = TM_{\text{route}}T^\top$. Consequently, $\operatorname{tr}(\widetilde{\Gamma}_j \widetilde{M}_{\text{route}}) = \operatorname{tr}(\Gamma_j M_{\text{route}})$. Euclidean state distance generally changes because $\|\widetilde{\delta}\|^2 = \delta^\top T^\top T\delta$ for a linear change; the Rayleigh quotient also depends on this coordinate-dependent denominator.

### A.4 Geometry of depth mixtures

For a fixed model, using the same future distribution at each depth across mixtures, the weighted loss $\mathcal{L}_\alpha = \sum_j \alpha_j \mathcal{L}_j$ has quadratic-form matrix $\Gamma_\alpha = \sum_j \alpha_j \Gamma_j$. For $S = \{j : \alpha_j > 0\}$, positive semidefiniteness gives

$$\ker \Gamma_\alpha = \bigcap_{j \in S} \ker \Gamma_j. \tag{32}$$

$v^\top \Gamma_\alpha v$ is a sum of nonnegative terms with positive coefficients, so it vanishes exactly when every $v^\top \Gamma_j v$ vanishes for $j \in S$. Changing the positive weights can change the quadratic penalty, but its zero-penalty directions remain those unpenalized at every included depth.

For the EuRoC mixture $\pi = \text{Uniform}\{1, 2, 4\}$, the induced weights in Eq. 7 are $\alpha^{\text{MS}} = (31/90, 31/90, 8/45, 1/15, 1/15)$. Cumulative-4 assigns weight $1/5$ to each depth $0 \le j \le 4$. Both mixtures assign positive weights to depths $\{0, \dots, 4\}$, so $\ker \Gamma_{\text{MS}} = \ker \Gamma_{\le 4}$. Compared with cumulative-4, the multiscale mixture places greater weight on depths 0 and 1 and less on depths 2, 3, and 4.

### A.5 Propagation and contraction of state differences

Along a length-$j$ future, let $A_t = D_h F_\theta(h_{t-1}, u_t)$, $\Phi_u = A_j \cdots A_1$, and $C_u = D_h Q_\omega(h_j)$. The chain rule gives $J_u = C_u \Phi_u$. If $\|A_t\| \le \rho < 1$ and $\|C_u\| \le c$, submultiplicativity gives

$$\lambda_{\max}(\Gamma_j) \le c^2 \rho^{2j}. \tag{33}$$

For affine transitions and a linear readout, the quadratic discrepancy is exact and $\Gamma_j = \mathbb{E}_u[A_u^\top C^\top C A_u]$, with $A_u$ the product of the future linear transitions.

### A.6 Agreement, supervision, and anchoring

Write $\bar{y} = (y_p + y_q)/2$ and $d = (y_p - y_q)/2$. Expanding $\frac{1}{2}(\|\bar{y} - y^\star + d\|^2 + \|\bar{y} - y^\star - d\|^2)$ cancels the cross terms and proves 12. The nonnegative mean-error term gives $\|y_p - y_q\|^2 \le 2\|y_p - y^\star\|^2 + 2\|y_q - y^\star\|^2$. Thus squared-error supervision of both outputs already constrains their agreement. On the same supervised pairs, symmetric FACES increases the weight on disagreement, while anchoring changes its gradient routing. Additional route/future pairs extend agreement supervision to other outputs.

For square-integrable errors, the $L_2$ triangle inequality gives

$$\sqrt{\mathbb{E}\|y_S - y^\star\|^2} \le \sqrt{\mathbb{E}\|y_T - y^\star\|^2} + \sqrt{\mathbb{E}\|y_S - y_T\|^2}. \tag{34}$$

Therefore, for Anchored FACES, small student-reference disagreement guarantees small student error only when the reference error is also small.

## B Affine Program Analysis

### B.1 Affine task, data generation, and model

We use affine programs because their exact dynamics and equivalent routes are known. Each primitive $g_{\alpha,\beta}$ is an affine transformation

$$g_{\alpha,\beta}(x) = \alpha x + \beta, \qquad \alpha \sim U[0.8, 1.2], \qquad \beta \sim U[-0.5, 0.5]. \tag{35}$$

Table 4: Affine error for different learned action dimensions. Lower is better.

| Action dim. | Validation RMSE | L12 map RMSE | L96 map RMSE |
|---|---|---|---|
| 1 | 0.13194 | 0.26733 | 0.50595 |
| 2 | **0.00544** | **0.03475** | **0.23137** |
| 8 | 0.00671 | 0.03765 | 0.24365 |

Table 5: Length-96 range-matched map RMSE using the same frozen learned actions with different dynamics. Lower is better.

| Dynamics | L96 map RMSE |
|---|---|
| Learned GRU | 0.23137 |
| Exact dynamics with decoded primitives | **0.02815** |

For an accumulated map $G_t(x) = A_t x + B_t$, exact ordered composition obeys

$$\log A_t = \log A_{t-1} + \log \alpha_t, \qquad B_t = \alpha_t B_{t-1} + \beta_t. \tag{36}$$

We generate programs online by sampling every primitive independently from Eq. 35. The groundtruth target after every prefix is the accumulated coordinate pair $(\log A_t, B_t)$, and the task loss is the mean squared error over all prefixes and both coordinates.

The model first maps $(\log \alpha/0.2, \beta/0.5)$ through a fixed random $2 \to 256 \to 128$ embedding with $\tanh$ activations. A stateless controller MLP with dimensions $128 \to 128 \to d_a$ and a GELU activation (Hendrycks & Gimpel, 2017) produces the learned action. A GRUCell with a 128-dimensional hidden state processes the action sequence from a learned initial state. The readout consists of layer normalization followed by a $128 \to 128 \to 2$ MLP with a GELU activation and predicts $(\log A_t, B_t)$. The controller observes only the current primitive embedding, so information about the preceding program must persist through the recurrent state.

Two successive primitives and their exact composition provide the equivalent routes used by FACES. We sample pairs until the composed primitive also lies inside the primitive domain in Eq. 35. This keeps both routes within the controller's training domain.

### B.2 Action-information diagnostic

We first ask whether long-program error comes from missing information in the learned actions. For this diagnostic, we train action dimensions $d_a \in \{1, 2, 8\}$ on programs of length 2-6. We cycle through these lengths over 5,000 updates with batch size 512. We use AdamW with learning rate $10^{-3}$, weight decay $10^{-5}$, and gradient clipping at 5. Validation is performed every 250 updates, and we retain the checkpoint with the lowest validation RMSE.

One action dimension is not sufficient, while increasing the action from two to eight dimensions does not improve long-program execution. For the two-dimensional model, shuffling programs and transplanting actions between histories show that the actions represent reusable instructions for the current primitive rather than summaries of the whole program.

We then freeze the two-dimensional controller and train a nonlinear probe to predict $(\log \alpha, \beta)$ from its actions. The probe is a $2 \to 64 \to 64 \to 2$ MLP with SiLU activations (Elfwing et al., 2017). We train it for 5,000 updates with batch size 2,048 using AdamW (Loshchilov & Hutter, 2019) with learning rate $10^{-3}$ and weight decay $10^{-5}$. The probe uses 100,000 training primitives, 10,000 validation primitives, and 50,000 test primitives, all sampled independently.

To test how much of the length-96 error remains when these actions are used by the exact dynamics, we replace the GRU with Eq. 36 while keeping the learned actions fixed. The probe predicts the primitive parameters with RMSE $0.00111$ for $\alpha$ and $0.00245$ for $\beta$. Using the exact dynamics reduces length-96 map RMSE by $87.83\%$. This counterfactual does not decompose total error, but it shows that the learned actions retain information that supports much more accurate execution.

