MINELOOP: INNER-TIME COMPUTATION FOR AUTOREGRESSIVE VISUAL WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Autoregressive visual world models turn each sampled token into evidence for later predictions. Standard feed-forward predictors sample each token after a single pass, without an explicit stage for refining uncertain predictions under the same history. We introduce MINELOOP for *inner-time world modeling*, giving each prediction an internal trajectory before it enters the world history. MINELOOP encodes the committed visual-action prefix as causal evidence, initializes a target-indexed future state, and recurrently updates it with a shared prediction core while the evidence remains fixed. Only the terminal state emits a visual token. On action-conditioned Minecraft rollout, MINELOOP improves FVD, LPIPS, SSIM, and PSNR against MineWorld controls at Small, Medium, and Large model tiers, with FVD reductions of 16.3%, 10.5%, and 7.3%, respectively. Counterfactual camera-action tests show stronger action alignment, and an inverse-dynamics evaluator recovers conditioning actions with comparable macro-F1. Across 64 held-out sources from the Small model, 23.13% of visual tokens change from wrong at K1 to correct at K4, while 0.093% change in the opposite direction. The advantage over a standard autoregressive model is concentrated in motion and occlusion regions. Continuing beyond the trained terminal step raises confidence while reducing accuracy, showing that MINELOOP learns a finite-depth prediction program with a specific terminal state. These results identify token commitment as a productive locus for recurrent computation in visual world models.

1 INTRODUCTION

Learned world models support planning (Sutton, 1991; Kaiser et al., 2020; Schrittwieser et al., 2020) and latent imagination (Ha & Schmidhuber, 2018; Hafner et al., 2019; 2020; 2021; 2023). Transformer dynamics (Chen et al., 2022; Robine et al., 2023) and discrete visual tokenizers (van den Oord et al., 2017; Esser et al., 2021; Yan et al., 2021) enable autoregressive visual generation (Ramesh et al., 2021; Sun et al., 2024; Tian et al., 2024) and action-conditioned simulation (Micheli et al., 2023; Hu et al., 2023; Bruce et al., 2024; Guo et al., 2025). In a standard token-autoregressive simulator, each visual token is sampled from a feed-forward prediction and appended to the causal prefix, becoming evidence for subsequent predictions.

This immediate commitment matters because visual futures are spatially heterogeneous. Static texture can often be continued from local evidence. Motion boundaries, newly revealed regions, and action-dependent changes require information from different locations and times. Under ordinary autoregression, both cases receive one feed-forward prediction before commitment. Any error then acquires a second role: it is an incorrect output and new evidence for later tokens. Increasing backbone depth improves the function used at each step, but it leaves this coupling intact. The world still advances at the instant the next distribution is read out.

Recent research on recursive self-improvement (RSI) places iterative refinement within a broader effort to build self-improving AI systems (Li et al., 2026; Chen et al., 2026). Recurrent-depth models implement refinement within inference by repeatedly updating latent states (Geiping et al., 2025). In autoregressive world models, the timing of this refinement matters: each sampled visual token becomes evidence for subsequent predictions.

We propose *inner-time world modeling*. The central object is an uncommitted future state that can change while the modeled world remains fixed. This introduces two clocks. *World time* advances

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

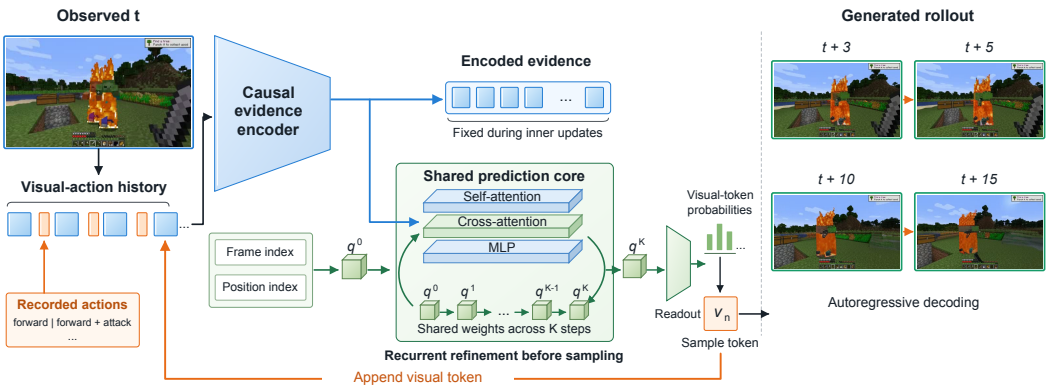


Figure 1: **MINELOOP refines each visual-token prediction before sampling.** Left: an observed frame and recorded actions form the visual-action history. Center: the causal encoder supplies evidence that stays fixed across the recurrent updates. Frame and position embeddings initialize q^0; recurrent applications of the shared prediction core produce q^K, which is read out to sample a visual token. Appending this token updates the history for the next prediction. Right: actual frames at $t+3$, $t+5$, $t+10$, and $t+15$ from a held-out Small-model rollout conditioned on one observed frame and the recorded control stream.

when a visual token is sampled and appended to history. *Inner time* advances when the model updates its current future hypothesis under the same causal evidence. The separation turns prediction formation into an explicit dynamical process. A model can retrieve evidence, coordinate target positions, and reshape its future representation before that representation has consequences for the rollout.

MINELOOP instantiates this principle. It first encodes the interleaved visual-action prefix with a causal Transformer. Position and frame embeddings initialize future-state queries without exposing target tokens. A shared prediction cell then performs causal query self-attention, cross-attention to the fixed evidence, and a nonlinear state update. We apply this cell recurrently and supervise only the terminal state. The resulting visual token is sampled and appended to the prefix, at which point world time advances. Parameter sharing makes the recurrent computation a learned prediction program whose same transition is refined across inner time.

