TEMPOROUTE: CONTROL-RELEVANCE ROUTING OVER TEMPORAL SKIPS AND RECURRENT DEPTH IN WORLD-MODEL PLANNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Fixed-step world-model planners spend one latent transition on every imagined environment step, even when action sensitivity is concentrated around brief contacts. TempoRoute turns this mismatch into joint routing over temporal coverage and recurrent depth: at each imagined node, uncertainty growth, critic sensitivity, and latent context select an aggregate `skip`, a one-step `standard`, or a recurrent-depth `burst`. A physical-time actor–critic scores these operators under a common 15-step horizon and 15-query cap while receding-horizon execution retains one primitive action per observation. Across six event-heavy DMControl and ManiSkill2 tasks, TempoRoute reduces mean planner latency by 12.6% (95% CI 11.0–14.1%) and GFLOP by 27.3% (95% CI 25.9–28.6%) relative to the matched Uniform-Time planner, with 5/6 task-level quality-equivalence decisions. A histogram-matched random placement attains nearly the same cost but only 2/6 equivalence decisions, showing that where computation is placed matters beyond operator frequency. At the reported means, TempoRoute is 12.2–20.8% faster than each task's strongest-quality fixed-step comparator, with a maximum task-level mean-quality shortfall of 2.4%; complete-loop median latency falls by 10.8% on A100 and 13.9% on Jetson AGX Orin. The router also supplies its own applicability signal: smooth tasks use `standard` on 94.7% and 96.1% of nodes, and cue corruption at $\sigma = 0.20$ contracts the three-task latency reduction from 15.4% to 1.7% [0.5, 2.9]%. These results establish node-wise time-and-depth allocation as a practical route to faster event-heavy planning without lowering the environment-facing control rate.

1 INTRODUCTION

One Walker-Run route trace places its largest burst-probability peaks near stance contacts (Figure 5). Across evaluation, TempoRoute assigns 33% of imagined nodes to multi-step skips, 63% to ordinary one-step updates, and 4% to recurrent-depth bursts. The useful computation pattern is therefore asymmetric: aggregate predictable flight intervals, preserve ordinary dynamics across most of the rollout, and spend extra recurrent updates near brief phases where the value landscape changes rapidly.

A fixed-step imagination model cannot express that pattern. One transition per environment step spends the same model depth everywhere; globally coarsening the rollout can erase contact structure, whereas globally increasing depth repeats expensive updates during stable phases. State-dependent skip predictors (Neitz et al., 2018) and adaptive-depth world models (Lu et al., 2026; Sivasankar, 2026) address one side of this choice. Planning, however, requires a joint decision over physical coverage and recurrent computation while preserving comparable action queries, lookahead, and execution cadence.

TempoRoute makes that joint decision inside each candidate imagination. A shared selective-SSM body supports three planner-visible operators: `skip` predicts a policy-marginal aggregate endpoint over multiple environment steps, `standard` advances one step with one body application, and `burst` uses multiple recurrent applications to predict one one-step endpoint. Uncertainty growth and critic sensitivity, interpreted in latent context, route among them. The actor–critic conditions on

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.

accumulated physical time, the training objective couples aggregate and one-step predictions, and the first body application used to form routing cues is reused by the selected operator. Every candidate shares a 15-step physical horizon and a 15-query cap; execution still applies only the winning root action, observes one environment step, and replans.

The resulting frontier shift is consistent across two task families. Relative to the same-backbone Uniform-Time control, TempoRoute lowers planner latency by 12.6%, actor-query nodes by 31.4%, and GFLOP by 27.3%, with 5/6 task-level equivalence decisions. Replacing learned locations by samples from the same task-level operator-and-factor histogram preserves almost all arithmetic saving but only 2/6 equivalence decisions. Smooth dynamics push the learned routes back toward `standard`, cue corruption progressively removes the cost advantage, and full-loop timing retains 10.8% and 13.9% median reductions on A100 and Jetson AGX Orin. Together these tests connect the gain to selective placement rather than to a globally shorter rollout.

TempoRoute adapts a different object than neighboring control mechanisms. Options and action repeat change executed action persistence (Sutton et al., 1999; Precup, 2000; Bacon et al., 2017; Vezhnevets et al., 2017; Kipf et al., 2019); adaptive ODE and horizon methods vary numerical work or rollout depth (Chen et al., 2018; Rubanova et al., 2019; Georgiev et al., 2024); adaptive replanning changes when an external plan is refreshed (Cheng et al., 2026); and multiscale or event-aware models organize representations through duration conditioning, fixed branches, skills, or learned hierarchies (Nhu et al., 2025; Kumar et al., 2023; Freed et al., 2023; Gumbsch et al., 2024; Wang et al., 2026; Peng et al., 2026; Zhang et al., 2026). TempoRoute couples temporal skipping and recurrent depth at the planner node, then exposes actor queries, body applications, physical coverage, GFLOP, and complete measured latency under one execution interface.

Contributions.

1. We introduce a joint time-and-depth imagination operator: a control-relevance router selects policy-marginal skip, one-step standard, or recurrent-depth burst updates, with duration-aware return scoring and cross-operator temporal consistency.
2. We define a planner comparison interface that holds physical coverage, actor-query capacity, backbone size, and primitive execution cadence fixed while separating routed nodes, body applications, GFLOP, planner latency, and complete-loop latency.
3. We identify the value of learned placement through a six-task matched-backbone matrix and cost-matched route controls, then test the implied selection rule with operator and signal ablations, alternate targets, zero-hint initialization, smooth dynamics, cue corruption, operating-point sweeps, and two hardware regimes.

2 Time-and-Depth Routing under Shared Planner Caps

A recurrent state-space model represents the POMDP belief by $x_t = (\mathbf{h}_t, \mathbf{z}_t)$, with posterior $q_\theta(\mathbf{z}_t \mid \mathbf{h}_t, \mathbf{o}_t)$ and imagined prior $p_\theta(\mathbf{z}_{t+1} \mid \mathbf{h}_{t+1})$ (Hafner et al., 2019). TempoRoute assigns each imagined node an operator label, covered duration d_k in environment-step units, and logged operator-body evaluation count e_k:

$$\begin{aligned} \texttt{skip}(n): &\quad (d_k, e_k) = (n, 1), \\ \texttt{standard}: &\quad (d_k, e_k) = (1, 1), \\ \texttt{burst}(m): &\quad (d_k, e_k) = (1, m), \end{aligned} \tag{1}$$

where $n \in \{2, \dots, N_{\max}\}$ and $m \in \{2, \dots, M_{\max}\}$. Skip predicts an aggregate endpoint n environment steps ahead from one body application. Standard predicts a one-step endpoint from one application. Burst also predicts a one-step endpoint, but composes the shared body m times as adaptive recurrent depth. Its intermediate iterates are computation states: they neither advance the physical-time counter nor represent certified $\Delta t_{\text{env}}/m$ samples.

The three operators share one semi-Markov scoring rule (Bradtke & Duff, 1994; Sutton et al., 1999). For a replay segment of covered duration d, define its discounted integrated reward and continuation target as

$$R_t^{(d)} = \sum_{j=0}^{d-1} \gamma^j r_{t+j}, \qquad C_t^{(d)} = \prod_{j=0}^{d-1} c_{t+j}. \tag{2}$$

A reward-rate head $\hat{\rho}_k$ is supervised by $R_t^{(d_k)}/d_k$, while the continuation and elapsed-time heads predict $C_t^{(d_k)}$ and d_k. The operator-level critic target is

$$y_k = \hat{\rho}_k d_k + \gamma^{d_k} \hat{c}_k V_\psi(x_{k+1}, \tau_k + d_k). \tag{3}$$

For skip, the target segment begins with the conditioned root action and follows the continuation actions present in replay. Writing that replay behavior as μ, skip learns the policy-marginal endpoint and reward under μ after the root action; it is not a counterfactual claim that the root action repeats for n executed steps. For burst, $d = 1$ and equal-rate supervision partitions the same observed one-step reward across the m recurrent updates, with continuation applied at the endpoint. Terminal padding contributes zero reward and continuation.

Every routed node consumes one actor query, hence $Q = R$, while $E = \sum_k e_k$ records body applications. The common feasible set is

$$\mathcal{F} = \{(Q, T) : Q \le B = 15,\ T = \textstyle\sum_k d_k \le H_{\text{phys}} = 15\}. \tag{4}$$

Planning stops at the first active boundary. Thus "shared budget" means a common feasible cap, not equal realized Q or E; the latter are measured outcomes that define the compute–quality frontier. GFLOP and latency include routing-cue computation and the complete candidate batch. After candidate ranking, the controller executes only the winning root action for one environment step and replans from the next observation.