### B.3 COMMON CHECKPOINT AND CONTINUED TRAINING

We next test FACES from competent common checkpoints. For each seed in $\{42, 43, 44, 45, 46\}$, we initialize a new model with two-dimensional actions and add a primitive decoder. This decoder is a $2 \to 64 \to 64 \to 2$ MLP with SiLU activations that predicts $(\log \alpha, \beta)$ from each action. We train each common model for 7,000 updates on programs of length 2-12, cycling through the lengths with batch size 512. The common-model loss combines the prefix task loss with the normalized primitive-decoding loss weighted by $0.1$.

We use AdamW with learning rate $10^{-3}$, weight decay $10^{-5}$, and gradient clipping at $5$. We evaluate every 350 updates and select the checkpoint with the lowest validation prefix RMSE. This gives one independently trained common checkpoint per seed.

From each common checkpoint, we start Ordinary, FACES-0, FACES-0 after random history, and Continuation FACES. Every branch receives 3,500 additional updates with the same online task stream, batch size, optimizer, task loss, and primitive supervision. We evaluate every 175 updates and again select the checkpoint with the lowest validation prefix RMSE. Ordinary continues only the task and primitive losses. The other branches add the FACES loss with weight $1$.

For each FACES sample, we choose a receiver state from a random prefix of the current program and detach it. We then compare two sampled in-domain primitives applied sequentially with their direct composition. FACES-0 compares the two route readouts immediately. Continuation FACES samples $k$ uniformly from $\{1, 2, 4, 8, 16\}$ and applies the same $k$ newly sampled primitives to both routes before comparing their readouts. FACES-0 after random history first advances the receiver through a detached random prefix with a length drawn from the same set, then applies immediate agreement. This control tests whether visiting deeper recurrent states is sufficient to explain the benefit of continuation agreement.

The auxiliary loss uses detached controller actions and a frozen readout, so its gradient updates only the recurrent transition. The task and primitive losses continue to update the full model. Before computing agreement, we normalize $(\log A, B)$ coordinatewise by the empirical standard deviation of exact accumulated states over all prefixes of training-length programs.

### B.4 RANGE-MATCHED EVALUATION

We evaluate terminal map RMSE at lengths 12, 24, 48, and 96 using 4,096 newly sampled programs per length. To keep the accumulated affine maps within a controlled range, we retain a program only when every exact prefix satisfies $|\log A_t| \le 0.62$ and $|B_t| \le 2.0$. We use the same fixed bounds for all seeds and methods.

For each accepted program, we evaluate the predicted and exact terminal maps on 17 equally spaced inputs $x_j \in [-1, 1]$. For a batch of $B$ programs, the reported map RMSE is

$$E_{\mathrm{map}} = \left[ \frac{1}{17B} \sum_{b=1}^{B} \sum_{j=1}^{17} \left( (\widehat{A}_b x_j + \widehat{B}_b) - (A_b x_j + B_b) \right)^2 \right]^{1/2}. \tag{37}$$

This measures error in the executed affine function over the fixed input grid. Reported means and sample standard deviations are over the five training seeds.

### B.5 STATE-SPACE MAGNITUDE DOES NOT DETERMINE FUTURE EFFECT

We use three seed-42 checkpoints for the directional diagnostic: ordinary continued training, local quotient repair, and continuation agreement. They start from the same common checkpoint and use matched continued-training settings. Three disjoint held-out evaluation sets each contain 48 receiver states at every depth from 0 through 12. These sets vary the evaluation samples, not the training seed. Sequential application of two primitives and direct application of their composition produce $h_{\mathrm{seq}}$ and $h_{\mathrm{dir}}$; we define $h = h_{\mathrm{dir}}$ and $\delta = h_{\mathrm{seq}} - h$.

**Behavioral Jacobian and components.** Let $\widehat{Q} = S^{-1}Q$ be the decoded affine-map coordinates normalized by the diagnostic's fixed diagonal scale matrix $S$. Because $\log A$ and $B$ have different

Table 6: Length-96 range-matched affine map RMSE over five seeds. Lower is better.

| Method | Mean | Sample standard deviation |
|---|---|---|
| Ordinary | 0.10626 | 0.02804 |
| FACES-0 | 0.07724 | 0.00979 |
| FACES-0 after random history | 0.09888 | 0.03962 |
| Continuation FACES | **0.05794** | 0.01665 |

numerical scales, we normalize them before comparing route effects so that neither coordinate dominates the discrepancy solely because of its units. We sample 4,096 programs at each training length, collect the exact accumulated $(\log A_t, B_t)$ coordinates at every prefix, and compute the empirical standard deviation of each coordinate. This gives $S = \mathrm{diag}(0.253515, 0.639144)$. We compute $S$ once and keep it fixed across the diagnostic checkpoints, then use the normalized readout $\widehat{Q} = S^{-1}Q$. We stack the immediate readout and four sampled one-step probe readouts as $\Psi(h) = [\widehat{Q}(h); \widehat{Q}(F_{\tilde{a}_1}(h)); \ldots; \widehat{Q}(F_{\tilde{a}_4}(h))]$ and write $J_\Psi = D_h\Psi(h)$. We then form

$$\delta_\parallel = J_\Psi^\top (J_\Psi J_\Psi^\top + \lambda I)^{-1} J_\Psi \delta, \qquad \delta_{\mathrm{res}} = \delta - \delta_\parallel. \tag{38}$$

The positive ridge $\lambda$ is $10^{-4}$ times the mean diagonal of $J_\Psi J_\Psi^\top$, with a numerical floor. The aligned component lies in the regularized Jacobian row space. Because of the ridge, $\delta_{\mathrm{res}}$ is not an exact null-space component and the two components need not be orthogonal.

Local quotient repair samples normalized residual directions $v$ and penalizes the squared norm of the centered finite difference $[\Psi(F_a(h+\epsilon v)) - \Psi(F_a(h-\epsilon v))]/(2\epsilon)$, with $\epsilon = 10^{-2}$. It therefore targets transitions that turn locally weakly observable perturbations into observable differences. Continuation agreement instead penalizes disagreement between the actual equivalent routes after shared futures.

**Future effects and aggregation.** For a shared length-$H$ continuation $u$, define $e_H(v) = \widehat{Q}(H_\theta(u; h+v)) - \widehat{Q}(H_\theta(u; h))$. We evaluate $H \in \{0, 1, 2, 4, 8, 16\}$, keeping the decomposition fixed while extending the shared continuation. The full route-effect curve is the mean $\|e_H(\delta)\|_2$; it is a normalized disagreement norm, not task RMSE.

For $c \in \{\parallel, \mathrm{res}\}$, let $s_c = \langle \|\delta_c\|^2/\|\delta\|^2 \rangle$ and $q_c(H) = \langle \|e_H(\delta_c)\|^2/(\|e_H(\delta_\parallel)\|^2 + \|e_H(\delta_{\mathrm{res}})\|^2) \rangle$. Angle brackets denote equal-weight averaging over examples, receiver depths, and held-out evaluation sets. The plotted effect-density ratio is

$$R(H) = \frac{q_\parallel(H)/s_\parallel}{q_{\mathrm{res}}(H)/s_{\mathrm{res}}}. \tag{39}$$

This is a ratio of averaged energy fractions, not an average of per-example sensitivity quotients. Numerical denominator floors prevent division by zero. The two component effects are measured separately and need not sum to the full effect under nonlinear dynamics.

At $H = 16$, $R(H)$ is 44.9, 28.1, and 65.1 for ordinary continued training, local quotient repair, and continuation agreement, respectively. Their mean full route effects are 0.00993, 0.01730, and 0.00496. Because the decomposition is constructed from the immediate and one-step readouts, a strong short-horizon sensitivity contrast is expected. We therefore examine whether this contrast persists under longer held-out futures. The effect-density ratio measures how sensitivity is distributed across directions, while the full route effect measures overall route disagreement.