MINELOOP improves all four rollout metrics against MineWorld controls at Small, Medium, and Large model tiers, reducing FVD by 16.3%, 10.5%, and 7.3%. The Small and Medium comparisons match estimated training compute; the Large comparison holds the 20-layer backbone fixed. Under counterfactual camera actions, MINELOOP follows the action direction more accurately and reduces displacement error at all three tiers. Across 64 held-out sources in the Small-tier analysis, 23.13% of visual tokens change from wrong at K1 to correct at K4, while 0.093% change in the opposite direction. Relative to the 25-layer MineWorld control, the K4 negative log-likelihood gain is 0.427 nat in motion regions and 0.429 nat in occlusion regions, compared with 0.073 nat on static regions. A fifth application of the shared cell increases top-1 confidence from 0.226 to 0.301 while reducing accuracy from 23.43% to 14.22%. Inner time learns a terminal computation state, with confidence continuing past the accurate endpoint.

Our contributions are threefold:

- We introduce MINELOOP for *inner-time world modeling*, enabling iterative refinement of visual-token predictions under fixed historical evidence before sampling and incorporation into the autoregressive history.

- MINELOOP improves all four rollout metrics across Small, Medium, and Large tiers and strengthens counterfactual camera-action alignment. At Small and Medium, these gains are achieved with fewer parameters under matched estimated training FLOPs.

- Token-level analyses show that inner-time computation repairs initially incorrect predictions, with gains concentrated in motion and occlusion regions. Extending computation beyond the trained endpoint raises confidence but reduces accuracy.

2 Related Work

Autoregressive visual world models. Learned world models have progressed from recurrent latent-state transitions to generative prediction in visual token space (Ha & Schmidhuber, 2018; Hafner et al., 2019; Micheli et al., 2023). IRIS and Δ-IRIS use autoregressive Transformers over compact visual tokens (Micheli et al., 2023; 2024). GAIA-1 and Genie scale tokenized video dynamics to driving and interactive environments (Hu et al., 2023; Bruce et al., 2024). MineWorld interleaves visual and action tokens in a causal sequence and generates controllable Minecraft futures (Guo et al., 2025). Diffusion world models provide a complementary route to visual fidelity (Alonso et al., 2024). These systems establish visual generation as a world-model interface. MineLoop focuses on the computation immediately before a causal visual token joins that interface: it retains the autoregressive factorization while giving the next future state an internal trajectory.

Recurrent depth and latent computation. Shared and adaptive computation treat depth as a modeling resource. ACT and PonderNet learn how long to update a prediction (Graves, 2016; Banino et al., 2021); Universal Transformers tie Transformer blocks across depth (Dehghani et al., 2019); deep equilibrium models read from the fixed point of a shared transformation (Bai et al., 2019). Recurrent-depth language models scale this principle to latent test-time computation (Geiping et al., 2025), while visual generators and latent world models adopt looped computation (Goyal et al., 2026; Lu et al., 2026). MineLoop assigns recurrence a specific role in token-autoregressive simulation: committed evidence is stable, the target state evolves, and terminal readout advances the outer world process.

Future-state prediction and iterative visual construction. Predictive representation learning and video forecasting build latent states for missing or future content (van den Oord et al., 2018; Lotter et al., 2017; Wang et al., 2017; Gao et al., 2022; Voleti et al., 2022). Joint-embedding architectures make the target query explicit (Assran et al., 2023; 2025), and action-conditioned video models gather future evidence from observations and controls (Ali et al., 2025). Future-aware targets improve visual autoregressive training (Yu et al., 2026); masked and self-refining generators construct visual hypotheses over multiple updates (Chang et al., 2022; Gupta et al., 2023; Wang et al., 2025). MineLoop brings these ideas into one causal transition. An explicit future state develops under the committed visual-action history, intermediate states remain latent, and the terminal state becomes the next visual token.

3 MineLoop

3.1 Sequence and inner time

Let $\mathbf{v}_t = (v_{t,1}, \ldots, v_{t,N_v})$ denote the visual tokens of frame t and $\mathbf{a}_t$ the subsequent action tokens. A trajectory is represented as

$$\mathbf{s} = (\mathbf{v}_0, \mathbf{a}_0, \mathbf{v}_1, \mathbf{a}_1, \ldots). \tag{1}$$

For visual target position n in this sequence, MineLoop preserves the autoregressive factorization

$$p(\mathbf{v}_{1:T} \mid \mathbf{v}_0, \mathbf{a}_{0:T-1}) = \prod_n p(v_n \mid \mathbf{s}_{<n}). \tag{2}$$

Each conditional uses K shared prediction updates under fixed evidence from $\mathbf{s}_{<n}$ (Figure 1). Inner time indexes these updates; world time advances when the terminal prediction is sampled and appended to the sequence.

3.2 Causal visual-action evidence

The input follows Equation 1. Each frame contributes $N_v = 336$ visual tokens on a 14×24 grid, followed by $N_a = 11$ action tokens. A LLaMA-style causal Transformer (Touvron et al., 2023) maps

the sequence to

$$\mathbf{E} = \mathrm{Enc}_\theta(\mathbf{s}) \in \mathbb{R}^{L\times d}. \tag{3}$$

The evidence sequence contains visual appearance, previous generated states, and known controls in their original temporal order. At training time, all frames can be processed in one pass. A target-specific prefix mask enforces that a future query at sequence position n can access only evidence positions $j < n$. At generation time, the evidence cache grows when a visual or action token is committed.

3.3 Target-indexed future states

Every predicted visual position has a spatial index i and frame index t. We initialize its future state with learned embeddings:

$$\mathbf{q}^0_{t,i} = \mathrm{RMSNorm}\Big(\mathbf{p}^{\mathrm{image}}_i + \mathbf{p}^{\mathrm{frame}}_t\Big). \tag{4}$$

The query specifies the target position and frame without embedding the target visual code.

During parallel training, the queries for all future visual positions form $\mathbf{Q}^0$. Their self-attention mask is lower triangular in the flattened visual order. A query can therefore coordinate with earlier future positions while remaining independent of later targets. During rollout, this interaction is maintained through a depth-specific cache of previously committed prediction states.