3 Method

3.1 Architecture Overview

TempoRoute couples a shared selective-SSM transition, a control-relevance router, three operator bodies, and a physical-time actor–critic. Figure 1 shows the data path and the separate counters used at deployment.

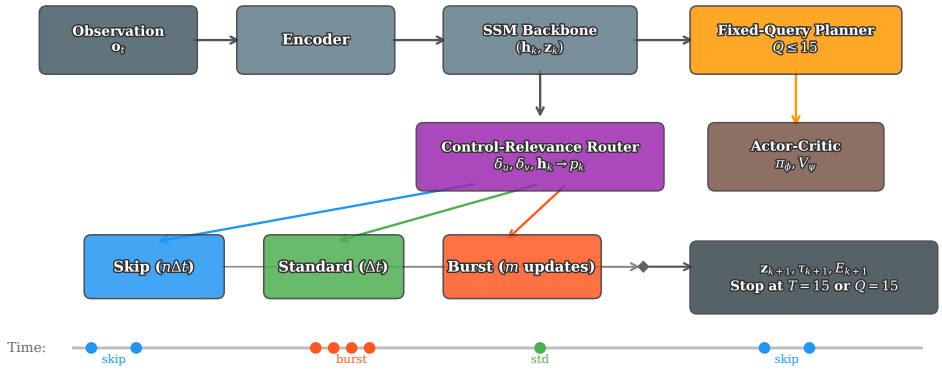

Figure 1: TempoRoute planning interface. The control-relevance router selects a multi-step aggregate, one-step update, or one-step recurrent-depth operator. The candidate body application used to compute routing cues is reused as the selected operator's first application. Each node consumes one actor query; E counts all body applications, while GFLOP and latency include cue computation and batching. Physical coverage terminates at 15 steps, and receding-horizon execution emits one primitive action after each observation.

All routes share the observation encoder, selective-SSM backbone (Gu et al., 2022; Smith et al., 2023; Gu & Dao, 2024), stochastic latent, reward and continuation heads, actor, and critic. Given $x_k = (\mathbf{h}_k, \mathbf{z}_k)$ and action $\mathbf{a}_k$, the backbone produces one candidate transition:

$$(\tilde{\mathbf{h}}_{k+1}, \mu_{k+1}, \sigma_{k+1}) = \text{SSM}_\theta(\mathbf{h}_k, \mathbf{z}_k, \mathbf{a}_k), \qquad \tilde{\mathbf{z}}_{k+1} \sim \mathcal{N}(\mu_{k+1}, \text{diag}[\sigma_{k+1}^2]). \tag{5}$$

3.2 CONTROL-RELEVANCE ROUTING HEAD

The candidate distribution from Eq. (5) supplies

$$\delta_u(x_k, \mathbf{a}_k) = \|\sigma_{k+1} - \sigma_k\|_2, \tag{6}$$

$$\delta_v(x_k, \mathbf{a}_k, \tau_k) = \left| V_\psi(x_k, \tau_k) - V_\psi(\tilde{\mathbf{h}}_{k+1}, \tilde{\mathbf{z}}_{k+1}, \tau_k + 1) \right|. \tag{7}$$

The first term measures predictive-scale growth and the second measures critic sensitivity across the sampled candidate transition. A small MLP maps these cues and the latent state to route probabilities:

$$p_k = \mathrm{softmax}(W_2\, \mathrm{ReLU}(W_1[\delta_u; \delta_v; \mathbf{h}_k] + b_1) + b_2)\,. \tag{8}$$

Training uses straight-through Gumbel–Softmax for the three-way route (Jang et al., 2017; Maddison et al., 2017); inference uses argmax after masking infeasible choices. Two categorical factor heads map the same router representation to q_n over $\{2, \ldots, N_{\max}\}$ and q_m over $\{2, \ldots, M_{\max}\}$. Only the selected branch's factor is instantiated, and route–factor pairs exceeding the residual physical horizon are masked before normalization. Straight-through samples carry endpoint-model and imagined-return gradients to both route and active factor logits; the discrete d_k and e_k counters themselves receive no gradient. Observation-derived velocity spikes and simulator contact flags warm-start early optimization and decay to zero before evaluation; zero-hint training reaches the same final region with slower early learning (Appendix Table 8).

3.3 VARIABLE-TIME TRANSITION OPERATORS

Standard retains the sampled candidate transition from Eq. (5). Skip injects a duration embedding into the candidate's deterministic path and evaluates the shared stochastic prior at the transformed state:

$$\mathbf{h}^{\mathrm{skip}}_{k+1} = \tilde{\mathbf{h}}_{k+1} + g_\theta(e(n_k \Delta t_{\mathrm{env}})), \qquad \mathbf{z}^{\mathrm{skip}}_{k+1} \sim p_\theta(\mathbf{z} \mid \mathbf{h}^{\mathrm{skip}}_{k+1}), \tag{9}$$

where g_θ is a learned residual network and $e(\cdot)$ is sinusoidal. A training tuple for skip is $(x_t, a_t, n, x_{t+n}, R^{(n)}_t, C^{(n)}_t)$; intervening actions $a_{t+1:t+n-1}$ come from the recorded continuation policy and are marginalized rather than passed as inputs. Temporal consistency aligns the direct aggregate with a standard rollout driven by that same recorded action sequence:

$$\mathcal{L}_{\mathrm{consist}} = \mathbb{E}\Big[D_{\mathrm{KL}}\Big(p^{\mathrm{skip}}_\theta(\mathbf{z}_{k+n} \mid \mathbf{z}_k, \mathbf{a}_k, n; \mu) \,\|\, p^{\mathrm{seq}}_\theta(\mathbf{z}_{k+n} \mid \mathbf{z}_k, \mathbf{a}_{k:k+n-1}) \Big) \Big]. \tag{10}$$

This defines skip as a policy-marginal semi-Markov model after the root action, not as execution-time action repeat. Burst is adaptive recurrent depth:

$$u^{(1)}_k = \tilde{x}_{k+1}, \qquad u^{(j)}_k = F_\theta(u^{(j-1)}_k, a_k)\ (j = 2, \ldots, m_k), \qquad x^{\mathrm{burst}}_{k+1} = u^{(m_k)}_k. \tag{11}$$

Here F_θ is the shared transition body and $\tilde{x}_{k+1}$ is the already-computed routing candidate. Therefore burst incurs m_k, not $m_k + 1$, body applications. Only the final iterate is matched to the recorded next posterior; intermediate iterates are latent computation states at the same operator time. The observed one-step reward is divided equally across their auxiliary reward losses, intermediate continuation targets equal one, and the final continuation equals the recorded target. Terminal-lumped and endpoint-only alternatives preserve the reward integral and produce the same qualitative allocation result (Appendix Table 9).

3.4 TRAINING OBJECTIVE AND ROUTING GRADIENT

Let $\mathcal{L}_{\mathrm{belief}}$ be the standard RSSM prediction objective with posterior–prior regularization. The reported operator-model objective is

$$\mathcal{L}_{\mathrm{model}} = \mathcal{L}_{\mathrm{belief}} + \mathcal{L}_{\mathrm{reward}} + \mathcal{L}_{\mathrm{cont}} + \alpha \mathcal{L}_{\mathrm{time}} + \beta \mathcal{L}_{\mathrm{consist}}, \tag{12}$$

with pixel reconstruction on DMControl, state regression elsewhere, reward-rate supervision from Eq. (2), continuation prediction, elapsed-time prediction, $\alpha = 1.0$, and $\beta = 0.1$. The critic minimizes squared error to Eq. (3), and the actor maximizes the corresponding imagined operator-level return. Straight-through route samples carry both endpoint-model and actor–critic gradients to the route and factor logits. No latency, GFLOP, compute, or entropy term rewards cheap routes directly: skip is retained when an aggregate endpoint preserves prediction and return targets, whereas burst is retained when additional recurrent depth improves those targets. The resulting cost reduction is therefore measured rather than inserted into the optimization objective.