### B.6 Continuation training across five seeds

We finally test whether agreement under shared futures improves long-program execution. Table 6 reports the range-matched length-96 result for the four matched branches. Continuation FACES improves over ordinary training on all five seeds and over FACES-0 on four of five. Applying FACES-0 after a random history does not reproduce this improvement. Thus, the result is not explained simply by applying agreement at deeper recurrent states. Requiring equivalent routes to remain interchangeable after the same future computation provides an additional benefit in this setting.

Because we restrict the FACES gradient to the recurrent transition here, this experiment directly tests its effect on the recurrent model. The visual $\mathrm{SE}(2)$ experiment complements it by allowing the complete model to update under the auxiliary objective.

# C Visual $\mathrm{SE}(2)$ Experimental Details

This appendix records the construction used for the controlled visual experiment in Section 4.2.

## C.1 Task construction and visual demonstrations

Each semantic primitive $g \in \mathrm{SE}(2)$ has rotation $\theta \sim \mathrm{Uniform}[-0.14, 0.14]$ and translations $t_x, t_y \sim \mathrm{Uniform}[-0.04, 0.04]$. These parameters are used only by the generator and are not given to the learned controller. Instead, each primitive is demonstrated by a before/after RGB image pair of size $24 \times 24$. Image coordinates span $[-1, 1]^2$, with $(0, 0)$ at the center, $x$ increasing to the right, and $y$ increasing downward; for example, $(-1, 0)$ is the midpoint of the left edge. We form the before image by randomly rotating, scaling, and translating a fixed template of three colored noncollinear landmarks, then apply $g$ to obtain the after image. The template rotation is uniform over $[-\pi, \pi]$, its scale is uniform over $[0.7, 1.25]$, and each translation coordinate is uniform over $[-0.38, 0.38]$. The landmarks are rendered as Gaussian blobs with width $0.075$.

The before and after images are concatenated into a $6 \times 24 \times 24$ input. The visual pair contains no receiver scene or transformation history.

The inferred primitive is applied to a separate receiver containing 2-5 objects. Each occupied coordinate is sampled uniformly from $[-0.45, 0.45]^2$. The exact transformation sequence applies the same sequence of planar transformations to these coordinates and provides a groundtruth target after every transformation step.

## C.2 Model and information boundary

The controller receives only the current before/after image pair. It does not observe the receiver scene, previous visual pairs, previous actions, the recurrent state, the accumulated transformation, the current step index, or the total sequence length. We impose this information boundary so that the controller represents the current primitive, while the recurrent state carries the receiver and the previously executed transformation sequence. Any persistent effect of a route difference must therefore pass through the recurrent state.

The visual controller uses three $3 \times 3$ convolutional layers with channel widths $6 \to 32 \to 32 \to 64$, strides $1, 2, 2$, and GELU activations. Adaptive average pooling to $4 \times 4$ is followed by $1024 \to 64 \to 3$ fully connected layers with GELU, producing the learned action $a(g)$.

The receiver uses five padded object slots, each containing $[x, y, m]$, for a 15-dimensional input, where the binary mask $m \in \{0, 1\}$ indicates whether the slot is occupied. An occupied slot has $m = 1$, while a padded slot is $[0, 0, 0]$. A $15 \to 192 \to 192$ encoder with GELU produces the initial recurrent state $h_0$. The transition module is a GRUCell with three-dimensional input and 192-dimensional state (Cho et al., 2014); it receives no explicit time or sequence position. The readout is $\mathrm{LayerNorm}(192) \to 192 \to \mathrm{GELU} \to 10$ (Ba et al., 2016), reshaped into five two-dimensional coordinates.

Because receiver scenes contain a variable number of objects, ordinary task supervision uses the binary occupancy mask to exclude padded slots. For a training sequence of length $L$, the model predicts the receiver coordinates after every step $t = 1, \ldots, L$. We average the squared error over these steps, occupied objects, and both coordinate dimensions. Thus, the model is supervised after every primitive, not only at the terminal state.

## C.3 Sequence lengths and range-matched evaluation

Training cycles through sequence lengths 2-6, while evaluation uses lengths 6, 8, 12, 18, and 24.

Because a long random sequence can move the exact receiver far outside the region used for training, we report range-matched results for the main long-horizon comparisons. We retain a sequence only if

every occupied receiver coordinate at every exact prefix satisfies $|x| \leq 0.75$ and $|y| \leq 0.75$. We use rejection sampling until 4096 sequences are accepted at each evaluation length. For each seed, all methods are evaluated on the same accepted sequences.

### C.4 Exact route construction

We independently sample $g_1$ and $g_2$ from the primitive distribution and compute $g_{1,2} = g_2 \circ g_1$ in float64. We accept the pair only when the composed primitive also satisfies $\theta_{1,2} \in [-0.14, 0.14]$ and $t_{x,1,2}, t_{y,1,2} \in [-0.04, 0.04]$, keeping the direct route within the primitive distribution used during training. We then render separate demonstrations of $g_1$, $g_2$, and $g_{1,2}$.

During training, we select $h_r$ uniformly from the initial state or any reachable prefix state in the current ordinary-task minibatch. Starting from this same state, the sequential route is

$$h_{\text{seq}} = F_\theta\left(F_\theta(h_r, a(g_1)), a(g_2)\right), \tag{40}$$

while the direct route is

$$h_{\text{dir}} = F_\theta\left(h_r, a(g_{1,2})\right). \tag{41}$$

The two routes represent the same exact receiver transformation, but their learned actions and recurrent states may differ. For a shared continuation $u_{1:k}$, both route states receive exactly the same future primitives. The visual $\mathrm{SE}(2)$ FACES loss is the masked coordinate MSE between $Q_\omega(H_\theta(u_{1:k}; h_{\text{seq}}))$ and $Q_\omega(H_\theta(u_{1:k}; h_{\text{dir}}))$, averaged over occupied objects and both coordinates. FACES-0 uses $k = 0$, while FACES-4 uses $k = 4$.

### C.5 Main training conditions

For each seed, we first train the complete visual model using only the ordinary task objective for 5000 updates. We validate every 250 updates on 128 unseen sequences at each length from 2-6 and select the checkpoint with the lowest mean terminal RMSE across those lengths. All six methods, including the exact-target and additive-action controls, start from this common checkpoint and train for 2500 further updates. Ordinary continued training uses no auxiliary. FACES-0 compares the routes at their immediate endpoints, while FACES-4 applies four shared future primitives before comparing them. Both use auxiliary batches of 128 examples and update the controller, receiver encoder, recurrent transition, and readout end to end. Every branch retains the ordinary task loss.

Using the common checkpoint for seed 42, we calibrate the auxiliary coefficients on $B = 32$ batches. For the mixed-horizon calibration loss, each batch samples $K_b \sim \mathrm{Uniform}\{1, 2, 4, 8, 16\}$. Let $\boldsymbol{\vartheta}$ denote all trainable model parameters. We define

$$\begin{aligned} G_{\text{mix}} &= \mathrm{median}_{1 \leq b \leq B} \left\| \nabla_{\boldsymbol{\vartheta}} \mathcal{L}^{(K_b)}_{\text{route},b} \right\|_2, \\ G_k &= \mathrm{median}_{1 \leq b \leq B} \left\| \nabla_{\boldsymbol{\vartheta}} \mathcal{L}^{(k)}_{\text{route},b} \right\|_2, \qquad \eta_k = \frac{G_{\text{mix}}}{G_k}. \end{aligned} \tag{42}$$

These gradients come from the auxiliary loss alone. The measured scales are $G_{\text{mix}} = 9.72297 \times 10^{-4}$, $G_0 = 5.90117 \times 10^{-4}$, and $G_4 = 7.97788 \times 10^{-4}$, giving $\eta_0 = 1.64763$ and $\eta_4 = 1.21874$. We fix these coefficients across all five seeds before branch training. The exact-target controls and additive-action control are calibrated against the same $G_{\text{mix}}$, but each receives its own coefficient based on its median unweighted auxiliary gradient norm.

