When to Re-Plan: Subgoal Persistence in Hierarchical Latent Reasoning

Ayushi Chadha [1]

Abstract

Long-horizon reasoning requires a system to commit to medium-horizon intent without becoming rigid: re-plan too often and computation never coheres into multi-step structure; commit too long and the plan goes stale. We study this stability–adaptivity tradeoff in the latent reasoning setting, where multi-step computation occurs inside hidden state rather than externalized token traces. We extend the Hierarchical Reasoning Model (HRM) with a feudal-style manager–worker interface: a slow high-level module periodically emits a normalized directional subgoal that persists for P low-level steps, biasing the worker's hidden-state updates and supplying an intrinsic cosine alignment loss. On ARC and ConceptARC, we find that subgoal *persistence* — not subgoal injection alone — is the central knob: moderate periods $P \in [3, 6]$ consistently outperform both very frequent (P=1) and longer horizons, with a clear minimum LM loss at P=3 (1.544 vs. 1.674 at P=1 and a 1.640 baseline; replicated over five seeds with mean 1.595 and standard deviation 0.045). The intrinsic alignment weight λ shows a complementary narrow optimum ($\lambda \approx 0.05$), and a controlled ablation at past-sweet-spot λ isolates *learned* directional structure — not architectural capacity or auxiliary loss alone — as the source of interference when the alignment signal exceeds its optimum. Together these findings implicate a design principle for compositional planning in latent reasoning systems: medium-horizon intent must be coherent across enough computational steps for compositional structure to form.

1. Introduction

A long-standing observation in cognitive science is that human reasoning operates on at least two distinct timescales: a fast, automatic, intuition-driven mode and a slow, deliberate, effortful mode (Kahneman, 2011; Evans & Stanovich, 2013). Modern reasoning systems in machine learning have begun to recapitulate something like this structure. Chain-of-thought prompting (Wei et al., 2022) makes deliberation explicit by externalizing it as a token sequence; latent reasoning architectures internalize multi-step computation inside hidden state. Among the latter, adaptive computation time (Graves, 2016) allocates variable depth as needed, and the Hierarchical Reasoning Model (HRM) (Wang et al., 2025) couples a slow high-level loop with a fast low-level loop and a halting mechanism, achieving deep latent computation in compact models trained from scratch. These architectures gain speed and flexibility by giving up the legibility of the reasoning trace — but in doing so, they raise a question the externalized setting does not have to confront.

When deliberation is a sequence of explicit tokens, each token implicitly commits the system to a piece of intent: the next word constrains what comes after. When deliberation is a sequence of hidden-state updates, no such commitment is forced. A latent reasoner can in principle revise its medium-horizon intent at every micro-step, or never. The architectural form does not, by itself, determine the temporal scale of planning. **How long should a latent reasoner commit to an intent before revising it?**

We study this question through the lens of *subgoal persistence*. One way to give a latent reasoner explicit medium-horizon intent is to add a slow module that periodically emits a subgoal — a directional vector in latent space — that persists across multiple fast-module steps before being revised. The choice of how long to commit to a subgoal is then the central knob, and it embodies a stability–adaptivity tradeoff that any planner organized around intermediate intent must resolve. Update too frequently and each subgoal is overwritten before the worker can compose it into multi-step computation; commit too long and the subgoal ossifies as the worker's hidden state evolves past it. This tradeoff has been studied extensively in feudal reinforcement learning (Vezhnevets et al., 2017; Sutton et al., 1999), where it underwrites the design of options and manager–worker hierarchies. We import this commitment-duration lens into latent reasoning, where the worker's "actions" are hidden-state updates rather than environment interactions and the cost of stale plans is representational rather than behavioral.

[1]Independent Researcher, India. Correspondence to: Ayushi Chadha <ayushichadha48@gmail.com>.

Accepted to the 2nd Workshop on Compositional Learning at ICML 2026, Seoul, South Korea. *Copyright 2026 by the author(s).*

We propose **Subgoal-Augmented HRM**, a feudal-style extension of HRM that operationalizes persistent latent intent. Every P low-level micro-steps, the high-level module emits a normalized directional subgoal g; the worker's hidden-state updates are biased by g through a learned projection, and a cosine alignment loss rewards net low-level displacement that progresses along g. Crucially, P — the manager period — is exposed as a hyperparameter we sweep, allowing us to characterize the persistence regime in which subgoals actually help. We complement the persistence sweep with two further studies: a sweep over the alignment loss weight λ that distinguishes "subgoal as prior" from "subgoal as constraint," and a controlled ablation that isolates the contribution of *learned* directional structure from architectural capacity alone.