3.5 Shooting Planner and Accounting

At deployment, the actor–critic conditions on cumulative elapsed time:

$$\mathbf{a}_k \sim \pi_\phi(\mathbf{a}_k \mid \mathbf{h}_k, \mathbf{z}_k, \tau_k), \quad V_\psi(\mathbf{h}_k, \mathbf{z}_k, \tau_k). \tag{13}$$

Let P denote the candidate-batch size, held fixed within every matched task comparison. The planner draws one action per active candidate and node from π_ϕ in a single device batch. It samples the next latent in Eq. (5), computes routing cues from that sample, and reuses the sample as standard, skip's base state, or burst's first recurrent iterate. Routes are recomputed from each candidate's own latent state. If candidate i ends at node K_i, its operator-level score is

$$G_{i,K_i} = V_\psi(x_{i,K_i}, \tau_{i,K_i}), \qquad G_{i,k} = \hat{\rho}_{i,k} d_{i,k} + \gamma^{d_{i,k}} \hat{c}_{i,k} G_{i,k+1}. \tag{14}$$

The root action of $\arg\max_i G_{i,0}$ is executed for one primitive environment step; the next observation updates the posterior before the shooting procedure repeats. Invalid route–factor pairs, including skips longer than the residual horizon, are masked before sampling or argmax.

We report Q (actor queries), R (routed nodes, equal to Q), E (body or ODE-solver applications), and T (physical steps covered). Table counts are per candidate; full-decision GFLOP and latency include all P candidates, route-cue heads, factor heads, scoring, and synchronization. Each candidate stops at $Q_i = 15$ or $T_i = 15$, whichever occurs first. A skip can yield $Q_i = R_i < 15$; a burst can yield $E_i > R_i$; and Adaptive ODE can satisfy $Q_i \leq 15$ while using more than 15 solver evaluations. Algorithm 2 gives the complete decision-time procedure.

4 Temporal Adaptation in World-Model Planning

State-dependent time and depth. Adaptive Skip Intervals (ASI) learns situation-dependent prediction intervals in recurrent simulators (Neitz et al., 2018); TAWM conditions transitions on elapsed time (Nhu et al., 2025); and MTS3, THICK, and SF-RSSM organize dynamics through multiple temporal levels or slow/fast components (Kumar et al., 2023; Gumbsch et al., 2024; Wang et al., 2026). ASI is the closest precedent for learned skipping, but it optimizes predictive interval selection rather than action-ranked planner trajectories under shared query and execution semantics. On the depth axis, LoopWM adaptively repeats a parameter-shared block (Lu et al., 2026), while Sivasankar (2026) show that the benefit of predictor depth can reverse across rollout configurations. TempoRoute joins these axes in one node-wise action-conditioned choice and tests placement with histogram-matched controls.

Planning and execution cadence. Fixed-step latent planners from World Models and PlaNet through DreamerV3, TD-MPC2, and MuZero couple one model transition to each imagined environment step (Ha & Schmidhuber, 2018; Hafner et al., 2019; 2020; 2021; Sekar et al., 2020; Hansen et al., 2022; Hafner et al., 2025; Hansen et al., 2024; Schrittwieser et al., 2020). Adaptive ODE and horizon methods allocate solver evaluations or rollout depth (Chen et al., 2018; Rubanova et al., 2019; Georgiev et al., 2024); AdaReP instead adapts when a cached plan is refreshed (Cheng et al., 2026). Options and action repeat change executed action persistence (Sutton et al., 1999; Precup, 2000; Bacon et al., 2017; Vezhnevets et al., 2017; Kipf et al., 2019). TempoRoute leaves replanning and primitive-action cadence unchanged and routes computation only inside the current imagination.

Event structure. Temporally abstract, event-aware, and hierarchical world models learn skills, segments, or abstraction levels (Freed et al., 2023; Peng et al., 2026; Zhang et al., 2026; Hansen et al., 2025); event-triggered control allocates executed controller updates (Heemels et al., 2012; Åström, 2008). TempoRoute does not require event labels at deployment: model uncertainty, critic sensitivity, and latent context choose planner-visible operators, while zero-hint training tests independence from privileged warm starts. Its selective-SSM backbone (Gu et al., 2022; Smith et al., 2023; Gu & Dao, 2024; Krinner et al., 2025) is the shared implementation substrate rather than the novelty claim. Appendix Table 16 compares these adapted objects directly.

5 Experiments

The evaluation follows the operator path from allocation to executed decision. First, the full six-task matched-backbone matrix tests whether routing changes the compute–quality frontier. Placement

and operator controls then separate route location from route frequency. Smooth dynamics and cue corruption probe when the router returns to uniform computation, and complete-loop timing tests whether planner savings survive controller and environment overhead. All experiments use five seeds and report mean $\pm$ standard deviation. DMControl uses 50 episodes per seed; ManiSkill2 and smooth control use 100. Training uses 8×A100 80GB GPUs, 1M environment steps, a 10M-transition replay buffer, Adam (Kingma & Ba, 2015) at 3×10^{-4}, and batches of 16 length-64 sequences. Models have approximately 14M parameters. At deployment, $B = 15$, $H_{\text{phys}} = 15$, $N_{\max} = 4$, and $M_{\max} = 3$.

We predeclare a 2% relative-return equivalence margin for DMControl and a two-percentage-point success margin for ManiSkill2, representing the smallest quality changes treated as consequential at each benchmark's reported scale. Decisions use paired seed-level differences under the task's margin; bootstrap intervals resample paired seed identities. The six-task aggregates are unweighted task means. Each task is a separate estimation target, so we report all six intervals and make no family-wise equivalence claim. The headline 2.4% is PegInsertion's relative shortfall, whereas its criterion is two absolute success points. Planner timing uses FP32, batch size one, a fixed warm-up, CUDA synchronization, and 10000 decisions per task and seed. Planner-only timings omit environment stepping; Section 5.6 measures the full loop.

5.1 Baselines

The identification control is Uniform-Time: it keeps TempoRoute's selective-SSM backbone, elapsed-time actor–critic, optimizer, replay, training steps, physical horizon, query cap, and candidate batching, but selects `standard` at every node. The histogram-matched control replaces the learned decision at evaluation: for each task and trained seed, it samples the complete $(\text{operator}, n, m)$ choice from that run's empirical joint route histogram, preserving factor frequencies as well as the operator mix. No model parameter is retrained; paired seeds, candidate actions, and stopping rules remain unchanged. Appendix Table 13 adds uncertainty-only, value-only, skip-disabled, burst-disabled, and fixed-cycle routers under the same interface.

Four ecological comparators—fixed-step RSSM, DreamerV3 (Hafner et al., 2025), TD-MPC2 (Hansen et al., 2024), and adaptive latent ODE (Rubanova et al., 2019)—retain their model-specific objectives, proposal rules, and transition parameterizations while sharing the 1M-step training budget, approximately 14M parameters, $Q \leq 15$, and 15-step coverage. Their rows measure competitiveness rather than isolate routing; causal attribution comes from Uniform-Time and the route controls. Appendix Table 6 details the common cap and method-specific compute unit.

5.2 Matched-Backbone Identification

Table 1: Full-suite routed-operator identification matrix. Uniform-Time (U) and TempoRoute (T) share the selective-SSM stack; T additionally routes among skip/standard/burst operators. Quality is return for DMControl and success percentage for ManiSkill2, reported as mean $\pm$ standard deviation over five seeds; latency and GFLOP are U/T means. Q is TempoRoute actor-query nodes, E its logged operator-body applications, and S/St/B its routed-node percentage. Every rollout targets the same 15-step physical horizon under the same query cap; latency and GFLOP include routing cues.

Task	U quality	T quality	U/T ms	Q	E	U/T GFLOP	S/St/B
Walker	878±23	882±21	4.83/4.20	10.8	11.3	79.1/58.8	33/63/4
Hopper	333±34	335±33	4.76/4.10	10.2	10.9	78.6/56.6	38/56/6
Cheetah	834±22	837±20	4.71/4.30	11.4	11.8	78.8/60.5	29/68/3
PickCube	89.7±3.0	90.4±2.6	4.95/4.30	9.8	10.4	80.2/54.7	43/52/5
StackCube	72.4±4.3	73.8±4.0	4.89/4.20	9.4	10.8	79.7/56.0	39/47/14
PegInsert	65.6±5.0	66.2±4.8	5.02/4.40	10.1	11.5	80.0/59.9	31/56/13

Table 1 identifies the routed-operator effect across the complete event-heavy suite. TempoRoute reduces mean planner latency by 12.6% (seed-bootstrap 95% CI 11.0–14.1%), actor-query nodes by 31.4%, and GFLOP by 27.3% (95% CI 25.9–28.6%). Paired differences meet task-level equivalence on 5/6 tasks, with PegInsertion inconclusive. The task-specific mixtures rule out one global stride:

skip ranges from 29% to 43%, while manipulation assigns up to 14% of nodes to recurrent depth. Appendix Figure 5 adds a trajectory-level contact-alignment check.