### C.6 Evaluation quantities

The main table reports three different questions rather than three versions of the same metric. All coordinate RMSE values average squared error over evaluation examples, occupied receiver objects, and both coordinate dimensions before taking the square root.

**L24 task RMSE.** This is the ordinary task evaluation on a range-matched sequence of length 24. The model executes the 24 visually specified transformations in sequence, and the predicted terminal receiver coordinates are compared with the exact terminal receiver coordinates. This is the primary measure of whether the model can execute a long sequence correctly.

Table 7: Range-matched terminal coordinate RMSE over five seeds. Values are means; lower is better.

| Method | L6 | L8 | L12 | L18 | L24 |
|---|---|---|---|---|---|
| Ordinary continued training | 0.00896 | 0.01187 | 0.02616 | 0.07147 | 0.14235 |
| FACES-0 | **0.00650** | **0.00795** | 0.01365 | 0.03385 | 0.07018 |
| FACES-4 | 0.00771 | 0.00970 | 0.01472 | **0.02340** | **0.03396** |
| Single-route exact target | 0.00742 | 0.00868 | 0.01242 | 0.02550 | 0.05163 |
| Dual-route exact target | 0.00720 | 0.00845 | **0.01199** | 0.02493 | 0.05064 |
| Additive-action control | 0.00860 | 0.01227 | 0.02928 | 0.08086 | 0.15609 |

**Suffix-16 route gap.** This diagnostic first executes a six-step receiver sequence, constructs $h_{\mathrm{seq}}$ and $h_{\mathrm{dir}}$ from an exact equivalent route pair, and then appends the same 16-step continuation to both states. The diagnostic is the RMSE between the two decoded receiver predictions after that common continuation:

$$G_{16} = \mathrm{RMSE}\left(Q_\omega(H_\theta(u_{1:16}; h_{\mathrm{seq}})), Q_\omega(H_\theta(u_{1:16}; h_{\mathrm{dir}}))\right). \tag{43}$$

It does not compare either prediction with the task target. Its purpose is to measure whether the two route states remain functionally interchangeable after substantial additional recurrent computation.

**Transition-only L24.** For this diagnostic, the controller, receiver encoder, and readout are restored from the common pretrained checkpoint used before the six training conditions split. Only the recurrent transition module from the training condition being evaluated is retained. The resulting hybrid model is then evaluated on the same L24 task. This asks whether the improvement remains when the other learned modules are returned to their common starting point.

### C.7 Optimization and evaluation protocol

We report five independent training seeds: $42$, $43$, $44$, $45$, and $46$. All runs use AdamW (Loshchilov & Hutter, 2019), batch size $256$, learning rate $10^{-3}$, weight decay $10^{-5}$, a constant learning rate, and gradient-norm clipping at $5$. Branches with an auxiliary objective use auxiliary batch size $128$. We validate every $125$ updates and select the checkpoint with the lowest mean terminal RMSE over lengths 2-6. Final evaluation uses the selected checkpoint. Neural computation uses float32, while exact transformations and targets use float64.

### C.8 Controls matched in continuation depth

FACES-4 differs from FACES-0 not only in what it compares, but also in the fact that its auxiliary gradient passes through four additional recurrent transitions. We therefore construct two controls that use exactly those same four future transitions but replace agreement between routes with exact task supervision.

For each example, the single-route control uniformly selects either the sequential or direct route and supervises it at the exact receiver state reached after the four future primitives. The dual-route control supervises both equivalent routes independently against the same exact target. Neither control contains a loss comparing one route with the other.

Using the same $G_{\mathrm{mix}}$ calibration gives $\eta_{\mathrm{single}} = 0.06607684$ and $\eta_{\mathrm{dual}} = 0.06549282$ at the common checkpoint of seed 42. We hold both coefficients fixed across all five seeds. These controls match continuation depth and use exact future targets, but the allocation of the auxiliary gradient across modules can differ.

The two controls are competitive at the shorter sequence lengths. Their disadvantage appears as the sequence grows, rather than as uniform superiority of FACES-4 at every length.

FACES-4 has lower L24 task error, Suffix-16 route gap, and Transition-only L24 error than both controls on all five seeds (Table 9).

These controls match the number of future recurrent transitions and provide the exact target at the same future state. They do not force the gradient allocation across modules to be identical to FACES-4.

Table 8: L24 task RMSE by training seed. SD is the sample standard deviation across the five seeds.

| Seed | Ordinary | FACES-0 | FACES-4 | Single target | Dual target | Additive |
|---|---|---|---|---|---|---|
| 42 | 0.12272 | 0.07364 | 0.03619 | 0.04905 | 0.04804 | 0.13365 |
| 43 | 0.17117 | 0.08121 | 0.03225 | 0.05308 | 0.05161 | 0.17412 |
| 44 | 0.14560 | 0.06418 | 0.03228 | 0.05174 | 0.05432 | 0.17191 |
| 45 | 0.13722 | 0.06147 | 0.03285 | 0.05207 | 0.04535 | 0.13584 |
| 46 | 0.13505 | 0.07040 | 0.03623 | 0.05222 | 0.05390 | 0.16492 |
| Mean | 0.14235 | 0.07018 | 0.03396 | 0.05163 | 0.05064 | 0.15609 |
| SD | 0.01807 | 0.00784 | 0.00207 | 0.00153 | 0.00387 | 0.01979 |

Table 9: Shared-future agreement alongside route-specific task RMSE and transition-only L24 RMSE. All entries are five-seed means. Lower is better.

| Method | Route gap | Suffix-16 Sequential task | Direct task | Transition-only L24 |
|---|---|---|---|---|
| Ordinary continued training | 0.01483 | 0.14289 | 0.13018 | 0.20643 |
| FACES-0 | 0.00806 | 0.07011 | 0.06325 | 0.18217 |
| **FACES-4** | **0.00362** | **0.03286** | **0.03183** | **0.10946** |
| Single-route exact target | 0.00672 | 0.05145 | 0.04651 | 0.13994 |
| Dual-route exact target | 0.00662 | 0.05042 | 0.04556 | 0.14303 |
| Additive-action control | 0.01524 | 0.15674 | 0.14360 | 0.27888 |

They therefore rule out the simpler explanation that four extra transitions or exact supervision at that future state are sufficient by themselves.

### C.9 Control on additive structure in the learned action

We also test whether the result can be reproduced by changing the learned action representation directly. The control is inspired by ALAM (Tang et al., 2026). For separately rendered demonstrations of $g_1$, $g_2$, $g_2 \circ g_1$, and $g_1^{-1}$, it uses

$$\mathcal{L}_{\mathrm{add}} = \|a(g_2 \circ g_1) - (a(g_1) + a(g_2))\|_2^2 \tag{44}$$

and

$$\mathcal{L}_{\mathrm{rev}} = \left\|a(g_1) + a(g_1^{-1})\right\|_2^2 . \tag{45}$$

The control uses the equal weight average of the two terms. Its auxiliary gradient updates only the visual controller, while the ordinary task loss continues to update the full model. It encourages an additive relation in the learned action space; vector addition is not the $\mathrm{SE}(2)$ group law.

Using the same $G_{\mathrm{mix}}$ calibration as FACES-4 gives $\eta_{\mathrm{alg}} = 0.0003897986$ at the common checkpoint of seed 42. We then fix this coefficient across all five seeds.

As a separate diagnostic, we feed $a(g_1) + a(g_2)$ to the recurrent transition as one action and compare the decoded receiver coordinates with the exact target for $g_2 \circ g_1$. For the additive-action control, the five-seed mean normalized additivity and reversal residuals are $0.01871$ and $0.01233$, and summed-action execution RMSE is $0.01039$. On the smaller-motion subset (the lowest half by maximum normalized primitive magnitude), these values are $0.01587$, $0.01311$, and $0.01018$. These are action-space diagnostics, not comparisons against Ordinary.