3.4 Shared recurrent prediction cell

The same cell D_ϕ is applied at each inner step. For state matrix $\mathbf{Q}^k$ and fixed evidence $\mathbf{E}$, one application is

$$\mathbf{H}^k = \mathbf{Q}^k + \mathrm{SelfAttn}_\phi(\mathrm{Norm}(\mathbf{Q}^k); \mathbf{M}_q), \tag{5}$$

$$\mathbf{G}^k = \mathbf{H}^k + \mathrm{CrossAttn}_\phi(\mathrm{Norm}(\mathbf{H}^k), \mathrm{Norm}(\mathbf{E}); \mathbf{M}_e), \tag{6}$$

$$\widetilde{\mathbf{Q}}^{k+1} = \mathbf{G}^k + \mathrm{MLP}_\phi(\mathrm{Norm}(\mathbf{G}^k)). \tag{7}$$

Here $\mathbf{M}_q$ is the causal query mask and $\mathbf{M}_e$ is the target-prefix evidence mask. Self-attention connects earlier future positions, while cross-attention retrieves visual and action evidence.

The recurrent update is damped and RMS-normalized:

$$\mathbf{Q}^{k+1} = \mathrm{RMSNorm}\Big((1-\beta)\mathbf{Q}^k + \beta\widetilde{\mathbf{Q}}^{k+1}\Big), \qquad \beta = 0.25. \tag{8}$$

3.5 Terminal supervision

For a training depth K, we apply the visual-token head only to the terminal state:

$$\mathcal{L}_{\mathrm{MineLoop}} = -\sum_{(t,i)\in\mathcal{T}} \log p_\Theta\big(v_{t,i} \mid \mathbf{Q}^K_{t,i}, \mathbf{s}_{<t,i}\big). \tag{9}$$

Gradients propagate through the complete recurrent trajectory. Intermediate states $\mathbf{Q}^1$ through $\mathbf{Q}^{K-1}$ can specialize to evidence organization for the terminal prediction.

3.6 Causal execution

For visual target position n, the recurrent computation follows

$$\mathbf{q}^{k+1}_n = D_\phi\big(\mathbf{q}^k_{\le n}, \mathbf{E}_{<n}\big), \qquad p(v_n \mid \mathbf{s}_{<n}) = \mathrm{softmax}(W\mathbf{q}^K_n). \tag{10}$$

The query state specifies the target frame and spatial position, while $\mathbf{E}_{<n}$ contains the committed visual-action history. During training, all target queries are updated in parallel at each inner step. The causal query mask exposes only earlier target states, and the evidence mask exposes only prefix evidence.

During rollout, the prediction cache is organized by both visual position and recurrent depth. The self-attention update for $\mathbf{q}^k_n$ reads the cached states $\mathbf{q}^k_j$ of earlier visual positions $j < n$, together

with the current state. Each depth retains its own version of the earlier queries. The evidence cache follows the interleaved sequence in Equation 1 and supplies a common history to every inner step. In cached decoding, the cross-attention keys and values of this history are computed once and reused across the recurrent updates. The changing query determines the attention weights at each step.

After the terminal state is read out, the sampled code extends the encoded evidence for the next visual position. At the end of a frame, the recorded action tokens enter the sequence before generation of the following frame begins. The first query in that frame therefore conditions on the completed visual frame and its associated action; later queries also observe the visual tokens already generated within the new frame. Each query starts from its own frame and spatial embeddings and keeps the accessible evidence fixed during its inner updates.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Data and training. We train MINELOOP and MineWorld on the same action-labelled Minecraft corpus, using MineWorld's visual and action representation (Guo et al., 2025). Each training example contains 16 frame–action pairs at a frame resolution of 224×384. Both models use the same frozen tokenizer with 8,192 visual codes and learn by next-token cross-entropy under ground-truth histories. We compare Small, Medium, and Large model tiers and use $K = 4$ for MINELOOP training and the main evaluations. Small and Medium use deeper MineWorld controls selected to match MINELOOP's estimated training FLOPs. At Large, both models use the 20-layer, 2048-width backbone. Table 1 reports parameter counts; Appendix A lists all configurations and decode FLOPs.

Evaluation. Each model generates 15 future frames from one observed frame on the same 256 held-out trajectories, with identical action sequences and sampling seeds. The initial prompt is $[\mathbf{v}_0, \mathbf{a}_0]$, where $\mathbf{a}_0$ controls the first generated transition. We use nucleus sampling with $p = 0.8$. We report Fréchet Video Distance (FVD) (Unterthiner et al., 2018), Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018), structural similarity (SSIM) (Wang et al., 2004), and peak signal-to-noise ratio (PSNR). FVD is averaged over generated prefixes of lengths 10 through 15; LPIPS, SSIM, and PSNR are averaged over paired generated and target frames.

4.2 ACTION-CONDITIONED ROLLOUT

Table 1: Action-conditioned 15-frame rollout from one observed frame. Small and Medium MineWorld rows are training-FLOP-matched feed-forward controls; Large retains the same 20-layer backbone in both models. Params gives each model's parameter count. Every comparison uses the same 256 held-out initial states, action sequences, and sampling seeds. MINELOOP improves distributional, perceptual, structural, and pixel-level quality at all three tiers.

Tier	Model	Params	FVD ↓	LPIPS ↓	SSIM ↑	PSNR ↑
Small	MineWorld	397.1M	1040.707	0.31330	0.46429	18.5601
	MINELOOP	339.2M	**870.606**	**0.30169**	**0.46849**	**18.7570**
Medium	MineWorld	732.5M	985.765	0.33280	0.44885	18.1193
	MINELOOP	641.0M	**882.519**	**0.30601**	**0.46754**	**18.5847**
Large	MineWorld	1,229.3M	938.000	0.32150	0.45650	18.4200
	MINELOOP	1,299.2M	**869.500**	**0.30150**	**0.47100**	**18.7200**

MINELOOP improves both video-distribution quality and agreement with the recorded future at every tier (Table 1). FVD decreases by 16.34%, 10.47%, and 7.30% at Small, Medium, and Large, respectively. The Small and Medium comparisons match training compute; the Large comparison holds the 20-layer backbone fixed. The corresponding frames also have lower perceptual error and higher structural and pixel-level similarity at every tier. At Medium, LPIPS decreases by 8.0% and PSNR increases by 0.465 dB. The gains therefore extend from aggregate video statistics to the appearance and structure of the paired future frames.