Our findings on ARC and ConceptARC point to a single design principle. Persistence is necessary, not optional: at $P{=}1$, the full subgoal infrastructure performs *below* the no-subgoal baseline — the cleanest evidence in our study that the persistence mechanism, not the injection mechanism, drives the improvement. The benefit emerges sharply at $P{=}3$ and decays only gradually out to $P{=}8$, an asymmetry suggesting that staleness is a softer failure mode than instability. The intrinsic weight λ shows a complementary narrow optimum at $\lambda \approx 0.05$, consistent with the mechanism functioning as a lightweight planning prior rather than a dominant geometric constraint. A controlled ablation at past-sweet-spot λ isolates *learned* directional structure — not architectural capacity or auxiliary loss alone — as the source of interference when the alignment signal exceeds its optimum. Results are replicated across 5 seeds at the best configuration.

Scope. This work is an empirical and behavioral characterization of when and how persistent directional subgoals improve hierarchical latent reasoning, complemented by a controlled ablation that isolates the source of past-sweet-spot interference. A representation-level analysis of how persistent subgoals shape worker hidden-state geometry is a natural complement we leave for follow-up work; we discuss this in Section 7.

We frame this as a contribution to compositional learning. The manager–worker decomposition gives an explicit interface between intent and execution, allowing the worker to assemble a sequence of subgoal-conditioned hidden-state updates into longer-horizon computation. The persistence finding identifies the *necessary condition* under which this assembly actually occurs. A subgoal mechanism without persistence has the architectural form of compositionality (manager, worker, intent vector) but not its function: it cannot produce the temporally extended structure that distinguishes "planning a sequence of steps" from "taking a single step with extra signal." The $P{=}1$ result is the cleanest evidence of this gap.

Contributions.

1. We characterize a stability–adaptivity tradeoff in subgoal persistence for hierarchical latent reasoning, identifying $P \in [3, 6]$ as a consistent sweet spot on ARC/ConceptARC and $P{=}1$ as underperforming the no-subgoal baseline. We replicate the best configuration across 5 seeds (Sections 3–4).
2. We show that the alignment loss weight λ exhibits a complementary narrow optimum at $\lambda \approx 0.05$, consistent with directional subgoals acting as a lightweight planning prior rather than a dominant constraint (Section 4).
3. We provide a controlled ablation at past-sweet-spot λ that isolates learned directional structure — not architectural capacity or auxiliary loss alone — as the source of interference when the alignment signal exceeds its optimum (Section 5).

2. Background and Related Work

Latent reasoning and adaptive computation. Chain-of-thought prompting (Wei et al., 2022) externalizes reasoning as token sequences, gaining interpretability and trainability at the cost of latency and brittleness to step ordering. Latent reasoning architectures internalize multi-step computation in hidden state. Adaptive Computation Time (Graves, 2016) allocates variable depth as needed by learning a halting probability, increasing effective depth when the input warrants it. The Hierarchical Reasoning Model (HRM) (Wang et al., 2025) combines multi-timescale recurrence with halting: coupled fast and slow loops exchange state through learned dynamics, with an ACT-style halting head choosing the number of segments executed per example. HRM achieves deep latent computation in compact models trained from scratch on abstract reasoning tasks. However, coordination between the two levels in HRM is implicit: there is no explicit signal carrying medium-horizon intent from the slow loop to the fast one, and the temporal scale of any planning that occurs must emerge from the recurrent dynamics alone.

Feudal hierarchies and temporal abstraction. The gap left implicit in HRM — an explicit medium-horizon intent signal between hierarchical levels — is precisely the design problem that feudal hierarchies in reinforcement learning address. The framework we extend originates in feudal reinforcement learning, where a manager emits latent goals and a worker is incentivized to follow them (Vezhnevets et al., 2017). The broader principle that temporally extended actions admit semi-Markov treatment and provide credit-assignment leverage is formalized by the options

framework (Sutton et al., 1999). Both lines of work establish that *commitment duration* is a first-class design choice: a manager that re-issues goals every step is functionally not a manager, and an option without a stable termination condition is not an option. We import this commitment-duration lens from RL planning into latent reasoning, treating the worker's hidden-state updates as the analogue of environment-interaction "actions" and the manager's emission period as the analogue of an option's expected duration.