Table 2: Matched placement test across all six event-heavy tasks. At evaluation, the random route replaces each learned choice with an independent sample from the same seed and task's joint operator–factor histogram. Compute remains nearly equal while locations change; full controls appear in Appendix Table 13.

Router	Quality equiv.	Latency reduction	GFLOP reduction	Mean Q
TempoRoute joint cues	5/6	12.6%	27.3%	10.3
Random matched histogram	2/6	11.8%	25.9%	10.4

Table 2 is the direct placement test: the two routers differ by 0.8 latency points, 1.4 GFLOP points, and 0.1 mean queries, yet learned routing retains three additional quality-equivalent tasks. Seeded operator and signal ablations in Figure 2 and Appendix Table 13 reinforce the same placement result.

5.3 Ecological Comparisons

DMControl locomotion (Tassa et al., 2020) and ManiSkill2 manipulation (Gu et al., 2023) provide periodic ground contacts and discrete grasp, release, and insertion events. Tables 3–4 place TempoRoute beside the strongest-quality fixed-step comparator; Appendix Table 17 reports every comparator and Q/E value.

Table 3: DMControl, DreamerV3/TempoRoute.

Task	D/T return	D/T ms
Walker	891/882	5.1/4.2
Hopper	341/335	5.0/4.1
Cheetah	842/837	4.9/4.3

Table 4: ManiSkill2, DreamerV3/TempoRoute.

Task	D/T success	D/T ms
PickCube	91.2/90.4	5.2/4.3
StackCube	74.6/73.8	5.3/4.2
PegInsert	67.8/66.2	5.4/4.4

At the reported means, TempoRoute stays inside the predeclared quality margin against DreamerV3 on all six tasks while lowering planner latency by 0.6–1.1 ms. It uses 10.2–11.4 queries on locomotion and 9.4–10.1 on manipulation. PegInsertion is the tightest quality case (1.6 absolute success points, or 2.4% relative); the matched-seed comparison with Uniform-Time remains inconclusive there. The route mixtures and Appendix Figure 5 connect these frontier shifts to skips across predictable phases and concentrated burst probability near contacts.

5.4 Smooth-Control Boundary

Reacher-Easy and Pendulum-Swingup test the prediction that low temporal heterogeneity should route almost all decisions through `standard`. TempoRoute selects `standard` on 94.7% and 96.1% of their nodes, respectively. Its reported returns are 964 versus DreamerV3's 968 on Reacher and 856 versus 862 on Pendulum, while latency changes by less than 5% relative to fixed-step RSSM. The full Q/E comparison in Appendix Table 11 shows that both physical coverage and recurrent depth converge toward the uniform operating point.

5.5 Routing Mechanism and Applicability

We ablate TempoRoute's routing rule on Walker-Run and StackCube while holding the backbone, training budget, physical horizon, and actor-query cap fixed (Figure 2).

The seeded error bars separate learned joint routing from random allocation. Single-operator and consistency ablations reveal complementary task structure. The staged zero-hint runs in Appendix Table 8 retain fixed-step pretraining and delayed router introduction while removing heuristic overrides; Tables 9 and 13 extend the mechanism test to reward-target shape and signal-specific routers.

We next perturb both standardized router cues at planning time. Figure 3 reports latency reduction and its seed-aware confidence band relative to fixed-step RSSM.

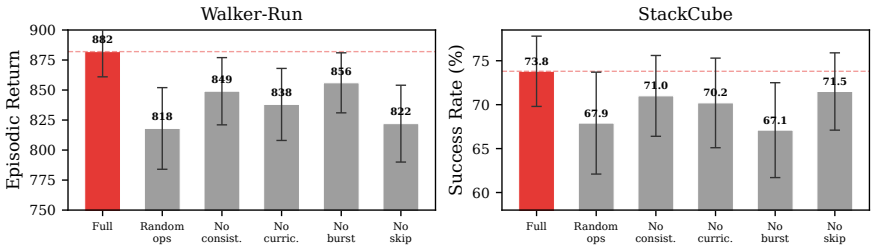

Figure 2: Routing-rule ablation on Walker-Run (left) and StackCube (right), mean ± standard deviation over five seeds. Random operators lose the most quality; removing temporal consistency or the staged curriculum also weakens the result. "No curriculum" activates routing from initialization, unlike the staged zero-hint condition in Appendix Table 8. Skip is especially valuable for locomotion coverage, while burst contributes more strongly to contact-rich manipulation.

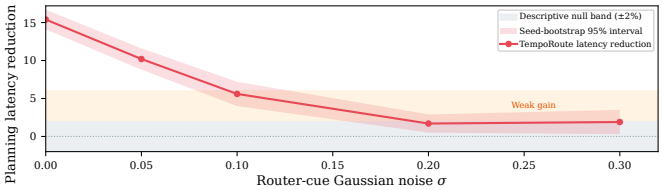

Figure 3: Held-out cue corruption on Walker-Run, StackCube, and PegInsertion over five seeds. Gaussian noise with standard deviation σ is applied independently to standardized uncertainty-growth and value-sensitivity inputs; the line averages task-level latency reductions and the band is a seed bootstrap 95% interval. The gray $\pm 2\%$ region is descriptive. Appendix Table 15 reports the corresponding quality response.

The point estimate contracts from 15.4% on this three-task subset to 1.7% at $\sigma = 0.20$, where the mean first enters the descriptive $\pm 2\%$ region; the confidence interval is [0.5, 2.9]%. The subset-specific clean value is distinct from the six-task 12.6% headline. Because latent context remains unperturbed, this experiment establishes that the two named cues materially affect routing, while the smooth rows show that the learned policy can also recognize a near-uniform regime.

5.6 End-to-End Decision Latency

Table 5: End-to-end decision latency pooled across Walker-Run, StackCube, and PegInsertion. Timings include observation transfer, encoding, planning, actor selection, simulator step, and synchronization. Each row aggregates 150000 decisions (10000 per task and seed). "Quality eq." is the count of these three tasks meeting the matched quality criterion.

Hardware	Method	Median	p90	p99	Plan share	Quality eq.
A100 80GB	Uniform-Time	6.21	7.04	8.31	78%	3/3
A100 80GB	TempoRoute	5.54	6.26	7.39	75%	3/3
AGX Orin	Uniform-Time	23.8	27.4	32.9	86%	3/3
AGX Orin	TempoRoute	20.5	23.8	28.6	83%	3/3

Planner-side savings remain decision-relevant after simulator and controller overhead are included for policies from the matched-backbone evaluation. Median latency improves by 10.8% on A100 and 13.9% on Jetson AGX Orin; p99 follows the same direction. The edge-device result shows that the allocation advantage survives a materially different compute regime, while the smaller A100 end-to-end percentage reflects fixed environment overhead.

6 Discussion

TempoRoute's gains arise from an asymmetric division of labor. Skip removes unit transitions during predictable phases, while the smaller burst share invests recurrent depth where uncertainty and critic sensitivity support it. Four observations connect that mechanism to the frontier shift: cost-matched random placement loses three additional equivalence decisions; disabling skip or burst hurts different task families; a Walker trace localizes burst probability around stance contacts; and corrupting the deployed cues contracts the three-task latency advantage to 1.7%. The complete-loop measurements then show that this planner-side allocation remains visible after controller and environment costs.

The evidence establishes this result for event-heavy simulated continuous control with a selective-SSM backbone, a 15-step lookahead, three operator families, and model-derived cues. Skip represents replay-policy-marginal continuation after the root action, and burst represents recurrent computation rather than physical substeps; these semantics make policy shift, longer horizons, other backbones, and richer route sets distinct operating regimes. Within the evaluated regime, smooth-task self-selection and cue corruption provide concrete diagnostics for when routing should approach the one-step default.