This quality advantage comes with fewer parameters. At Small and Medium, MINELOOP uses 14.6% and 12.5% fewer parameters than the deeper MineWorld controls. The cached decode costs are also closely matched at these tiers (Appendix A, Table 5). Reusing prediction weights thus provides an effective allocation of the computation budget: the model improves its generated trajectories while storing fewer learned weights.

4.3 ACTION RESPONSE UNDER HELD-OUT CONTROLS

Table 2: Action response under the 1+15 rollout protocol on 16 held-out sources. Counterfactual pairs flip horizontal camera actions; an independent inverse-dynamics model (IDM) recovers conditioning actions from generated frames.

		Counterfactual camera action			Inverse dynamics	
Tier	Model	Direction acc. ↑	L1-to-expected ↓	Off-axis leak ↓	Macro-F1 ↑	Camera-bin L1 ↓
Small	MineWorld	0.7958	15.7366	5.3134	0.8234	**0.6250**
	MINELOOP	**0.8282**	**11.0391**	**3.9505**	**0.8301**	0.7021
Medium	MineWorld	0.7615	23.7878	10.8821	0.8364	**0.6854**
	MINELOOP	**0.8583**	**9.9049**	**6.7199**	**0.8380**	0.6896
Large	MineWorld	0.8076	20.1847	8.7624	0.8443	**0.6817**
	MINELOOP	**0.8642**	**10.2673**	**6.3187**	**0.8461**	0.6908

The visual gains accompany a stronger response to changed camera controls. Counterfactual pairs keep the first frame and sampling seed fixed and flip only horizontal camera actions. Direction accuracy and displacement error measure whether the requested change appears in the generated motion; off-axis leak measures unintended motion perpendicular to it. MINELOOP improves all three measures at every tier (Table 2). At Medium, direction accuracy rises by 9.68 percentage points and displacement error falls by 58.4%. The independent inverse-dynamics evaluator complements this intervention by recovering actions from the generated trajectory. Discrete actions remain comparably recoverable, with slightly higher macro-F1 for MINELOOP, while MineWorld retains lower camera-bin L1. The clearest control gain is in how the generated motion responds to a deliberate change in the conditioning action.

4.4 CONDITIONAL PREDICTION UNDER TRUE HISTORY

We measure visual-token NLL and top-1 accuracy under ground-truth histories to isolate conditional prediction quality. Short, long, and extra-long use one, two, and three observed context frames, respectively.

Table 3: Teacher-forced visual-token prediction on 16 held-out trajectories. MineWorld rows are training-FLOP-matched feed-forward controls. True preceding frames isolate the quality of the conditional predictor from rollout error accumulation.

		Short		Long		Extra-long	
Tier	Model	NLL ↓	Acc. ↑	NLL ↓	Acc. ↑	NLL ↓	Acc. ↑
Small	MineWorld	3.0257	0.2874	2.9427	0.3047	2.9601	0.3067
	MINELOOP	**3.0004**	**0.2896**	**2.9130**	**0.3078**	**2.9301**	**0.3100**
Medium	MineWorld	2.9540	0.2957	2.8686	0.3120	2.8832	0.3145
	MINELOOP	**2.8961**	**0.3002**	**2.8106**	**0.3183**	**2.8252**	**0.3217**
Large	MineWorld	2.8913	0.3030	2.8038	0.3184	2.8160	0.3213
	MINELOOP	**2.8392**	**0.3081**	**2.7516**	**0.3256**	**2.7638**	**0.3296**

MINELOOP lowers NLL and raises accuracy at every context length and all three tiers (Table 3). Its advantage persists as additional observed frames become available: the NLL reduction remains approximately 0.058 nat at Medium and 0.052 nat at Large across contexts. NLL rewards probability assigned to the target code even when the top-ranked code is unchanged; the accompanying accuracy

gains show that improvements also reach the predicted token identities. Together with the rollout results, this establishes gains under both true and generated histories. The conditional predictor assigns greater probability to the recorded continuation, and its sampled trajectories improve on all four visual metrics.

5 What Happens During Inner Time?

Pre-commitment recurrence improves rollout by changing the token trajectory before commitment. The analysis below traces that trajectory through five measurements: token fate, spatial location of the gain, token-level depth demand, confidence and target margin, and evidence retrieval. We use the Small pair because it gives the largest rollout gain and permits exact per-step tracing at visual-token resolution.

5.1 Analysis protocol

We analyze 64 held-out sources. At every target, we record logits, correctness, confidence, and recurrent state from K1 through K5; intervals use a source-level bootstrap. The comparison model is the 25-layer MineWorld training-FLOP-matched control (397.1M versus 339.2M parameters). Under the same true history, MineLoop K4 lowers NLL by 0.246 nat (95% CI [0.116, 0.391]) and raises accuracy by 0.0362 (95% CI [0.0154, 0.0615]).

5.2 What changes: inner time rescues wrong predictions

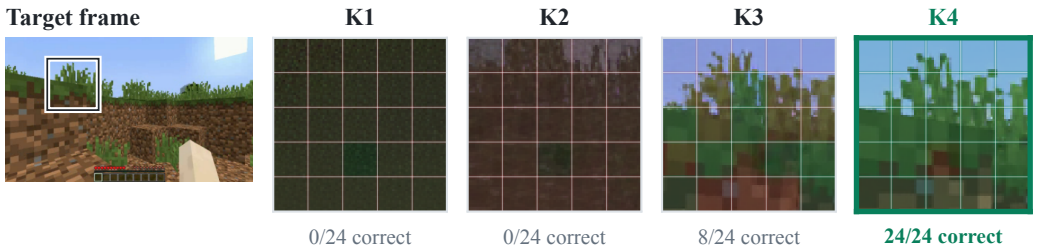


Figure 2: **The same initially wrong tokens are repaired across inner time.** The same target region is decoded at K1–K4. Outlined cells are the cohort that is wrong at K1 and correct at K4; green cells have reached the target visual code. Counts below each crop track this fixed cohort. K4 is the trained readout. The example is selected by the deterministic rule in Appendix B; aggregate fate statistics use all 64 held-out sources.