The additive control has higher mean L24 task RMSE ($0.15609$ versus $0.14235$), Suffix-16 route gap ($0.015239$ versus $0.014828$), and Transition-only L24 RMSE ($0.27888$ versus $0.20643$) than Ordinary. It is worse on four, three, and five of the five seeds, respectively. Thus, this regularizer does not reproduce the FACES-4 task gains.

### C.10 Checking the roles of the controller and recurrent state

We evaluate these checks on 4096 length-6 sequences for each of the five common pretrained models. The interpretation of this experiment relies on a simple division of roles. The controller should

describe the transformation shown in the current visual pair, while the recurrent state should carry the receiver configuration and the effect of the primitives that have already been applied. We therefore checked that the trained model actually behaves in this way.

We first test whether the learned action represents the current transformation independently of a particular receiver. An action inferred from one landmark demonstration can be used to transform a different receiver configuration. We also apply the same current primitive after unrelated preceding sequences. The inferred action remains usable in these different recurrent states, which provides evidence that it is not tied to one particular point in a sequence.

We next check that the recurrent computation is sensitive to the order of the actions. When a sequence is shuffled or reversed, the model prediction moves toward the exact target obtained by applying that new ordering, rather than remaining close to the target for the original sequence. This shows that the sequence of inferred primitives is actually being accumulated by the recurrent computation.

Finally, the controller signal is necessary for successful execution. Replacing the inferred actions with zeros, a constant vector, or actions from mismatched visual demonstrations degrades the prediction. As an additional diagnostic, we freeze the visual controller and fit a $3 \to 64 \to 64 \to 3$ probe with SiLU activations. The probe takes the controller's three-dimensional output action $a(g)$ as input and predicts the known rotation and translation parameters. We train the probe for 1500 AdamW updates with batch size 512, learning rate $10^{-3}$, and weight decay $10^{-5}$, then evaluate it on 4096 new primitives. Across the five common pretrained models, its mean RMSE is $0.00903$ radians for rotation, $0.00274$ for $x$-translation $t_x$, and $0.00270$ for $y$-translation $t_y$.

Together, these checks support the interpretation used in this benchmark: the controller tells the recurrent transition what transformation to apply now, while the recurrent state carries what has happened to the receiver so far. This separation is a property we check for this controlled experiment; it is not an assumption required by FACES in general.

# D EuRoC Experimental Details

**Data split.** We use `MH_01_easy`, `MH_02_easy`, `MH_03_medium`, `V1_01_easy`, `V1_02_medium`, and `V2_01_easy` for training; `V2_02_medium` and `V1_03_difficult` for validation; and `MH_04_difficult`, `MH_05_difficult`, and `V2_03_difficult` as the test set.

## D.1 Data construction and recurrent estimator

We first clip each EuRoC IMU sequence to the time span covered by its groundtruth trajectory. We divide the IMU stream into consecutive, nonoverlapping native intervals of approximately $50\,\mathrm{ms}$. We discard any native interval containing an unusually large gap between consecutive IMU measurements, defined as more than twice the sequence's median sampling interval. At each interval boundary, we obtain the groundtruth orientation by spherical linear interpolation and the groundtruth position and velocity by linear interpolation.

At each ordinary training update, we uniformly sample a training sequence, a window length $\ell \in \{4, 6, 8, 10, 12\}$ measured in native intervals, and a valid starting interval. We initialize the estimator with the exact groundtruth navigation state at the beginning of the window and a zero GRU hidden state. We supervise its prediction after every native interval in the sampled window.

Each transition receives a 25-dimensional vector containing $\log(\Delta R)$, $\Delta v$, $\Delta p$, $\Delta t$, the current rotation matrix, velocity, and position. An MLP maps this vector through $25 \to 48 \to 48$ layers with SiLU activations. A 96-dimensional GRUCell updates the persistent hidden state, and a residual head maps the concatenated 48-dimensional embedding and GRU state through $144 \to 64 \to 9$. The hidden layer uses SiLU. The final nine outputs are passed through $\tanh$ and scaled in three-dimensional blocks by $0.25$, $1.5$, and $0.75$ to produce the rotation, velocity, and position corrections, respectively. We initialize the final residual layer to zero so that training begins from nominal propagation.

The task loss averages normalized squared rotation, velocity, and position errors over every predicted interval. We use fixed scales $s_R = 0.104422$, $s_v = 1.165387$, and $s_p = 0.613556$ for every method.

Rotation remains part of the task and AFACES losses, while the reported metric defined below includes only position and velocity.

### D.2 Deterministic IMU preintegration and equivalent routes

We use a deterministic specialization of standard manifold IMU preintegration (Forster et al., 2017). We retain the nominal kinematic increments, but do not propagate their covariance or perform online bias correction. The nominal gyroscope and accelerometer biases are set to zero.

For IMU samples $(\omega_k, a_k)$ separated by $\delta t_k$, we initialize $\Delta R_0 = I$, $\Delta v_0 = 0$, $\Delta p_0 = 0$, and $\tau_0 = 0$, then compute

$$\Delta p_{k+1} = \Delta p_k + \Delta v_k \delta t_k + \tfrac{1}{2}\Delta R_k a_k \delta t_k^2, \tag{46}$$
$$\Delta v_{k+1} = \Delta v_k + \Delta R_k a_k \delta t_k, \tag{47}$$
$$\Delta R_{k+1} = \Delta R_k \operatorname{Exp}(\omega_k \delta t_k), \tag{48}$$
$$\tau_{k+1} = \tau_k + \delta t_k. \tag{49}$$

The resulting primitive is $g = (\Delta R, \Delta v, \Delta p, \Delta t)$, with $\Delta t = \tau_K$. Each native primitive summarizes approximately $50\,\mathrm{ms}$ of IMU measurements.

To show why adjacent primitives give equivalent routes, let $s = (R, v, p)$ be a navigation state and let $\gamma$ denote gravity. The nominal propagation used by the experiment is

$$\Phi_g(R, v, p) = \left(R\Delta R,\ v + \gamma\Delta t + R\Delta v,\ p + v\Delta t + \tfrac{1}{2}\gamma\Delta t^2 + R\Delta p\right). \tag{50}$$

For two consecutive primitives $g_1$ and $g_2$, we construct $g_{1,2} = g_2 \circ g_1$ as

$$\begin{aligned} \Delta R_{1,2} &= \Delta R_1 \Delta R_2, & \Delta v_{1,2} &= \Delta v_1 + \Delta R_1 \Delta v_2, \\ \Delta p_{1,2} &= \Delta p_1 + \Delta v_1 \Delta t_2 + \Delta R_1 \Delta p_2, & \Delta t_{1,2} &= \Delta t_1 + \Delta t_2. \end{aligned} \tag{51}$$

Substitution into Eq. 50 gives $\Phi_{g_2}(\Phi_{g_1}(s)) = \Phi_{g_{1,2}}(s)$. In particular, the gravity-dependent position terms combine because $\frac{1}{2}\Delta t_1^2 + \Delta t_1 \Delta t_2 + \frac{1}{2}\Delta t_2^2 = \frac{1}{2}(\Delta t_1 + \Delta t_2)^2$. Thus,

$$[g_1][g_2] \simeq [g_{1,2}]. \tag{52}$$

This equivalence is exact for the deterministic nominal propagation.

Because the native two-step route matches the factorization used during ordinary prefix-supervised training, we use it as the AFACES reference and treat the composed one-step route as the student. We then apply the same continuation of native increments to the two resulting recurrent states.