7 Conclusion

TempoRoute makes temporal coverage and recurrent depth jointly selectable at each imagined node. The shared horizon/query interface exposes a clear consequence: across six event-heavy tasks, the learned routes reduce mean planner latency by 12.6% and GFLOP by 27.3% with 5/6 matched quality-equivalence decisions, whereas histogram-matched random locations retain only 2/6. Complete-loop median latency also falls by 10.8% on A100 and 13.9% on Jetson AGX Orin. Allocating model computation by control relevance can therefore move the planning frontier without slowing the primitive control loop.

Reproducibility Statement

Equations (1)–(14), Algorithms 1–2, and Section 5 specify the reported model, operator targets, stopping rule, accounting, hardware, timing, seeds, and evaluation margins. The appendices provide router provenance, complete matched-family comparisons, ablations, sensitivity grids, evaluation denominators, and accounting definitions for every reported table and figure.

AI Use Statement

Generative AI tools assisted with manuscript preparation, including prose editing, formatting and consistency checks, and bibliographic reference verification. This statement reflects the documented manuscript-preparation record.

References

Karl Johan Åström. Event based control. In Alessandro Astolfi and Lorenzo Marconi (eds.), *Analysis and Design of Nonlinear Control Systems*, pp. 127–147. Springer, 2008. doi: 10.1007/978-3-540-74358-3_9.

Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, pp. 1726–1734, 2017. doi: 10.1609/aaai.v31i1.10916.

Steven J. Bradtke and Michael O. Duff. Reinforcement learning methods for continuous-time Markov decision problems. In *Advances in Neural Information Processing Systems 7*, volume 7, pp. 393–400. MIT Press, 1994. URL https://proceedings.neurips.cc/paper/1994/hash/07871915a8107172b3b5dc15a6574ad3-Abstract.html.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, pp. 6571–6583, 2018.

Yutian Cheng, Xiaojian Ma, Xianhao Wang, Min Yang, Rongpeng Su, Hangxin Liu, Xi Chen, Shuai Li, and Qing Li. AdaReP: Adaptive re-planning under model mismatch for neural world-model predictive control. *arXiv preprint arXiv:2606.23079*, 2026. doi: 10.48550/arXiv.2606.23079. URL https://arxiv.org/abs/2606.23079. Accepted at ICANN 2026.

Benjamin Freed, Siddarth Venkatraman, Guillaume Adrien Sartoretti, Jeff Schneider, and Howie Choset. Learning temporally abstract world models without online experimentation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 10338–10356. PMLR, 2023.

Ignat Georgiev, Krishnan Srinivasan, Jie Xu, Eric Heiden, and Animesh Garg. Adaptive horizon actor-critic for policy learning in contact-rich differentiable simulation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 15418–15437. PMLR, 2024. URL https://proceedings.mlr.press/v235/georgiev24a.html.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *Proceedings of the First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=tEYskw1VY2.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022.

Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, Xiaodi Yuan, Pengwei Xie, Zhiao Huang, Rui Chen, and Hao Su. ManiSkill2: A unified benchmark for generalizable manipulation skills. In *International Conference on Learning Representations*, 2023.

Christian Gumbsch, Noor Sajid, Georg Martius, and Martin V. Butz. Learning hierarchical world models with adaptive temporal abstractions from discrete latent dynamics. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=5qappsbO73r.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems 31*, pp. 2451–2463, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 2555–2565. PMLR, 2019.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025. doi: 10.1038/s41586-025-08744-2.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.16828.

Nicklas Hansen, S V Jyothir, Vlad Sobal, Yann LeCun, Xiaolong Wang, and Hao Su. Hierarchical world models as visual whole-body humanoid controllers. In *International Conference on Learning Representations*, 2025. URL https://iclr.cc/virtual/2025/poster/30793.

Nicklas A Hansen, Hao Su, and Xiaolong Wang. Temporal difference learning for model predictive control. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 8387–8406. PMLR, 2022. URL https://proceedings.mlr.press/v162/hansen22a.html.

W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada. An introduction to event-triggered and self-triggered control. In *IEEE Conference on Decision and Control*, pp. 3270–3285, 2012. doi: 10.1109/CDC.2012.6425820.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Thomas Kipf, Yujia Li, Hanjun Dai, Vinicius Zambaldi, Alvaro Sanchez-Gonzalez, Edward Grefenstette, Pushmeet Kohli, and Peter Battaglia. CompILE: Compositional imitation learning and execution. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 3418–3428. PMLR, 2019.

Maria Krinner, Elie Aljalbout, Angel Romero, and Davide Scaramuzza. Accelerating model-based reinforcement learning with state-space world models. In *ICLR 2025 Workshop on World Models: Understanding, Modelling and Scaling*, 2025. URL https://iclr.cc/virtual/2025/37529.

Vaisakh Shaj Kumar, Saleh Gholam Zadeh, Ozan Demir, Luiz Ricardo Douat, and Gerhard Neumann. Multi time scale world models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Hongyuan Adam Lu, Z. L. Victor Wei, Qun Zhang, Jinrui Zeng, Bowen Cao, Lingwei Meng, Mocheng Li, Zezhong Wang, Haonan Yin, Naifu Xue, Minyu Chen, Cenyuan Zhang, Zefan Zhang, Hao Wei, Jiawei Zhou, Haoran Xu, Hao Yang, Ronglai Zuo, Tongda Xu, Yonghao Li, Jian Chen, Hebin Wang, Zeyu Gao, Yang Li, Wei Zhao, Qimin Zhong, Siqi Liu, Yumeng Zhang, Leyan Cui, Zhangyu Wang, and Wai Lam. Looped world models. *arXiv preprint arXiv:2606.18208*, 2026. doi: 10.48550/arXiv.2606.18208. URL https://arxiv.org/abs/2606.18208.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017.

Alexander Neitz, Giambattista Parascandolo, Stefan Bauer, and Bernhard Schölkopf. Adaptive skip intervals: Temporal abstraction for recurrent dynamical models. In *Advances in Neural Information Processing Systems 31*, pp. 9838–9848. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/hash/0f0ee3310223fe38a989b2c818709393-Abstract.html.

Anh N. Nhu, Sanghyun Son, and Ming C. Lin. Time-aware world model for adaptive prediction and control. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 46265–46287. PMLR, 2025.

Zhao-Han Peng, Shaohui Li, Zhi Li, Shulan Ruan, Yu Liu, and You He. From observations to events: Event-aware world model for reinforcement learning. *arXiv preprint arXiv:2601.19336*, 2026. doi: 10.48550/arXiv.2601.19336. URL https://arxiv.org/abs/2601.19336.

Doina Precup. *Temporal Abstraction in Reinforcement Learning*. PhD thesis, University of Massachusetts Amherst, 2000.

Yulia Rubanova, Ricky T. Q. Chen, and David Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In *Advances in Neural Information Processing Systems*, volume 32, pp. 5320–5330, 2019.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588: 604–609, 2020. doi: 10.1038/s41586-020-03051-4.

Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 8583–8592. PMLR, 2020.

Achyuthan Sivasankar. Adaptive compute in latent world models: When depth helps, hurts, or doesn't matter. *arXiv preprint arXiv:2607.10203*, 2026. doi: 10.48550/arXiv.2607.10203. URL https://arxiv.org/abs/2607.10203.

Jimmy T H Smith, Andrew Warrington, and Scott W Linderman. Simplified state space layers for sequence modeling. In *International Conference on Learning Representations*, 2023.

Richard S Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1-2):181–211, 1999. doi: 10.1016/S0004-3702(99)00052-1.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. DeepMind Control Suite. *Software Impacts*, 6:100022, 2020. doi: 10.1016/j.simpa.2020.100022.

Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. FeUdal networks for hierarchical reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 3540–3549. PMLR, 2017.

Hongwei Wang, Yangru Huang, Guangyao Chen, Xu Wang, and Yi Jin. Beyond single-speed reasoning: Coordinating fast and slow dynamics for efficient world modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 26215–26223, 2026. doi: 10.1609/aaai.v40i31.39825.

Wancong Zhang, Basile Terver, Artem Zholus, Soham Chitnis, Harsh Sutaria, Mido Assran, Randall Balestriero, Amir Bar, Adrien Bardes, Yann LeCun, and Nicolas Ballas. Hierarchical planning with latent world models. *arXiv preprint arXiv:2604.03208*, 2026. doi: 10.48550/arXiv.2604.03208. URL `https://arxiv.org/abs/2604.03208`.