Across the 64 analysis sources, 23.132% of visual tokens move from wrong at K1 to right at K4, whereas 0.093% move in the opposite direction: roughly 249 rescues per damaged token. Figure 2 follows individual rescues across depth; Appendix B, Table 6, reports source-bootstrap intervals.

5.3 Where it helps: gains concentrate at dynamic uncertainty

Aligning the paired NLL gain with spatial masks locates the largest improvements in motion and occlusion regions, at about six times the static-region gain (Figure 3). The advantage concentrates where motion and visibility changes make the next frame uncertain. Appendix B, Table 8, gives the numerical estimates and intervals.

5.4 Prediction depth, margin, and confidence

Correct predictions form at different depths within the same frame (Figure 4A). All positions receive the same update budget; stable resolution depth records when a position first becomes correct and stays correct through K4. Requiring correctness to persist distinguishes a settled prediction from a temporary match to the target code. Some predictions settle early, while 71.2% of the tokens correct at K4 first stabilize at the final update. The prediction head is supervised at K4, and its intermediate

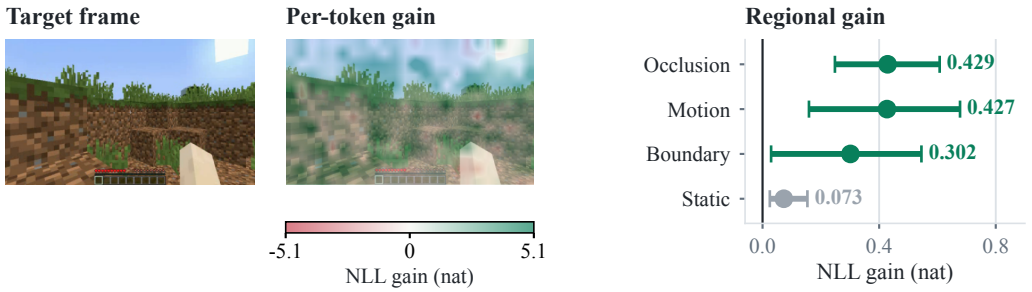


Figure 3: **MineLoop's advantage is spatially concentrated.** Left: the target frame. Middle: per-token MineWorld NLL minus MINELOOP K4 NLL, restored to the visual-token grid. Green denotes a MINELOOP advantage. Right: estimates over motion, motion-boundary, occlusion, and static regions from 64 sources; bars show 95% source-bootstrap intervals.

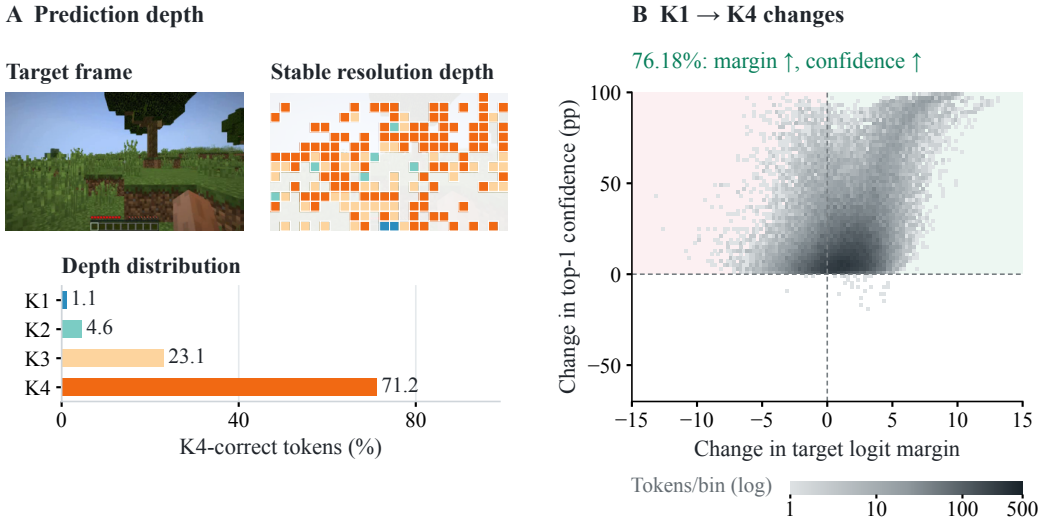


Figure 4: **Prediction depth varies spatially, and most tokens gain margin and confidence. A**: for tokens correct at K4, stable resolution depth is the first step after which the prediction stays correct through K4. Map and bar colors identify the same depths; the distribution aggregates K4-correct tokens across 64 held-out sources. **B**: paired K1-to-K4 changes aggregated across the same 64 sources. Target margin is the target logit minus the largest competing logit. Cell intensity shows log-scaled token counts; pp denotes percentage points.

readouts expose how the learned computation prepares that final output. These depth labels describe the evolution of token choices under a common recurrent schedule.

Target margin and confidence describe how these updates change the distribution (Figure 4B). Margin measures the target code's advantage over its strongest competitor; confidence measures probability on the current top-ranked code. Across the 64 analysis sources, 76.18% of visual tokens increase both quantities from K1 to K4. For this majority, recurrence strengthens the target relative to competing codes while making the top prediction more decisive. Stable depth identifies when a correct token choice persists, while margin reveals changes in target competitiveness even before the top-ranked identity changes. The two views connect the timing of correct predictions with the direction of the underlying distributional update.