Compositional planning. Subgoal decomposition is a canonical mechanism by which planners achieve compositional generalization in feudal and options-based reinforcement learning (Vezhnevets et al., 2017; Sutton et al., 1999): a complex task is solved by composing solutions to simpler subgoals over extended horizons. We adopt this lineage in latent reasoning, treating each persistent subgoal as a directional primitive that the worker composes — through a sequence of subgoal-conditioned hidden-state updates — into longer-horizon computation. The stability–adaptivity tradeoff we characterize is, in this view, a tradeoff about how long each primitive must remain in force for composition to occur.

3. Method

3.1. HRM Backbone

We build on HRM with two coupled latent states: a slow high-level state z_t^H and a fast low-level state z_t^L. Given an encoded input $\tilde{x}_t$, the model performs iterative latent computation within a segment using recurrent transformer blocks. At each micro-step t, the low-level state updates:

$$z_{t+1}^L = f_L(z_t^L, z_t^H, \tilde{x}_t; \theta_L). \tag{1}$$

The high-level state updates every T low-level steps:

$$z_{t+1}^H = \begin{cases} f_H(z_t^H, z_{t+1}^L; \theta_H), & (t+1) \bmod T = 0, \\ z_t^H, & \text{otherwise.} \end{cases} \tag{2}$$

After N micro-steps, the segment produces a prediction $\hat{y} = f_O(z_N^H; \theta_O)$. Here T is HRM's intrinsic high-level update period — a small fixed integer, typically T=2 in our experiments following Wang et al. (2025) — and is independent of the manager period P we introduce in Section 3.2 for subgoal emission. HRM training uses deep supervision across segments with state detached between segments, and an ACT-style halting objective (Graves, 2016) chooses the number of executed segments per example. We treat the halting objective as part of the HRM baseline.

3.2. Subgoal Emission and the Persistence Period P

We add a feudal manager–worker interface inside HRM's latent computation. The central design choice is the **manager period** P: the number of low-level steps a subgoal persists before being revised. This parameter controls the stability–adaptivity tradeoff directly — small P permits rapid adaptation but prevents temporally extended computation from cohering around any single intent; large P provides commitment but risks staleness as the worker's hidden state evolves past the issued goal.

Every P low-level micro-steps, at times $t_k = kP$, the high-level state emits a directional subgoal:

$$\tilde{g}_k = W_g\, z_{t_k}^H, \qquad g_k = \frac{\tilde{g}_k}{\|\tilde{g}_k\|_2 + \varepsilon}. \tag{3}$$

A scalar commitment gate $\alpha_k = \sigma(w_\alpha^\top z_{t_k}^H) \in (0, 1)$ optionally modulates the strength of the subgoal's influence, allowing the manager to soften commitment when the high-level state is uncertain. For steps $t \in [t_k, t_k + P)$, the active goal and gate are $g(t) = g_k$ and $\alpha(t) = \alpha_k$.

Why direction, not target. A directional goal encodes "where to move" in latent space without forcing the worker to hit a specific absolute target — a target-state formulation would be brittle under the nonstationary internal dynamics of a recurrent reasoner. Direction is the minimal structure that lets the manager commit to medium-horizon intent without overspecifying execution.

3.3. Subgoal Injection

The active goal is injected as an additive bias into the worker's update. Let V_L be a learned projection into the low-level latent space:

$$z_{t+1}^L = f_L\big(z_t^L,\ z_t^H,\ \tilde{x}_t + \alpha(t)\, V_L\, g(t);\ \theta_L\big)\,. \tag{4}$$

Optionally, the goal is also injected at the high-level update step via a projection V_H:

$$z_{t+1}^H = f_H\big(z_t^H,\ z_{t+1}^L + \alpha(t)\, V_H\, g(t);\ \theta_H\big)\,, \quad \text{when } (t+1) \bmod T = 0. \tag{5}$$

The injection converts medium-horizon intent into a persistent steering term that shapes multiple consecutive worker updates, rather than relying on emergent cross-level interactions to carry intent forward.