A Implementation Details

Algorithm 1 TempoRoute training

Require: Replay buffer $\mathcal{D}$, backbone SSM$_\theta$, router p_θ, actor π_ϕ, critic V_ψ
Require: Actor-query cap B, physical target horizon H_{phys}, curriculum schedule $\epsilon(t)$
for each training iteration **do**
  Sample batch of trajectories from $\mathcal{D}$
  Infer replay posterior states $x_{0:T}$ and fit standard one-step targets
  Build skip rate and continuation targets from recorded n-step segments
  Build burst targets by partitioning each one-step reward over m recurrent updates
  Compute the model loss in Eq. (12)
  Initialize actor imagination from sampled replay posterior x_0
  for $k = 0, \ldots, B-1$ **do**
    Stop if cumulative physical time reaches $H_{\text{phys}} = 15\Delta t_{\text{env}}$
    Sample $\mathbf{a}_k \sim \pi_\phi(\mathbf{a}_k \mid \mathbf{h}_k, \mathbf{z}_k, \tau_k)$
    Compute the candidate transition in Eq. (5) and evaluate δ_u, δ_v
    Apply any curriculum hint, mask infeasible skips, and sample the route
    Execute the selected operator; predict $(\hat{\rho}_k, \hat{c}_k, \hat{d}_k)$
  end for
  Compute Eq. (3) actor–critic losses; update θ, ϕ, ψ
end for

Replay is stored at environment-step granularity. Standard targets use adjacent posterior boundaries, reward, and continuation. Skip uses a recorded n-step segment: Eq. (2) is divided by n for reward-rate supervision, continuations are multiplied, and the root action conditions a model that marginalizes the recorded continuation actions. Burst anchors only its final recurrent state to the recorded next posterior; equal reward partitions sum to the observed one-step reward, intermediate continuation targets equal one, and the final update receives the recorded continuation. Table 9 evaluates two alternate reward shapes with the same integral.

Training begins with a pretrained fixed-step RSSM backbone. The control-relevance routing head is introduced after 50k environment steps, initially seeded with heuristic changepoint hints (contact

Algorithm 2 TempoRoute deployment decision

Require: Posterior state x_0, actor π_ϕ, critic V_ψ, candidate count P, caps (B, H_{phys})

Replicate x_0 into P active candidates; initialize $Q_i = E_i = T_i = 0$
for $k = 0, \ldots, B-1$ **do**
    Draw one actor action for each active candidate
    Batch one candidate body application and sampled latent per active candidate
    Compute (δ_u, δ_v); mask infeasible route–factor pairs; take route/factor argmax
    Reuse the candidate as standard, skip base, or burst iterate one; finish the selected operator
    Record $(\hat{\rho}, \hat{c}, d, e)$ and update (Q_i, E_i, T_i); deactivate if $Q_i = B$ or $T_i = H_{\text{phys}}$
end for
Evaluate terminal critics and compute each $G_{i,0}$ by Eq. (14)
Execute the root action from $\arg\max_i G_{i,0}$ for one environment step

Table 6: Planner contract used for interpretation. Uniform-Time and TempoRoute form the matched identification pair and share actor shooting, candidate batching, and all caps. Ecological comparators retain their native proposal and scoring rules; the common columns are physical coverage and actor-query capacity, while E records each method's internal transition unit.

Method	Proposal / score	Node transition	Q/T cap	E unit
Uniform-Time	Shared actor shooting	One-step SSM	$15/15$	SSM body
TempoRoute	Alg. 2	Routed skip/std/burst	$15/15$	SSM body
Fixed-step RSSM	Native actor shooting	One-step RSSM	$15/15$	RSSM transition
DreamerV3	Native actor imagination	One-step RSSM	$15/15$	RSSM transition
TD-MPC2	Native latent MPC	One-step latent model	$15/15$	Latent transition
Adaptive ODE	Native latent planner	Adaptive ODE solve	$\leq 15/15$	Solver evaluation

flags, velocity spike detectors) through an ϵ-greedy schedule that decays from $\epsilon_0 = 0.5$ to 0.0 over 200k steps. Velocity spikes are computed from observed state changes; contact flags, where used on contact-rich tasks, come from simulator state and serve only as privileged train-time warm-start hints. The policy and critic are always optimized against the same query cap and physical target horizon used at deployment. The hint-free comparison in Table 8 isolates this optimization aid: deployment uses only uncertainty growth and value sensitivity in every condition.

Table 7: Router signals, their source, and when they are visible. The table separates optional train-time warm-start hints from the deployed routing statistics used at evaluation time.

Signal	Source	Train-time use	Test-time use	Role
Velocity spike heuristic	Observation-derived state changes	Yes	No	Optional early warm-start hint for operator selection.
Contact flag heuristic	Simulator state on some contact-rich tasks	Yes	No	Optional privileged early warm-start hint for operator selection.
Uncertainty growth δ_u	Model predictive-scale change	Yes	Yes	Deployed router statistic used to choose `skip`/`standard`/`burst`.
Value sensitivity δ_v	Critic change across one candidate transition	Yes	Yes	Deployed router statistic used to choose `skip`/`standard`/`burst`.

Table 7 separates optional optimization warm-starts from the two model-derived statistics used at deployment. The zero-hint rows below establish that the final frontier is learned directly from those deployment-visible signals.

TempoRoute combines learned skip/standard/burst placement under a shared actor-query cap, elapsed-time prediction, a planner and critic conditioned on cumulative physical time, and temporal consistency linking aggregate and one-step rollouts. Algorithm 2 closes the path from sampled candidate to executed root action; Tables 13 and 14 quantify router and operator-bound choices.

DMControl locomotion uses pixel observations and therefore optimizes a reconstruction term in image space. ManiSkill2 and the smooth-control tasks use low-dimensional state observations, so the same loss slot is instantiated as state regression rather than pixel reconstruction. The Uniform-Time control in Table 1 keeps the observation pipeline, selective-SSM backbone, and time-aware actor–critic while selecting a standard single-step transition at every node. Each planner call logs Q, R, E, T, operator sequence, planner latency, and GFLOP. Logged E assigns one operator-body unit to `standard` or `skip`, m_k units to `burst`(m_k), and one unit to each adaptive-ODE solver evaluation. GFLOP and latency include routing-cue computation and the complete timed planner.

Each reported training run used approximately 48 A100 GPU-hours under the configuration in Section 5. Evaluation tables aggregate the five declared seeds and their task-specific episode or call denominators.

B Training and Target Robustness

Zero-hint initialization reaches a final quality and compute region close to the current recipe on all three tasks, while requiring 120–150k additional environment steps to cross T90. Observation-derived velocity recovers most of that optimization speed. Each Walker condition uses 5×50 evaluation episodes and each manipulation condition uses 5×100.

Equal-rate, terminal-lumped, and endpoint-only supervision all retain the qualitative saving. Task-dependent differences are modest relative to disabling burst on StackCube and PegInsertion, establishing that allocation gains survive principled changes to the within-step reward target.

Table 8: Hint-free router learning. "Current" is the decayed contact/velocity warm-start, "zero hint" removes both sources from initialization, and "obs. velocity" uses only observation-derived velocity for warm-start. T90 is the environment steps needed to reach 90% of each run's final quality. S/St/B is the deployment operator percentage after all hints have decayed.

Task	Training condition	Final quality	Lat. ms	GFLOP	S/St/B	T90 (k)
Walker	Current	882±21	4.20	58.8	33/63/4	520
Walker	Zero hint	874±25	4.24	59.5	31/64/5	640
Walker	Obs. velocity	880±22	4.21	59.1	32/63/5	560
Walker	Uniform-Time	878±23	4.83	79.1	0/100/0	610
StackCube	Current	73.8±4.0	4.20	56.0	39/47/14	620
StackCube	Zero hint	72.9±4.5	4.27	57.4	37/49/14	760
StackCube	Obs. velocity	73.5±4.2	4.24	56.8	38/48/14	680
StackCube	Uniform-Time	72.4±4.3	4.89	79.7	0/100/0	730
PegInsert	Current	66.2±4.8	4.40	59.9	31/56/13	700
PegInsert	Zero hint	65.0±5.2	4.48	61.2	28/58/14	850
PegInsert	Obs. velocity	65.9±4.9	4.43	60.5	30/57/13	770
PegInsert	Uniform-Time	65.6±5.0	5.02	80.0	0/100/0	820

Table 9: Burst-target robustness under a fixed reward integral. Terminal-lumped supervision assigns the observed reward to the final recurrent update; endpoint-only supervision applies reward loss at the recorded boundary while retaining latent consistency. ΔQuality is the paired quality difference from Uniform-Time with a 95% interval.