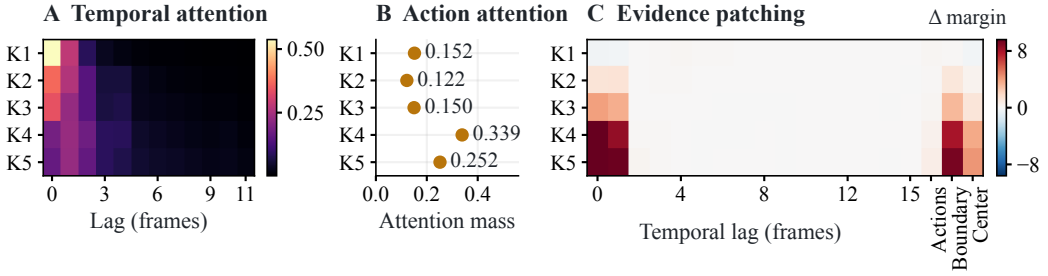


Figure 5: **Later inner steps retrieve and combine different evidence. A**: cross-attention mass by temporal lag. **B**: attention on action tokens. **C**: matched-donor patching, with columns for temporal lags 0–15, actions, spatial boundaries, and spatial centers. Color denotes clean minus patched target margin. Temporal and boundary evidence have their strongest effects near K4.

5.5 Evidence retrieval shifts with inner depth

The evolving target state changes how the model reads a fixed history. Each recurrent update produces a new cross-attention query, while the evidence keys and values remain fixed for that target. Attention can therefore shift across frames and actions as the prediction develops. Figure 5 traces these reads and tests their effect on the target distribution.

From K1 to K4, mean evidence lag grows from 0.983 to 3.154 frames. Attention on the two most recent frames falls from 0.818 to 0.417, while mass at lag four or greater rises from 0.063 to 0.318. Action-token attention reaches 0.339 at K4, compared with 0.152 at K1. The later states distribute their reads across a wider temporal range and place more weight on the control signal that conditions the future.

Matched-donor patching tests whether specific evidence groups affect the prediction. For the traced target positions, we replace one group's cross-attention value contribution while holding the query state and all other groups fixed. The clean-minus-patched target margin measures how much this substitution weakens the target relative to its strongest competitor. Recent temporal evidence and spatial-boundary evidence have their largest effects near K4. Broader attention and sensitivity to recent evidence coexist: the final prediction reads a wider history while remaining strongly dependent on recent frames and spatial boundaries. The recurrent state thus controls both which evidence is retrieved and how that evidence influences the eventual token choice.

The five measurements give a single mechanism: inner time rescues initially wrong visual codes, concentrates its gain where the future is dynamically uncertain, allocates different effective depth across spatial positions, integrates a changing evidence horizon, and reaches a learned terminal state for readout. A fixed causal history can support a stronger next-token distribution before the world state advances.

6 Conclusion

MineLoop improves all four rollout metrics against MineWorld controls at Small, Medium, and Large model tiers. Small and Medium match estimated training compute; Large keeps the 20-layer backbone fixed. MineLoop realizes inner-time world modeling through a shared recurrent predictor and terminal-only readout: causal history stays fixed while an uncommitted future state evolves. Its internal trajectory repairs initially wrong tokens, with gains concentrated in motion and occlusion regions. An additional update beyond the trained readout raises confidence while reducing accuracy. World time determines what has happened; inner time determines how the next state is formed.

Reproducibility statement

The architecture, causal masks, recurrent update, and evaluation protocol are specified in Sections 3 and 4. Appendix A records model configurations, optimization, sampling, and source-level statistics. Code, model identifiers, and evaluation metadata will be released with the submission.

References

Muhammad Qasim Ali, Aditya Sridhar, Shahbuland Matiana, Alex Wong, and Mohammad Al-Sharman. Humanoid world models: Open world foundation models for humanoid robotics. *arXiv preprint arXiv:2506.01182*, 2025.

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. In *Advances in Neural Information Processing Systems*, 2024.

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

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. Deep equilibrium models. In *Advances in Neural Information Processing Systems*, 2019.

Andrea Banino, Jan Balaguer, and Charles Blundell. Pondernet: Learning to ponder. *arXiv preprint arXiv:2107.05407*, 2021.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4603–4623. PMLR, 2024. URL `https://proceedings.mlr.press/v235/bruce24a.html`.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. MaskGIT: Masked generative image transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Chang Chen, Yi-Fu Wu, Jaesik Yoon, and Sungjin Ahn. Transdreamer: Reinforcement learning with transformer world models. *arXiv preprint arXiv:2202.09481*, 2022.

Mingguang Chen, Licheng Wang, and Bo Qu. Recursive self-improvement in AI: From bounded self-refinement to autonomous research loops. *arXiv preprint arXiv:2607.07663*, 2026.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. In *International Conference on Learning Representations*, 2019.

Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Zhangyang Gao, Cheng Tan, Lirong Wu, and Stan Z. Li. SimVP: Simpler yet better video prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach. In *Advances in Neural Information Processing Systems*, volume 38, pp. 41340–41391. Curran Associates, Inc., 2025. doi: 10.52202/085713-1380.

Sahil Goyal, Swayam Agrawal, Gautham Govind Anil, Prateek Jain, Sujoy Paul, and Aditya Kusupati. ELT: Elastic looped transformers for visual generation. In *Computer Vision – ECCV 2026*, volume 17056 of *Lecture Notes in Computer Science*, pp. 308–326. Springer, 2026. doi: 10.1007/978-3-032-37324-3_17.

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

Junliang Guo, Yang Ye, Tianyu He, Haoyu Wu, Yushu Jiang, Tim Pearce, and Jiang Bian. Mineworld: A real-time and open-source interactive world model on minecraft. *arXiv preprint arXiv:2504.08388*, 2025.

Agrim Gupta, Stephen Tian, Yunzhi Zhang, Jiajun Wu, Roberto Martín-Martín, and Li Fei-Fei. MaskViT: Masked visual pre-training for video prediction. In *International Conference on Learning Representations*, 2023.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, 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 domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. GAIA-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H. Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, Afroz Mohiuddin, Ryan Sepassi, George Tucker, and Henry Michalewski. Model-based reinforcement learning for atari. In *International Conference on Learning Representations*, 2020.