3.4. Cosine Alignment Loss

Persistent injection alone biases each update but does not guarantee the worker's trajectory progresses *along* the issued direction. We add a cosine-based intrinsic loss — the *feudal alignment loss* in the terminology of Vezhnevets et al. (2017), which we abbreviate to *alignment loss* for the remainder of this paper — that rewards net low-level displacement aligned with the active subgoal. Over the interval $[t_k, t_k + P)$, define $\Delta z_k^L = z_{t_k+P}^L - z_{t_k}^L$, the net

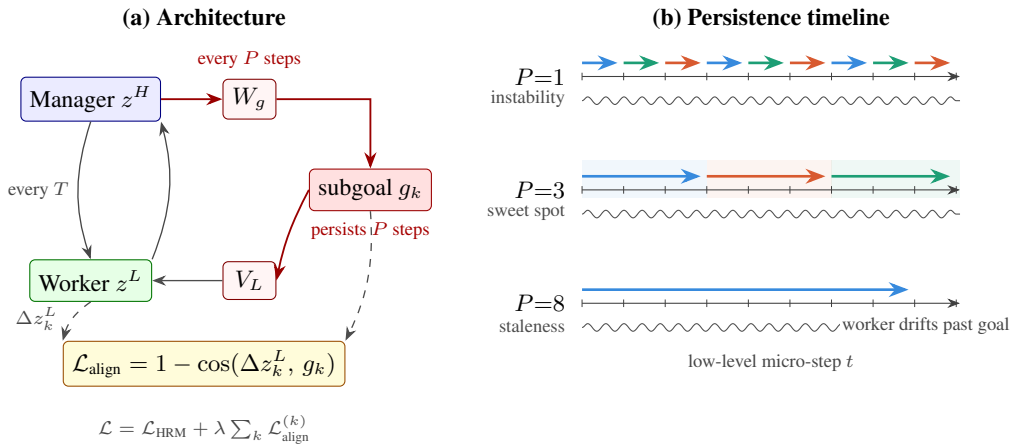

Figure 1. Subgoal-Augmented HRM. **(a)** Architecture. The manager (high-level state z^H, blue) projects through W_g to emit a normalized directional subgoal g_k every P low-level steps. The subgoal is injected into the worker's update via V_L (additive bias), persisting for P steps before re-emission. Worker displacement Δz_k^L over the commitment window is compared to g_k via $\mathcal{L}_{\text{align}}$, added to the HRM objective with weight λ. Grey arrows indicate HRM's standard slow–fast recurrent coupling. **(b)** Persistence timeline over nine micro-steps. Coloured arrows above each axis denote subgoal emissions; their length spans the persistence window. At P=1 a fresh direction is issued every step (no temporal coherence); at P=3 three windows accumulate aligned worker displacement (shading); at P=8 a single subgoal persists while the worker drifts past it. The wavy line sketches the worker's hidden-state trajectory z^L.

displacement of the worker's hidden state during the subgoal's commitment window. The alignment loss is:

$$\mathcal{L}_{\text{align}}^{(k)} = 1 - \cos\big(\Delta z_k^L,\, g_k\big) = 1 - \frac{\langle \Delta z_k^L,\, g_k \rangle}{\|\Delta z_k^L\|_2\, \|g_k\|_2}. \quad (6)$$

With optional gating, $\mathcal{L}_{\text{align}}^{(k)} = \alpha_k \big(1 - \cos(\Delta z_k^L, g_k)\big)$.

3.5. Full Training Objective

Let $\mathcal{L}_{\text{HRM}}$ denote the baseline HRM objective including task loss and halting losses, unchanged from Wang et al. (2025). We add the intrinsic alignment term, summed over the $\lfloor N/P \rfloor$ subgoal windows within the current segment of N micro-steps and weighted by λ:

$$\mathcal{L} = \mathcal{L}_{\text{HRM}} + \lambda \sum_{k=1}^{\lfloor N/P \rfloor} \mathcal{L}_{\text{align}}^{(k)}. \quad (7)$$

Because HRM detaches state between segments, the alignment loss is computed within each segment and does not propagate gradients across segment boundaries, consistent with HRM's truncated-gradient training.