For $N$ evaluated trajectory windows, we compute position and velocity RMSE as

$$\mathrm{RMSE}_p = \left(\frac{1}{N}\sum_{n=1}^{N} \|\widehat{p}_n - p_n\|_2^2\right)^{1/2}, \qquad \mathrm{RMSE}_v = \left(\frac{1}{N}\sum_{n=1}^{N} \|\widehat{v}_n - v_n\|_2^2\right)^{1/2}. \tag{53}$$

The reported EuRoC task error is

$$E = \left(\frac{\mathrm{RMSE}_p}{s_p}\right)^2 + \left(\frac{\mathrm{RMSE}_v}{s_v}\right)^2, \tag{54}$$

where $s_p$ and $s_v$ are fixed normalization scales used for every method. All EuRoC results below use this same error definition.

### D.3 Using the shared future as a differentiable probe

We want the AFACES auxiliary to correct the composed step that produces the student state, while using the later recurrent computation to measure how that state affects future predictions. We therefore use the shared future as a differentiable probe rather than directly updating its transition parameters.

Let $z = (R, v, p, r)$ denote the complete estimator state, where $R$, $v$, and $p$ are the navigation variables and $r$ is the persistent GRU hidden state. If $z_{\mathrm{in}}$ is the state before the equivalent routes, we detach it as $\bar{z}_{\mathrm{in}} = \mathrm{sg}(z_{\mathrm{in}})$. The live composed transition produces the student endpoint

$$z_S^{(0)} = F_\theta(\bar{z}_{\mathrm{in}}, g_{1,2}), \tag{55}$$

while the native route $[g_1][g_2]$ supplies the detached reference. For the same continuation $u_{1:j}$, we evaluate the student future as

$$z_S^{(j)} = H_{\bar{\theta}}(u_{1:j}; z_S^{(0)}), \qquad \bar{\theta} = \mathrm{sg}(\theta). \tag{56}$$

The continuation remains differentiable with respect to its entering state. In particular,

$$\frac{dz_S^{(j)}}{d\theta} = D_z H_{\bar{\theta}}(u_{1:j}; z_S^{(0)}) \frac{\partial F_\theta(\bar{z}_{\mathrm{in}}, g_{1,2})}{\partial \theta}. \tag{57}$$

The AFACES gradient therefore passes through the future-state Jacobians to the composed transition that produced the student state, but it does not directly update the transition parameters at every future reuse. Increasing the continuation horizon changes the behavioral probe without automatically adding more direct parameter-gradient paths. The ordinary task loss continues to update the complete estimator.

### D.4 Training protocol and auxiliary calibration

We use seeds 42-46. For each seed, we first train the complete estimator with ordinary task supervision for 1200 updates and select a common checkpoint. Every ordinary, Fact-Aug, AFACES, and MS-AFACES branch then starts from that checkpoint and trains for 1200 additional updates. All runs use Adam with learning rate $3 \times 10^{-3}$, no weight decay, and gradient-norm clipping at 1. We validate every 100 updates using 16 uniformly spaced starts from each validation sequence at lengths $\{4, 6, 8, 10, 12\}$ and select the checkpoint with the lowest validation error at the native factorization.

Each continued-training update contains one ordinary task trajectory. AFACES branches also sample one route pair for the auxiliary loss. Their discrepancy uses fixed component weights $0.74300$, $0.97771$, and $1.27929$ for rotation, velocity, and position, respectively.

We calibrate every agreement coefficient once at the common checkpoint of seed 42 using 32 task and auxiliary examples. For an unweighted auxiliary $\mathcal{L}_A$, let $r_A = \mathrm{median}(\|\nabla \mathcal{L}_A\| / \|\nabla \mathcal{L}_{\mathrm{task}}\|)$. We set $\eta_A = 0.5/r_A$ and then keep the coefficient fixed across all seeds. For $k = 0, 1, 2, 4, 6, 8$, the resulting coefficients are $3.06917$, $3.03019$, $2.98343$, $2.86414$, $2.72081$, and $2.54684$, respectively. MS-AFACES uses $\eta_{\mathrm{MS}} = 2.95895$.

### D.5 Temporal factorization levels

Each evaluation window contains the same 24 consecutive native inertial intervals, each lasting approximately $50\,\mathrm{ms}$. The underlying physical trajectory is fixed; only the grouping of those intervals changes. At $c = 0$, all 24 intervals are processed separately. We then make the computation coarser by replacing adjacent pairs with their exact two-interval composition. At $c = 1$, every interval belongs to a pair, so the same trajectory is processed in 12 recurrent transitions instead of 24.

The parameter $c \in \{0, 0.25, 0.50, 0.75, 1\}$ is the fraction of the 24 native intervals that belong to two-interval groups. The resulting computation is:

| $c$ | two-interval groups | single intervals | recurrent transitions |
|---|---|---|---|
| 0 | 0 | 24 | 24 |
| 0.25 | 3 | 18 | 21 |
| 0.50 | 6 | 12 | 18 |
| 0.75 | 9 | 6 | 15 |
| 1 | 12 | 0 | 12 |

For example, at $c = 0.25$, six of the 24 intervals are arranged into three adjacent pairs. The other 18 remain separate, giving 21 recurrent transitions in total. For $c = 0.25$, $0.50$, and $0.75$, we evaluate 64 fixed placements of the two-interval groups and average over them.

We evaluate continuation horizons $k \in \{0, 1, 2, 4, 6, 8\}$ using seeds 42, 43, 44, 45, and 46. Replacing two adjacent intervals by their exact composition changes the nominal increment sequence only at floating-point precision: the maximum numerical discrepancy is below $1.3 \times 10^{-14}$.

Table 10: EuRoC validation error across temporal factorizations. Values are means over seeds 42, 43, 44, 45, and 46. Blue bold values are the lowest error in each column; red values are the highest.

| Method | $c = 0$ | $c = .25$ | $c = .50$ | $c = .75$ | $c = 1$ |
|---|---|---|---|---|---|
| Ordinary | 0.07443 | 0.06372 | 0.07660 | 0.11458 | 0.17783 |
| Fact-Aug | **0.07364** | **0.06251** | 0.06511 | 0.08227 | 0.11395 |
| AFACES-0 | 0.08894 | 0.07920 | 0.08061 | 0.09362 | 0.11831 |
| AFACES-$\leq 1$ | 0.08309 | 0.06815 | 0.06199 | 0.06488 | 0.07600 |
| AFACES-$\leq 2$ | 0.11411 | 0.09142 | 0.07852 | 0.07548 | 0.08213 |
| AFACES-$\leq 4$ | 0.11500 | 0.08103 | **0.06009** | **0.05184** | **0.05519** |
| AFACES-$\leq 6$ | 0.10321 | 0.08634 | 0.07592 | 0.07190 | 0.07406 |
| AFACES-$\leq 8$ | 0.08795 | 0.07827 | 0.07611 | 0.08180 | 0.09519 |
| MS-AFACES | 0.08032 | 0.06915 | 0.06398 | 0.06484 | 0.07203 |

### D.6 Factorization augmentation control

We introduce Factorization Augmentation (Fact-Aug) to test whether direct training on coarsened trajectories can explain the benefit of AFACES. At each continued-training update, we sample a native trajectory window and construct a second execution of the same window by composing selected adjacent increments. The fraction of increments included in these compositions is sampled uniformly from $\{0, .25, .50, .75, 1\}$. We apply direct groundtruth supervision after every transition in both executions and give the native and factorized losses equal weight. Fact-Aug therefore exposes the model to both native and composed increments, as AFACES does, but its additional training signal comes from groundtruth targets rather than route agreement. It uses the same common checkpoint, optimizer, 1200 continued-training updates, and checkpoint-selection rule as the other branches.

### D.7 Validation results across temporal factorizations

Table 10 reports validation error for the controls, fixed continuation horizons, and MS-AFACES at each factorization level $c$. Values are averaged over seeds 42, 43, 44, 45, and 46. Lower is better.