Hanjing Li, Qiguang Chen, Chenyuan Zhang, Qionglin Qiu, Fanqing Meng, Mengkang Hu, Libo Qin, and Min Zhang. Towards AI that improves itself: A survey of recursive self-improvement, 2026. URL `https://lee1003-lee.github.io/Awesome-RSI-Research/`. Preprint.

William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. In *International Conference on Learning Representations*, 2017.

Hongyuan Adam Lu, Z. L. Victor Wei, Qun Zhang, Jinrui Zeng, Bowen Cao, Lingwei Meng, Mocheng Li, et al. Looped world models. *arXiv preprint arXiv:2606.18208*, 2026.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *International Conference on Learning Representations*, 2023.

Vincent Micheli, Eloi Alonso, and François Fleuret. Efficient world models with context-aware tokenization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 35623–35638. PMLR, 2024. URL `https://proceedings.mlr.press/v235/micheli24a.html`.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, 2021.

Jan Robine, Marc Höftmann, Tobias Uelwer, and Stefan Harmeling. Transformer-based world models are happy with 100k interactions. In *International Conference on Learning Representations*, 2023.

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.

Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024.

Richard S. Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM SIGART Bulletin*, 2(4):160–163, 1991.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *Advances in Neural Information Processing Systems*, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric and challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, 2017.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Vikram Voleti, Alexia Jolicoeur-Martineau, and Christopher Pal. MCVD: Masked conditional video diffusion for prediction, generation, and interpolation. In *Advances in Neural Information Processing Systems*, 2022.

Jiamian Wang, Ziqi Zhou, Chaithanya Kumar Mummadi, Sohail Dianat, Majid Rabbani, Raghuveer Rao, Chen Qiu, and Zhiqiang Tao. Visual self-refinement for autoregressive models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 21291–21300. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.1161.

Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S. Yu. PredRNN: Recurrent neural networks for predictive learning using spatiotemporal LSTMs. In *Advances in Neural Information Processing Systems*, 2017.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.

Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. VideoGPT: Video generation using VQ-VAE and transformers. *arXiv preprint arXiv:2104.10157*, 2021.

Yonghao Yu, Lang Huang, Zerun Wang, Runyi Li, and Toshihiko Yamasaki. Mirai: Autoregressive visual generation needs foresight. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 30511–30520, 2026. URL `https://openaccess.thecvf.com/content/CVPR2026/html/Yu_Mirai_Autoregressive_Visual_Generation_Needs_Foresight_CVPR_2026_paper.html`.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

A Implementation and Evaluation Details

A.1 Model configurations

Training. We train MineLoop at all three tiers with AdamW, using $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-8}$, and weight decay 0.1. The learning rate increases linearly to 3×10^{-4} over 300 warmup steps and then follows cosine decay to zero. A global batch contains 96 sequences of 16 frame–action pairs; 56,310 optimizer steps correspond to approximately 30 billion interleaved visual-action tokens. Training uses bfloat16 mixed precision and gradient-norm clipping at 1.0.

Table 4: Model configurations. Small and Medium MineWorld controls are deeper feed-forward baselines selected to match MINELOOP's estimated training FLOPs. Large retains the same 20-layer backbone in both models. The recurrent predictor shares its parameters across all four inner steps.

Tier	Model	Backbone layers	Width	Heads	Parameters	Inner steps	Comparison
Small	MineWorld	25	1024	16	397.1M	–	training-FLOP matched
	MINELOOP	20	1024	16	339.2M	4	training-FLOP matched
Medium	MineWorld	29	1536	24	732.5M	–	training-FLOP matched
	MINELOOP	24	1536	24	641.0M	4	training-FLOP matched
Large	MineWorld	20	2048	32	1,229.3M	–	same 20-layer backbone
	MINELOOP	20	2048	32	1,299.2M	4	same 20-layer backbone

A.2 SEQUENCE LAYOUT AND CAUSAL MASKS

Each 224×384 frame is encoded as 336 visual tokens arranged on a 14×24 grid. Eleven action tokens follow each frame, so one frame-action block has 347 sequence positions. Sixteen training frames occupy at most 5,552 positions. Both training and inference use the same order $[\mathbf{v}_0, \mathbf{a}_0, \mathbf{v}_1, \mathbf{a}_1, \ldots]$. For the 1+15 rollout protocol, the initial prompt is $[\mathbf{v}_0, \mathbf{a}_0]$, where $\mathbf{a}_0$ causes the transition to the first generated frame.

The evidence encoder uses ordinary causal attention. The query self-attention mask is lower triangular in flattened visual order. Its cross-attention mask permits target position n to read evidence position j exactly when $j < n$. The frame-index embedding is derived from the target frame, which keeps cached rollout aligned with the target indexing used in training.

A.3 RECURRENT STATE UPDATE

The prediction cell contains query self-attention, cross-attention to the evidence stream, and a gated MLP, each with pre-normalization and a residual connection. The damped update in Equation 8 uses $\beta = 0.25$. Query initialization combines a learned image-grid position with a learned target-frame embedding. Weights are shared across K1–K4, and only K4 contributes cross-entropy loss. At inference, each emitted visual code extends both the evidence cache and the depth-indexed prediction cache.

A.4 ROLLOUT METRICS

All models use the same 256 held-out source clips and one ground-truth context frame. For each source, the action sequence and random seed are held fixed between MineWorld and MINELOOP. LPIPS, SSIM, and PSNR are averaged over paired generated frames. FVD is first computed for generated prefixes of 10, 11, 12, 13, 14, and 15 frames; the reported number is their arithmetic mean. This ties the headline statistic to sustained rollout across the generated horizon.

Table 5 reports cache-correct decode FLOPs for the 1+15 protocol. FLOPs count active causal attention, linear projections, MLPs, and the visual-token readout with multiply-add equal to two operations. All models process the same 5,540-position interleaved evidence stream; standard decoder depth follows Table 4, and MINELOOP additionally performs four recurrent updates for each of 5,040 generated visual tokens. Normalization, activations, softmax, sampling, and embedding lookup are excluded symmetrically.