4. Experiments

4.1. Experimental Regimes

We report results from two complementary experimental setups, distinguished by batch size, hardware, and dataset-augmentation factor. The **main study** (Sections 4.2 through 4.5) characterizes the mechanism's behavior across hyperparameters — the persistence sweep over P, the alignment-weight sweep over λ, and the 5-seed replication of the best configuration — using a global batch size of 768 on CPU on the `arc-aug-1000` corpus (`num_aug=1000`). The **ablation study** (Section 5) characterizes the causal contribution of *learned* directional structure under controlled conditions, using a global batch size of 64 on a single NVIDIA L4 GPU on a comparable corpus constructed from the same source datasets but with reduced augmentation (`num_aug=100`) for tractability under the smaller batch size. Within each setup, all hyperparameters other than the variables under study are held fixed, and all reported numbers come from the same metric (`train/lm_loss`) on identically-sized batches at matched optimization steps.

The two setups produce different absolute LM-loss values for the same nominal configuration — for example, λ=0.10, P=4 yields 1.636 in the main study and 1.327 in the ablation study — reflecting differences in batch-size optimization dynamics and effective dataset size rather than any change in the model or objective. The *magnitude* of past-sweet-spot interference also differs: at λ=0.10 the main study shows the mechanism barely matching baseline (1.636 vs. 1.640), while the ablation study at the same λ shows the mechanism clearly underperforming baseline (1.327 vs. 1.227), consistent with smaller batches making auxiliary-loss interference more visible. We therefore make all comparisons *within* their respective study and do not interpret absolute differences *across* studies as meaningful. The two studies are nonetheless qualitatively consistent: both place λ=0.10 at or below the no-subgoal baseline within their own setup, consistent with λ=0.10 lying past

the optimum identified in the main λ sweep (Section 4.4). The ablation study extends this picture by isolating *why* the past-sweet-spot configuration interferes.

4.2. Setup

Data. Our primary training corpus is derived from ARC-AGI (Chollet, 2019) and ConceptARC (Moskvichev et al., 2023), using the `arc-aug-1000` configuration. Each puzzle is represented as a flattened 30×30 grid sequence over a vocabulary of 12 symbols (padding, end-of-sequence, and ten colors). Each puzzle is augmented up to 1000 times during training via dihedral transformations (rotations and reflections), random color permutations, and translational padding-based shifts, following standard ARC augmentation practice. For cross-task validation we additionally report results on ConceptARC-mini, a held-out subset of $\sim$4,900 training and $\sim$400 validation samples grouped by puzzle type.

Architecture. HRM backbone with hidden size 512, 4 high-level and 4 low-level transformer layers, 8 attention heads, recurrent cycles $H_{\text{cycles}} = L_{\text{cycles}} = 2$, ACT with maximum 16 internal steps, RoPE positional encoding, SwiGLU feedforward layers, and RMS normalization. The subgoal head projects z^H to a 512-dimensional directional vector. All comparisons within a study use matched compute and identical backbone configurations; the only variation is the subgoal mechanism's hyperparameters (P and λ).

Optimization. We use the AdamATan2 optimizer with base learning rate 10^{-4} and weight decay 0.1, with puzzle-specific embeddings on a separate learning rate of 10^{-2}. A cosine schedule with 2000 warmup steps controls the learning rate.

Metrics. We report LM loss as the primary metric: token-level stablemax cross-entropy, masked over valid output positions, computed on the training set at matched optimization steps. Secondary metrics include token-level accuracy, exact accuracy (all output tokens correct), the alignment loss itself, and average ACT halting depth.

Compute. The main study reported in this section was conducted on CPU with global batch size 768. The ablation study reported in Section 5 was conducted on a single NVIDIA L4 GPU (24 GB) hosted on E2E Networks with global batch size 64. Section 4.1 discusses how this difference is handled across the paper.

4.3. Persistence Is the Central Knob

We sweep the manager period P at fixed alignment-loss weight λ=0.05, holding all other hyperparameters fixed. Figure 2(a) plots the sweep curve.

Three observations are central to our claim.

Persistence is necessary, not optional. At P=1 — full subgoal infrastructure but no commitment beyond a single step — LM loss is 1.674, worse than the 1.640 baseline without any subgoal mechanism. A subgoal that is overwritten every step does not function as a subgoal. This is the cleanest evidence in our study that the persistence mechanism, not the injection mechanism, drives the improvement.