Among the fixed-horizon AFACES models, after averaging over the five seeds, the best $k$ changes with temporal factorization:

$$k^\star(c) = (1,\ 1,\ 4,\ 4,\ 4) \quad \text{for } c = (0, .25, .50, .75, 1). \tag{58}$$

The best $k$ also varies across individual seeds, so no single continuation horizon is best across all temporal factorizations.

The tradeoff is clear when comparing AFACES-$\leq 1$ and AFACES-$\leq 4$. At $c = 0$, their errors are $0.08309$ and $0.11500$, respectively. Averaged over $c = .75$ and $c = 1$, their errors are $0.07044$ and $0.05352$. A longer continuation helps when the trajectory is processed more coarsely, but hurts performance at the native temporal resolution in this comparison.

### D.8 Test results across temporal factorizations

We repeat the same evaluation on the three sequences in the EuRoC test set. Table 11 reports the controls, fixed continuation horizons, and MS-AFACES at each $c$, averaged over the five seeds.

Averaged over all five values of $c$, AFACES-$\leq 2$ is the best fixed continuation horizon, with error $0.07049$. Fact-Aug obtains $0.07902$: direct supervision on factorized inputs helps relative to ordinary training, but does not reproduce the best continuation-agreement result.

### D.9 MS-AFACES objective

MS-AFACES uses $k \in \{1, 2, 4\}$ and averages the corresponding objectives:

$$\mathcal{L}_{\text{MS}} = \frac{1}{3}\left(\mathcal{L}_{\text{AFACES-}\leq 1} + \mathcal{L}_{\text{AFACES-}\leq 2} + \mathcal{L}_{\text{AFACES-}\leq 4}\right). \tag{59}$$

Table 11: EuRoC test error across temporal factorizations. Values are means over seeds 42, 43, 44, 45, and 46. Blue bold values are the lowest error in each column; red values are the highest.

| Method | $c = 0$ | $c = .25$ | $c = .50$ | $c = .75$ | $c = 1$ |
|---|---|---|---|---|---|
| Ordinary | 0.09809 | 0.09096 | 0.10675 | 0.14607 | 0.20985 |
| Fact-Aug | 0.07686 | 0.06162 | 0.06258 | 0.07990 | 0.11416 |
| AFACES-0 | 0.10239 | 0.09091 | 0.09018 | 0.10029 | 0.12135 |
| AFACES-$\leq 1$ | 0.12785 | 0.10634 | 0.09415 | 0.09105 | 0.09620 |
| AFACES-$\leq 2$ | 0.08806 | 0.06852 | 0.05955 | 0.06139 | 0.07493 |
| AFACES-$\leq 4$ | 0.17973 | 0.13443 | 0.10262 | 0.08375 | 0.07726 |
| AFACES-$\leq 6$ | 0.13435 | 0.10967 | 0.09216 | 0.08166 | 0.07792 |
| AFACES-$\leq 8$ | 0.11489 | 0.09600 | 0.08574 | 0.08399 | 0.09086 |
| MS-AFACES | **0.07069** | **0.05393** | **0.04485** | **0.04393** | **0.05309** |

Table 12: Mean error across the five values of $c$ for MS-AFACES and the best fixed $k \in \{1, 2, 4\}$ selected separately for each seed on the EuRoC test set. Lower is better.

| Seed | MS-AFACES | Best fixed-$k$ error | Best $k$ | MS $-$ fixed |
|---|---|---|---|---|
| 42 | 0.04378 | 0.05187 | 4 | -0.00810 |
| 43 | 0.04385 | 0.04564 | 2 | -0.00179 |
| 44 | 0.08393 | 0.08891 | 2 | -0.00498 |
| 45 | 0.05093 | 0.07254 | 1 | -0.02161 |
| 46 | 0.04401 | 0.05330 | 1 | -0.00929 |

Expanding the three objectives gives

$$\mathcal{L}_{\mathrm{MS}} = \frac{31}{90}\left(\mathcal{L}_0^{\mathrm{A}} + \mathcal{L}_1^{\mathrm{A}}\right) + \frac{8}{45}\mathcal{L}_2^{\mathrm{A}} + \frac{1}{15}\left(\mathcal{L}_3^{\mathrm{A}} + \mathcal{L}_4^{\mathrm{A}}\right). \tag{60}$$

We use $\eta_{\mathrm{MS}} = 2.95895$ from the common calibration described above. Like the fixed-horizon branches, MS-AFACES starts from the common checkpoint, trains for 1200 updates, and selects its checkpoint using validation error at $c = 0$.

### D.10 MS-AFACES VALIDATION RESULTS

Averaged over the five values of $c$, MS-AFACES obtains error $0.07006$. Its error is $0.08032$ at $c = 0$ and $0.06843$ when averaged over $c = .75$ and $c = 1$. AFACES-$\leq 4$ is better at the two coarsest settings, with error $0.05352$, but has a much higher $c = 0$ error of $0.11500$.

Compared with AFACES-$\leq 1$, MS-AFACES has lower mean error across $c$ on three of five seeds. When the best value from $k \in \{1, 2, 4\}$ is selected separately for each seed, MS-AFACES is lower on two of five validation seeds. Thus MS-AFACES improves the average across temporal factorizations without being best for every seed and every factorization.

### D.11 MS-AFACES TEST RESULTS

Averaged over the five values of $c$, MS-AFACES obtains $0.05330$, compared with $0.07049$ for AFACES-$\leq 2$, the best fixed continuation horizon. The same advantage remains when $k$ is selected separately for each seed from $\{1, 2, 4\}$:

MS-AFACES is lower than the best value of $k \in \{1, 2, 4\}$ on all five seeds. It is also lower than AFACES-$\leq 2$ on all five paired seeds.

## E SPARSEWRITER: EXACT NONINVERTIBLE COMPUTATION

We construct SparseWriter to test whether continuation agreement improves over immediate agreement for discrete and noninvertible operations. The state is a binary memory $x \in \{0, 1\}^{64}$. Each

primitive $g = (m, b)$ overwrites exactly two distinct cells and is applied as

$$T_g(x) = (x \wedge m) \vee b, \qquad m \wedge b = 0, \tag{61}$$

where the retention mask $m$ is zero at the overwritten cells and $b$ contains their new values. An overwrite destroys the previous values and is therefore noninvertible. Two successive primitives have the exact composition

$$m_{21} = m_1 \wedge m_2, \qquad b_{21} = (b_1 \wedge m_2) \vee b_2. \tag{62}$$

The sequential route $(g_1, g_2)$ and the one-step route $g_{1,2} = g_2 \circ g_1$ therefore implement the same memory update.

All data are generated procedurally. Each initial memory bit is sampled independently with probability $1/2$. For every primitive, we uniformly sample two distinct memory locations and assign independently sampled binary values to them. At each training update, we generate a new batch and select its sequence length uniformly from $\{2, 3, 4, 5, 6\}$. The one-step composite examples used by $\mathcal{L}_{\text{direct}}$ are sampled independently from the sequence examples.

### E.1 Model and training protocol

A stateless controller maps the concatenated 64-bit retention mask and 64-bit write vector through $128 \to 128 \to 32$ to a learned action. A target encoder maps the initial memory through $64 \to 128 \to 128$, a 128-dimensional GRUCell processes the action sequence, and a $128 \to 128 \to 64$ readout predicts the memory-bit logits. The controller, encoder, and readout use SiLU activations.

Ordinary task training applies binary cross-entropy to the predicted memory after every operation. We also supervise sampled two-primitive compositions executed in one step, so the direct route used by FACES is itself trained. The task objective is $\mathcal{L}_{\text{task}} = \mathcal{L}_{\text{prefix}} + 0.5\,\mathcal{L}_{\text{direct}}$.