Task	Burst target	Quality	Lat. ms	GFLOP / burst %	ΔQuality [95% CI]
Walker	Equal rate	882±21	4.20	58.8/4	$+4$ $[-5, 13]$
Walker	Terminal lump	877±24	4.24	59.5/3	-1 $[-9, 7]$
Walker	Endpoint only	879±23	4.22	59.2/4	$+1$ $[-7, 9]$
Walker	Burst disabled	856±25	3.98	52.1/0	-22 $[-35, -9]$
Walker	Uniform-Time	878±23	4.83	79.1/0	0
StackCube	Equal rate	73.8±4.0	4.20	56.0/14	$+1.4$ $[-0.1, 1.9]$
StackCube	Terminal lump	73.1±4.3	4.25	57.2/13	$+0.7$ $[-0.6, 1.8]$
StackCube	Endpoint only	72.8±4.4	4.23	56.7/13	$+0.4$ $[-0.8, 1.7]$
StackCube	Burst disabled	67.1±5.4	3.90	50.4/0	-5.3 $[-8.0, -2.4]$
StackCube	Uniform-Time	72.4±4.3	4.89	79.7/0	0
PegInsert	Equal rate	66.2±4.8	4.40	59.9/13	$+0.6$ $[-1.8, 3.0]$
PegInsert	Terminal lump	65.7±5.0	4.44	60.7/12	$+0.1$ $[-2.4, 2.7]$
PegInsert	Endpoint only	65.4±5.1	4.43	60.3/12	-0.2 $[-2.8, 2.3]$
PegInsert	Burst disabled	61.8±5.7	4.05	52.8/0	-3.8 $[-6.5, -1.0]$
PegInsert	Uniform-Time	65.6±5.0	5.02	80.0/0	0

C External Assets and Licenses

Table 10 lists the benchmark packages and official baseline reference implementations used for task definitions, licensing, and reproduction notes. Where an official implementation has no tagged release, we record the public repository snapshot available on April 25, 2026; the submission does not redistribute third-party checkpoints.

Table 10: External assets, versions or public snapshots, licenses, and short terms-of-use notes.

Asset	Version / snapshot	License	Terms-of-use note
DMControl benchmark	`dm_control` tag `v1.0.39`	Apache-2.0	Used for Walker-Run, Hopper-Hop, and Cheetah-Run task definitions; reproduce under Apache-2.0 terms and cite Tassa et al. (2020).
ManiSkill2 benchmark	Public snapshot `63e0e7d`; rigid-body task docs `mani_skill2` 0.4.0	Apache-2.0 code; CC BY-NC 4.0 assets	Used for PickCube, StackCube, and PegInsertion task definitions; follow the benchmark asset download instructions and the non-commercial asset terms stated by the official ManiSkill2 documentation.
DreamerV3 reference implementation	Public snapshot `b65cf81` (no tagged release)	MIT	Used only as a reference implementation/specification for the baseline; no third-party checkpoints are redistributed in this submission.
TD-MPC2 reference implementation	Public snapshot `8bbc14e` (no tagged release)	MIT	Used only as a reference implementation/specification for the baseline; no third-party checkpoints are redistributed in this submission.
Adaptive latent ODE baseline	Method specification from Rubanova et al. (2019)	N/A	Implemented in the submission code from the cited paper description; no external code asset or checkpoint is imported.

Table 10 is included because asset ambiguity can block reproduction even when the method description is clear. It separates benchmark task definitions, reference implementations used only for baseline specification, and the one baseline reimplemented directly from the cited method description. No third-party checkpoints, hidden datasets, or proprietary simulator assets are redistributed with this submission. For ManiSkill2, code and task assets follow different licenses, so both are recorded explicitly instead of treating the benchmark as a single legal object. This explicit provenance trail removes version and license ambiguity from the evaluation definition.

D Matched-Family Statistical Detail

The paired design resolves 5/6 tasks within their declared margins and retains PegInsertion as an informative boundary. This conclusion follows the interval criterion directly and does not rely on overlap between marginal standard deviations.

For the random row, each evaluation route and its active factor are sampled from the corresponding trained seed's joint empirical histogram; the model, actor actions, and candidate states are otherwise unchanged. Cost-matched random allocation recovers much of the arithmetic saving but only two equivalence decisions, whereas the joint router preserves five. Uncertainty and value each contribute signal, and their combination provides the strongest quality–compute balance. Burst-disabled attains the lowest cost at the expense of contact-rich quality, complementing Figure 2.

The central frontier shift persists for $B \in \{15, 20\}$ and adjacent operator bounds. Aggressive $B = 10$, $N_{\max} = 6$, or either burst-bound extreme introduces one additional inconclusive task, giving a broad

Table 11: Smooth-control self-selection. TempoRoute routes 94.7% and 96.1% of nodes through `standard`; the displayed means and Q/E show the resulting quality and near-uniform computation.

	Reacher-Easy			Pendulum-Swingup		
Method	Return	Lat. (ms)	Q/E	Return	Lat. (ms)	Q/E
Fixed-step RSSM	941±12	4.5±0.1	15/15	824±19	4.4±0.1	15/15
DreamerV3	968±8	4.8±0.2	15/15	862±14	4.7±0.2	15/15
TD-MPC2	961±10	5.9±0.3	15/15	851±16	5.8±0.3	15/15
Adaptive ODE	952±11	5.2±0.3	15/16.1	838±18	5.1±0.3	15/15.8
TempoRoute (ours)	964±9	4.6±0.2	14.2/14.5	856±15	4.5±0.2	14.4/14.6

Table 12: Paired TempoRoute minus Uniform-Time quality differences. Intervals are seed-level paired 95% confidence intervals; equivalence requires the complete interval to lie inside the predeclared task margin. Raw totals are 5×50 episodes per DMControl task and 5×100 per ManiSkill2 task.

Task	Difference	Paired 95% CI	Margin	Decision
Walker-Run	$+4$	$[-5.0, 13.0]$	± 17.6 return	Equivalent
Hopper-Hop	$+2$	$[-4.1, 6.4]$	± 6.7 return	Equivalent
Cheetah-Run	$+3$	$[-5.2, 11.2]$	± 16.7 return	Equivalent
PickCube	$+0.7$	$[-0.8, 1.8]$	± 2 points	Equivalent
StackCube	$+1.4$	$[-0.1, 1.9]$	± 2 points	Equivalent
PegInsertion	$+0.6$	$[-1.8, 3.0]$	± 2 points	Inconclusive

Table 13: Full-suite router comparison under one backbone and matched operator accounting. Quality reports the number of six task-level equivalence decisions; latency and GFLOP are reductions from Uniform-Time; Q is mean actor-query nodes. The random router matches TempoRoute's task-level operator histograms, while fixed cycle repeats a deterministic skip/standard/burst cadence.

Router	Quality equiv.	Latency reduction	GFLOP reduction	Q
TempoRoute joint cues	5/6	12.6%	27.3%	10.3
Uncertainty only	4/6	10.4%	23.8%	10.7
Value only	4/6	9.1%	21.5%	11.2
Skip disabled	3/6	6.8%	16.7%	13.6
Burst disabled	3/6	14.9%	31.2%	9.7
Random matched histogram	2/6	11.8%	25.9%	10.4
Fixed operator cycle	2/6	10.9%	24.6%	10.8
Uniform-Time	6/6	0.0%	0.0%	15.0

Table 14: Operating-point sensitivity averaged over the six event-heavy tasks. Each row varies one factor from $(B, N_{\max}, M_{\max}) = (15, 4, 3)$, retains a 15-step physical horizon, and reports five-seed means. The last two rows summarize Reacher-Easy over the full grid.

Setting	Quality equiv.	Lat. ms	Q	GFLOP	S/St/B
$B = 10,\ N = 4,\ M = 3$	4/6	3.18	9.2	44.2	39/45/16
$B = 15,\ N = 4,\ M = 3$	5/6	4.25	10.3	57.8	35/51/14
$B = 20,\ N = 4,\ M = 3$	5/6	4.34	10.5	59.1	35/50/15
$B = 15,\ N = 2,\ M = 3$	4/6	4.54	12.2	63.9	22/62/16
$B = 15,\ N = 6,\ M = 3$	4/6	4.16	9.5	56.4	41/43/16
$B = 15,\ N = 4,\ M = 2$	4/6	4.10	10.1	54.3	36/52/12
$B = 15,\ N = 4,\ M = 4$	4/6	4.47	10.5	63.1	33/48/19
Reacher standard share, grid min	—	—	—	—	92.8%
Reacher max return change	1.3%	—	—	—	—

middle operating region rather than a single tuned point. Reacher remains predominantly standard throughout.