The benefit emerges sharply between P=2 and P=3. P=2 improves over baseline only marginally (1.638 vs. 1.640). The transition to P=3 drops LM loss by 0.094 — by far the largest single step in the sweep. This suggests a minimum coherence horizon below which subgoals cannot organize compositional structure.

Decay beyond the optimum is gradual. From P=3 to P=8, LM loss varies in a narrow band $[1.544, 1.590]$, with no monotone degradation. Subgoals tolerate moderate over-commitment but are substantially harmed by under-commitment. We interpret this asymmetry as evidence that staleness is a softer failure mode than instability: a slightly stale subgoal still provides directional consistency, while an instantaneously revised subgoal provides none.

Replication across seeds. We replicate the two best configurations across 5 independent training runs each. At λ=0.05, P=3: mean LM loss 1.595 (std 0.045, min 1.540, max 1.662). At λ=0.05, P=4: mean LM loss 1.601 (std 0.048). The seed-level variability is small relative to the gap between P=3 (1.544) and P=1 (1.674) and between P=3 and the no-subgoal baseline (1.640), confirming that the persistence finding is not a single-seed artifact.

4.4. The Alignment Weight λ Has a Narrow Optimum

Fixing P=4, we sweep the alignment-loss weight λ. Figure 2(b) plots LM loss and token-level accuracy against λ.

The optimum at λ=0.05 is narrow: λ=0.10 already approaches the no-subgoal baseline, and $\lambda \geq 0.20$ degrades below it. We interpret this as evidence that the alignment loss is most useful as a *prior* that gently shapes worker dynamics, rather than as a *constraint* that dominates task optimization. When the intrinsic geometric signal competes with the task gradient on roughly equal terms ($\lambda \geq 0.10$), the worker's representational flexibility is reduced enough to harm performance. This complements the persistence finding: subgoals work best when they are committed-to long enough to organize computation ($P \in [3, 6]$) but soft enough not to override it ($\lambda \approx 0.05$). We additionally veri-

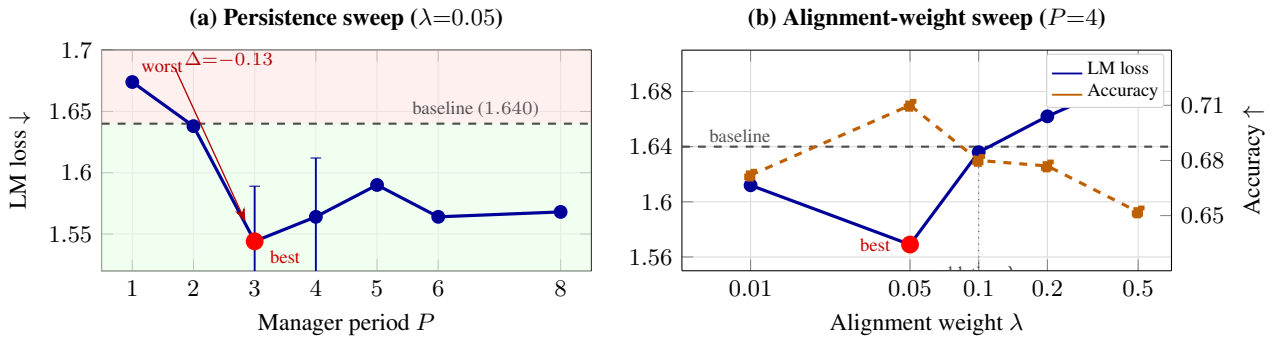

Figure 2. Main study results on ARC-AGI/ConceptARC. **(a)** Persistence sweep at λ=0.05. The shaded green/red regions mark below/above the no-subgoal baseline (1.640). P=1 underperforms baseline (worst); the benefit emerges sharply at P=3 (Δ=−0.13, red point) and decays only gradually out to P=8 — staleness is a softer failure mode than instability. Error bars at P=3 and P=4 are 5-seed standard deviations. **(b)** Alignment-weight sweep at P=4. LM loss (blue, left axis) shows a narrow optimum at λ=0.05, with token-level accuracy (orange dashed, right axis) tracking the same optimum. The ablation regime (λ=0.10) lies past the optimum and is the focus of Section 5.

fied that removing ℓ_2 normalization of g_k or the commitment gate α_k degrades performance, confirming the directional semantics and soft-prior role of the mechanism; a more substantive ablation separating learned directional structure from the architectural addition appears in Section 5.