For each seed from 42-46, we first train a common model for 12000 AdamW updates, then initialize Ordinary, FACES-0, and FACES-4 from that checkpoint and train each branch for 2500 further updates. All branches retain the same task supervision. We use batch size 128, learning rate $3 \times 10^{-4}$, weight decay $10^{-5}$, and gradient-norm clipping at 5. We use the checkpoints after the final updates without validation-based checkpoint selection.

To construct an agreement example, we first apply 0-4 random operations to obtain a receiver state, then execute either two sampled primitives or their exact composition. FACES compares the predicted bit probabilities from the two routes using mean squared error. FACES-0 compares them immediately, while FACES-4 applies exactly four common future operations before the comparison. The auxiliary gradient updates the complete model. For each common checkpoint, we use 16 calibration batches and set each auxiliary coefficient so that the median per-batch auxiliary-to-task gradient-norm ratio is 0.10.

### E.2 Evaluation and results

Before training, we generate a fixed unseen evaluation panel containing 2048 operation sequences at each length in $\{6, 12, 24\}$ and 2048 route pairs evaluated after 0, 4, and 16 common future operations. Every method and training seed is evaluated on this same panel. Evaluation uses the same initial-memory and primitive distributions as training; lengths 12 and 24 test execution beyond the training lengths. Bit error rate thresholds each predicted probability at 0.5 and averages the incorrect-bit indicator over examples and all 64 cells. The route gap is the RMSE between the two routes' predicted bit probabilities after the common continuation.

Ordinary continued training obtains an L24 bit error rate of 0.42117. FACES-4 reduces the L24 error by 14.0% relative to FACES-0 and reduces the suffix-16 route gap by 15.1%. It improves all four quantities in Table 13 on every seed, and the L24 ordering FACES-4, FACES-0, Ordinary holds on all five seeds. At length 6, FACES-0 and FACES-4 have similar bit error rates of 0.07615 and 0.07587, respectively. The larger difference at length 24 is therefore a long-horizon effect in this experiment.

These results show that the observed continuation-agreement benefit is not confined to continuous or invertible transformations.

Table 13: SparseWriter results averaged over five independently trained common models. Lower is better.

| Method | L24 bit error | Suffix-16 gap | Suffix-16 bit error | |
|---|---|---|---|---|
| | | | Sequential | Direct |
| FACES-0 | 0.39679 | 0.03965 | 0.33624 | 0.32462 |
| FACES-4 | **0.34115** | **0.03369** | **0.28819** | **0.28182** |

Table 14: Qualification outcomes for candidate FACES constructions.

| Setting | Qualification finding |
|---|---|
| RNAR Heapsort (Xu & Veličković, 2024; Veličković et al., 2022) | Future computation naturally contracts route differences. |
| 3D-R2N2 (Choy et al., 2016) | Common new views naturally contract route differences. |
| Streaming speech recognition (Rekesh et al., 2023; Panayotov et al., 2015) | Little gain beyond aligned-prefix consistency and no WER improvement. |
| HPatches (Balntas et al., 2017) | Lower route discrepancy does not reliably lower task error. |
| In-context learning (CHEN et al., 2025) | Future route differences are visible, but task gains are inconsistent. |

# F Task Qualification and Negative Findings

We use these studies to qualify candidate constructions before treating them as primary experiments. For each candidate, we specify the equivalent histories, shared continuation, behavioral readout, and task metric. We then test whether route differences survive the continuation and whether reducing them improves task performance. For probabilistic readouts, we measure route discrepancy with Jensen-Shannon divergence (JSD); for coordinate readouts, we use root mean squared error (RMSE). Table 14 summarizes the evidence that led us to stop further investigation of each candidate construction.

These studies are intended as low-cost qualification tests rather than exhaustive evaluations of FACES on each task. We therefore limit experimentation to what is needed to decide whether a candidate construction warrants further study. A negative qualification result does not imply that FACES is ineffective for the underlying task: it may instead reflect the particular route construction, continuation, behavioral readout, optimization setup, or other choices made here. Accordingly, we report only the experimental details needed to explain each stopping decision rather than complete task-specific evaluations.

**Natural contraction in RNAR Heapsort and 3D-R2N2.** For recurrent neural algorithmic reasoning (RNAR) Heapsort (Xu & Veličković, 2024; Veličković et al., 2022), we reorder the same four top-level heap-building operations while leaving the remaining build and extraction steps unchanged. The behavioral readout is the model's soft pointer distribution. Across two frozen model seeds, route JSD is initially on the order of $10^{-2}$ but falls to the order of $10^{-6}$ after the common remainder of the algorithm.

For 3D-R2N2 (Choy et al., 2016), the two histories present the same six views in different orders, followed by the same six new views. We compare the voxel occupancy distributions after each route. Across 520 objects, median JSD falls from $1.204 \times 10^{-4}$ before the common views to $1.093 \times 10^{-5}$ afterward, a $90.9\%$ reduction. In both settings, the ordinary future computation already removes most route dependence, leaving little route defect for FACES to correct.

**Little continuation-specific supervision in speech recognition.** We use a released streaming FastConformer (Rekesh et al., 2023) on the LibriSpeech dev-other split (Panayotov et al., 2015). Two supported attention-context schedules process the same speech prefix. Both routes then use the same streaming context on the remaining audio, with their accumulated caches retained. The behavioral readout is the aligned CTC token distribution (Graves et al., 2006). We compare FACES-4 with aligned-prefix consistency, which directly matches aligned output distributions over the observed prefix.

In the seed-42 qualification run, predictive JSD after the common future is $0.224688$ for FACES-4 and $0.225990$ for aligned-prefix consistency, only a $0.58\%$ reduction. Word error rate is $8.619\%$ for FACES-4, compared with $8.372\%$ under ordinary training. The continuation objective therefore adds little beyond the simpler consistency control in this construction and does not improve the task metric.

**Agreement without reliable correctness gains on HPatches.** We use a frozen LoFTR-outdoor frontend (Sun et al., 2021) and USAC-MAGSAC homography fitting (Raguram et al., 2013; Baráth et al., 2020). Sequential and composed homography routes act on the same eight-point receiver, then receive the same four projective transformations. Route discrepancy is the RMSE between their predicted receiver coordinates, while task error is RMSE against the exact transformed coordinates.

FACES-4 reduces route discrepancy across the controlled future conditions, but after separately tuning the weight of the auxiliary loss for FACES-0 and FACES-4, FACES-4 no longer has a consistent task-error advantage. In a separate experiment with visually constructed approximate equivalences, the route gap falls from $0.06788$ to $0.05790$, while the mean task-error benefit, defined as FACES-0 error minus FACES-4 error, is $-0.000157$. Agreement improves, but correctness does not improve reliably in these constructions.

**Visible continuation effects but mixed in-context learning gains.** We follow the four-task demonstration-permutation setting used by PEARL (CHEN et al., 2025), using Meta-Llama-3-8B (Grattafiori et al., 2024) with a frozen base model and LoRA adapters (Hu et al., 2022). The same four demonstrations are presented in different orders, and the query is their common continuation. We measure full-vocabulary next-token JSD as the query is processed. Across 128 examples, median route JSD rises from $0.008845$ immediately before the query to a maximum of $0.016864$ during the query. The median per-example ratio between maximum future and immediate JSD is $2.409$, so the continuation exposes route differences that are weak at the endpoint.

This diagnostic does not lead to consistent task gains. Relative to Pair-ERM, which uses the same paired route examples with only the ordinary task loss, MS-FACES changes worst-permutation performance by $+3.125$ percentage points on CoLA (Warstadt et al., 2019), $+6.250$ on CommonsenseQA (Talmor et al., 2019), $+4.5466$ on Curiosity Dialogue (Rodriguez et al., 2020), and $-6.9527$ on Tell Me Why (Lal et al., 2021). The pooled worst-permutation change is $+1.7422$ percentage points, while mean performance across permutations decreases by $1.6881$ points. We therefore do not use this construction as primary evidence for FACES.