Table 15: Cue-corruption detail for Walker-Run, StackCube, and PegInsertion. Latency reduction is relative to each task's fixed-step RSSM; quality is converted to percentage change from clean TempoRoute within each task before averaging. Intervals bootstrap seed-level task aggregates over 30000 planner decisions per seed and noise level (150,000 total).

σ	Latency reduction	95% CI	Quality change
0.00	15.4%	[14.1,16.7]%	0.0%
0.05	10.2%	[8.8,11.6]%	−0.1%
0.10	5.6%	[4.0,7.2]%	−0.4%
0.20	1.7%	[0.5,2.9]%	−0.9%
0.30	1.9%	[0.3,3.5]%	−1.6%

The gradual, slightly non-monotone contraction moves the router toward Uniform-Time cost while mean normalized quality changes from 0.0% to −1.6%. This mechanism-level response links cue reliability to expected benefit and explains both the crossover in Figure 3 and smooth-task standard routing.

E Additional Diagnostics

Across the nearest prior families, temporal adaptation becomes easier to compare once decomposed into three axes: what changes over time, what triggers that change, and what remains directly budget-comparable at deployment. TempoRoute occupies the planner-visible latent-operator cell: learned control-relevance signals route computation while fixed action cadence, actor-query semantics, and physical horizon preserve direct comparability. Options, adaptive ODEs, and event-aware representations move adjacent axes, making this decomposition operational for both method design and evaluation.

The fixed-cadence and query-semantics columns identify the direct control used in Table 1; the remaining families establish the surrounding design space.

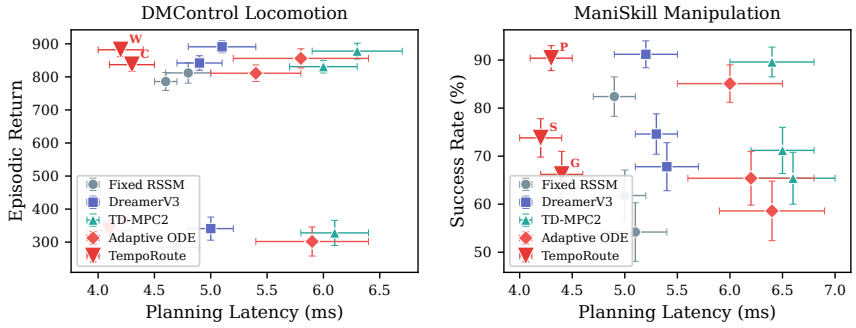

Figure 4: Latency–performance scatter for Appendix Table 17. Labels identify TempoRoute task points: W/H/C are Walker/Hopper/Cheetah and P/S/G are PickCube/StackCube/PegInsertion. Every point moves left of its strongest-quality fixed-step comparator at the reported means; PegInsertion has the largest relative shortfall at 2.4%.

Figure 4 makes the repeated leftward shift visible across both task families; Tables 1 and 17 provide the corresponding estimates and accounting.

Appendix Figure 5 is a qualitative mechanism check on one representative Walker-Run segment. Its localized burst-probability peaks at stance contacts complement the suite-level matched-placement, operator-ablation, and cue-corruption evidence.

Table 16: Design-space map against adjacent temporal-adaptation families. Columns identify the adapted quantity, trigger, executed action cadence, query semantics, and internal-compute readout. TempoRoute uniquely makes temporal coverage and recurrent depth one planner-node decision while separating actor queries, body applications, and measured cost.

Family	What changes	Main trigger	Fixed action cadence?	Query semantics	Compute readout
Fixed-step world models	Latent rollout stays at one environment step	None	Yes	One query / step	Latent transitions
Adaptive skip intervals	Prediction interval	Predictive fit	N/A	Prediction endpoints	Model evaluations
Recurrent-depth world models	Predictor depth	Exit / depth criterion	Yes	One query / step	Recurrent depth
Options / action repeat	Primitive action duration	Policy termination or fixed repeat	No	Option boundaries	Executed decisions
Adaptive-step / adaptive-horizon models	Solver step size or rollout depth	Numerical error, stiffness, or truncation rule	Usually yes	Policy nodes vs. solver steps	Solver evaluations
Event-aware / hierarchical latent models	Granularity or abstraction level	Learned segmentation or hierarchy	Varies	Varies by hierarchy	Rep./planning
TempoRoute (this paper)	Coverage plus recurrent depth	Control-relevance router	Yes	One query / routed node	Q/E separated

Table 17: Complete ecological comparator matrix for the compact DreamerV3/TempoRoute comparisons in Tables 3–4. Values are mean ± standard deviation over five seeds. Q/E reports actor-query nodes and operator-body evaluations; Adaptive ODE counts solver evaluations in E, and TempoRoute counts every burst recurrent update.

	Walker-Run			Hopper-Hop			Cheetah-Run		
Method	Return	ms	Q/E	Return	ms	Q/E	Return	ms	Q/E
Fixed-step RSSM	812±31	4.8±0.2	15/15	284±42	4.7±0.2	15/15	786±27	4.6±0.1	15/15
DreamerV3	891±18	5.1±0.3	15/15	341±35	5.0±0.2	15/15	842±22	4.9±0.2	15/15
TD-MPC2	878±24	6.3±0.4	15/15	328±38	6.1±0.3	15/15	831±19	6.0±0.3	15/15
Adaptive ODE	856±29	5.8±0.6	15/18.2	302±44	5.9±0.5	15/19.4	811±25	5.4±0.4	15/17.1
TempoRoute	882±21	4.2±0.2	10.8/11.3	335±33	4.1±0.2	10.2/10.9	837±20	4.3±0.2	11.4/11.8
	PickCube			StackCube			PegInsertion		
Method	Succ.%	ms	Q/E	Succ.%	ms	Q/E	Succ.%	ms	Q/E
Fixed-step RSSM	82.4±4.1	4.9±0.2	15/15	61.8±5.3	5.0±0.2	15/15	54.2±6.1	5.1±0.3	15/15
DreamerV3	91.2±2.8	5.2±0.3	15/15	74.6±4.2	5.3±0.2	15/15	67.8±5.0	5.4±0.3	15/15
TD-MPC2	89.6±3.1	6.4±0.4	15/15	71.2±4.8	6.5±0.3	15/15	65.4±5.4	6.6±0.4	15/15
Adaptive ODE	85.1±3.9	6.0±0.5	15/19.8	65.4±5.6	6.2±0.6	15/21.3	58.6±6.2	6.4±0.5	15/22.1
TempoRoute	90.4±2.6	4.3±0.2	9.8/10.4	73.8±4.0	4.2±0.2	9.4/10.8	66.2±4.8	4.4±0.2	10.1/11.5

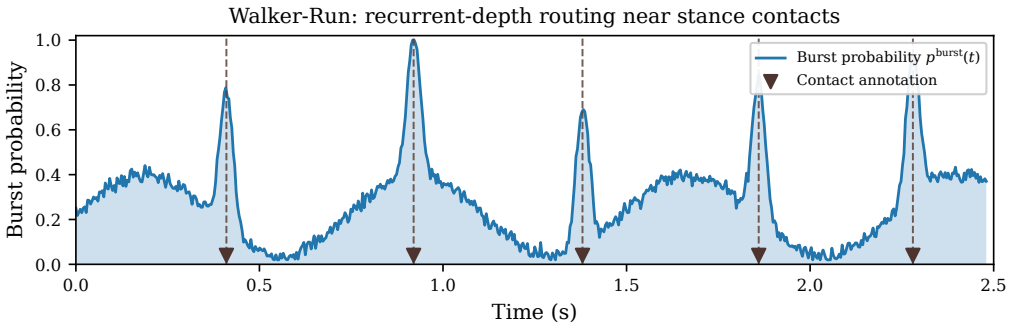

Figure 5: Walker-Run segment: deployed burst probability $p^{\text{burst}}(t)$ with stance-contact annotations. Localized peaks are qualitatively consistent with concentrating recurrent-depth operators near foot–ground contacts.