4.5. Cross-Task Validation on ConceptARC-mini

To check whether the persistence effect generalizes beyond the `arc-aug-1000` corpus, we evaluate on ConceptARC-mini, a held-out task family grouped by puzzle type. The no-subgoal baseline achieves LM loss 2.316; Subgoal-Augmented HRM at λ=0.05, P=3 achieves 2.308. The improvement is modest in absolute terms ($\sim$0.4%), and we therefore treat this as a directionally consistent secondary observation rather than independent benchmark evidence: the headline persistence and λ effects characterized above remain the load-bearing claims.

5. Past-Sweet-Spot Ablation: What Causes Interference?

The main λ sweep (Section 4.4) characterizes the *symptom* of past-sweet-spot interference but does not identify its *source*. We address this with a controlled three-cell ablation at λ=0.10 — past the optimum, where interference is reliably observable. The cells (Table 1) differ only in three flags governing the subgoal mechanism, isolating learned directional structure as the variable of interest while controlling for architectural capacity and the auxiliary loss term.

5.1. Setup

Hardware, batch size, and data. Training was performed on a single NVIDIA L4 GPU with global batch size 64, in contrast to the main study's CPU runs at batch size 768 (Section 4.1). All three cells trained on the same ARC-AGI plus ConceptARC augmented corpus (`num_aug=100`, `seed=42`; 95,993 puzzle identifiers), constructed from the same source corpora as the main study but with reduced augmentation factor for tractability under the smaller batch size.

Table 1. Ablation conditions at λ=0.10, P=4. All three cells share an identical HRM backbone, optimizer, dataset, and training duration, and differ only in the three subgoal-mechanism flags shown. **A_full** is Subgoal-Augmented HRM as proposed. **B_baseline** disables both injection and the alignment loss, reducing the model to vanilla HRM with no manager–worker interface. **E_random** keeps both active but replaces the manager's *learned* directions with random unit vectors, isolating the contribution of learned directional content from the architectural addition and auxiliary loss term.

Cell	`inject_subgoal`	`use_alignment_loss`	`random_directions`
A_full	true	true	false
B_baseline	false	false	false
E_random	true	true	true

Hyperparameters and training duration. All three cells used λ=0.10, P=4, learning rate 10^{-4}, 500 training epochs, and a single fixed random seed. The HRM backbone and optimizer match Section 4.2. All three cells terminated at the same optimization step (30,321 of an intended 31,143, the last full epoch boundary that fit within the iteration budget) under identical software conditions — this shared termination is a property of the training driver and makes the three cells directly comparable. We report training-set LM loss (the same metric as the main study), under the within-regime comparison rule described in Section 4.1.

5.2. Results

Three within-regime contrasts are central.

Table 2. Train LM loss at step 30,321 for the three ablation cells. The within-regime contrast structure — not the absolute values — is what the result speaks to.

Cell	Train LM loss ↓	Δ vs. B_baseline
A_full	1.327	+0.100
B_baseline	1.227	—
E_random	1.230	+0.003

A vs. B: at λ=0.10, the full mechanism interferes. Adding learned directional subgoals raises LM loss by $\sim$0.10 relative to the no-subgoal baseline, consistent with the main λ sweep (Section 4.4, Figure 2(b)) where λ=0.10 already performs at or below baseline.

E vs. B: random directions are nearly indistinguishable from baseline. When learned directions are replaced with random unit vectors, LM loss matches baseline within 0.003 — an order of magnitude smaller than the seed-level variability (0.045) observed in the main study's replication. The architectural addition and auxiliary loss term are not, by themselves, harmful at λ=0.10.

A vs. E: interference is specifically driven by *learned* directional content. The 0.097 gap between A_full and E_random — much larger than the 0.003 gap between E_random and baseline — isolates the interference as specifically due to the *learned* directional content of the subgoals, not the additional capacity, auxiliary objective, or injection mechanism per se.

5.3. Interpretation

This ablation refines the picture established by the λ sweep. The narrow optimum at λ≈0.05 shows *that* the alignment loss functions best as a soft prior; the ablation isolates *why* it stops functioning as a soft prior past the optimum. The issue is not generic auxiliary-loss interference — random directions, with the same loss term, do not interfere. Rather, learned directional structure is doing real representational work in A_full (which is why it helps at λ=0.05 and hurts at λ=0.10); when the alignment weight competes with the task gradient on roughly equal terms, the worker is pulled toward directions that serve alignment but not the LM objective. This contrast identifies learned directional structure as the active component: it helps near the optimum but becomes harmful when over-weighted.

6. Discussion

The persistence sweep identifies a regime in which medium-horizon intent measurably helps a latent reasoner, and identifies the failure modes outside it. We organize the discussion around the stability–adaptivity tradeoff.

The persistence tradeoff. A subgoal revised every step provides no temporal structure for the worker to compose: each micro-update receives a fresh, potentially uncorrelated direction, the alignment loss provides high-variance gradients with no medium-horizon signal, and the injection adds noise the worker must compensate for. This is why P=1 underperforms baseline. Between P=3 and P=6, subgoals persist long enough for multiple consecutive updates to accumulate aligned displacement, and the alignment loss rewards consistent progress rather than instantaneous alignment. The asymmetry between under- and over-commitment (P=1 catastrophic, P=8 only mildly degraded) suggests staleness is a soft failure mode while absence of commitment is not: compositional structure tolerates approximation in commitment duration but does not tolerate its absence.

Why λ has a narrow optimum, and what the ablation adds. The alignment loss is a structural prior, not a task signal. Too small, it fails to shape latent dynamics; too large, it competes with the task gradient and reduces representational flexibility. The narrow optimum at $\lambda \approx 0.05$ is consistent with auxiliary objectives being most effective as soft regularizers. The past-sweet-spot ablation (Section 5) refines this: the conflict past the optimum is not generic auxiliary-loss interference but specifically the conflict introduced by *learned* directional structure that, when over-weighted, captures representational capacity the worker would otherwise use for the task.

7. Limitations and Conclusion

Gains are modest in absolute terms, with evaluation restricted to ARC-style tasks (held-out check on ConceptARC-mini); broader validation is needed. The past-sweet-spot ablation uses a single fixed seed (train-set LM loss at one step), though main-study replication variance (std $\approx$ 0.045) is much larger than the E vs. B gap, bounding single-seed variability well below the contrast magnitude. A representation-level analysis of subgoal-induced hidden-state geometry would directly probe the compositional substrate but requires late-training checkpoints we did not retain; we leave this for follow-up work. Subgoal *persistence*, controlled by P, is the design knob determining whether directional latent subgoals function in HRM; the narrow λ optimum and past-sweet-spot ablation together identify the stability–adaptivity tradeoff any compositional planner in latent space must resolve.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning, specifically the design of compact latent reasoning systems with explicit temporal abstraction. The experiments are conducted on abstract reasoning bench-

marks and do not involve human-subjects data, private data, or deployed decision systems. We do not identify immediate negative societal consequences beyond those generally associated with improved machine-learning reasoning systems.

References

Chollet, F. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*, 2019.

Evans, J. S. B. T. and Stanovich, K. E. Dual-process theories of higher cognition: Advancing the debate. *Perspectives on Psychological Science*, 8(3):223–241, 2013.

Graves, A. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.

Kahneman, D. *Thinking, Fast and Slow*. Farrar, Straus and Giroux, New York, 2011.

Moskvichev, A., Odouard, V. V., and Mitchell, M. The ConceptARC benchmark: Evaluating understanding and generalization in the ARC domain. *Transactions on Machine Learning Research*, 2023.

Sutton, R. S., Precup, D., and Singh, S. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1–2): 181–211, 1999.

Vezhnevets, A. S., Osindero, S., Schaul, T., Heess, N., Jaderberg, M., Silver, D., and Kavukcuoglu, K. FeUdal networks for hierarchical reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3540–3549. PMLR, 2017.

Wang, G., Li, J., Sun, Y., Chen, X., Liu, C., Wu, Y., Lu, M., Song, S., and Yadkori, Y. A. Hierarchical reasoning model. *arXiv preprint arXiv:2506.21734*, 2025.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 2022.