Table 5: Cached decode FLOPs for one observed frame followed by 15 generated frames. FLOPs use active-prefix KV caches for both models and count each cached K/V projection once. Small and Medium MineWorld rows use training-FLOP-matched feed-forward controls; Large uses the same 20-layer backbone in both models.

Tier	Model	TFLOPs / 15f	FLOP ratio
Small	MineWorld	5.868	1.00×
	MINELOOP	5.866	1.00×
Medium	MineWorld	10.696	1.00×
	MINELOOP	10.733	1.00×
Large	MineWorld	15.930	1.00×
	MINELOOP	19.592	1.23×

A.5 ACTION-RESPONSE EVALUATION

The counterfactual action-pair evaluation uses 16 held-out sources with the 1+15 rollout protocol. Each pair keeps the initial frame, model, sampler, and random seed fixed, then flips the horizontal camera action. We estimate the expected visual displacement from the held-out action calibration and score active transitions where the action flip creates a measurable horizontal change. Direction accuracy measures whether the generated displacement follows the expected sign. L1-to-expected measures the absolute difference between generated and calibrated displacement magnitude. Off-axis leak measures motion on the orthogonal camera axis.

The inverse-dynamics evaluation applies an independent VPT inverse-dynamics model to adjacent generated frames and compares the recovered action labels with the conditioning actions. Macro-F1 averages the discrete action subtasks; camera-bin L1 measures the absolute difference between the recovered and conditioning camera bins. The reported values use 240 generated transitions.

B MECHANISM ANALYSIS DETAILS

B.1 TOKEN FATE DEFINITIONS

For target code y and inner-step distribution p_k, correctness is $\mathbb{1}[\arg\max p_k = y]$. A token is *rescued* when it is wrong at K1 and correct at K4. It is *damaged* when it is correct at K1 and wrong at K4. The K5 categories apply the same definitions to the K4–K5 transition. Stable-depth classes record the first step after which the top-1 prediction remains correct through K4. A token that becomes correct before K4 but is incorrect at K4 is marked unstable.

Disjoint calibration sources fix visualization ranges and numerical thresholds before the aggregate analysis. All percentages and confidence intervals in the paper use the 64 held-out analysis sources. Bootstrap intervals draw sources with replacement and recompute the full statistic for each draw.

Table 6: Token fates from K1 to the trained K4 readout. Percentages aggregate visual-token transitions across 64 held-out sources; confidence intervals resample sources.

Transition	Share (%)	95% CI
Wrong at K1 → right at K4	**23.132**	[18.310, 28.067]
Right at K1 → wrong at K4	0.093	[0.050, 0.137]

B.2 CHANGES IN TARGET MARGIN AND CONFIDENCE

For logits $\ell^{(k)}$ and target code y, target margin is $m_k = \ell_y^{(k)} - \max_{j \neq y} \ell_j^{(k)}$, and top-1 confidence is $c_k = \max_j p_k(j)$. Figure 4B plots $(m_4 - m_1, 100(c_4 - c_1))$ at every visual-token position from the 64 analysis sources. The histogram uses bins of 0.25 logit units by two percentage points and a logarithmic count scale. All tokens fall within the displayed ranges. Quadrant percentages use strict inequalities and all visual-token positions as the denominator; points on either zero axis are

counted separately. Positive margin change measures increased target competitiveness, while top-1 correctness is evaluated from the predicted code.

Table 7 summarizes the K4–K5 changes. At the token level, 12.528% move from correct at K4 to wrong at K5, while 3.322% move in the opposite direction.

Table 7: The trained endpoint is a terminal computation state. Applying the shared cell once more increases top-1 confidence but degrades every target-aware measure.

Measure	K4 (trained)	K5 (extra)	K5 − K4
NLL ↓	**3.6565**	5.0832	+1.4267
Top-1 accuracy ↑	**0.2343**	0.1423	−0.0921
Target probability ↑	**0.1282**	0.0990	−0.0292
Top-1 confidence	0.2257	0.3009	+0.0753

B.3 Spatial masks

We reshape each token score to the 14×24 tokenizer grid. Motion masks are obtained by thresholding frame-to-frame visual displacement. Motion boundaries are morphological bands around that mask. Occlusion and newly revealed regions are identified by forward–backward inconsistency; static regions are positions outside these masks. Each region statistic first averages tokens within a source, then applies the source-level bootstrap.

Table 8: Where inner time helps. Positive values are the 25-layer MineWorld control NLL minus MineLoop K4 NLL; larger values indicate a stronger MineLoop advantage. Regions are computed from frame-space motion and visibility masks.

Region	NLL gain	95% CI
Motion	**0.427**	[0.159, 0.677]
Motion boundary	0.302	[0.029, 0.545]
Occlusion / newly revealed	**0.429**	[0.248, 0.608]
Static	0.073	[0.025, 0.154]

The visual example in Figure 2 is selected before rendering by a deterministic rule: among analysis-source windows outside the HUD, choose the textured 5×5 token region containing the most K1-wrong/K4-correct tokens, with mean K1-to-K4 NLL reduction as the first tie-breaker. The decoded panels are teacher-forced top-1 token maps; aggregate claims use the 64 analysis sources, and the displayed example visualizes one deterministic case. The frame in Figure 4A is selected independently by the largest difference between the 90th percentile and median K1-to-K4 NLL gain.

B.4 Evidence attribution and activation patching

For each recurrent step, we aggregate cross-attention by evidence-frame lag, where lag is the target-frame index minus the evidence-frame index. Action tokens form a separate evidence group. The attention traces use exact, non-fused attention across the 64 held-out analysis sources. To test whether attended groups can change the prediction, matched-donor patching replaces the cross-attention value contribution of one temporal or spatial evidence group while keeping the query state and all other groups fixed. Its effect is the clean target margin minus the patched target margin. The intervention uses fixed target positions from the same sources and the same donor assignment at every recurrent depth.

C AI Use Statement

Generative AI tools were used to polish the language and improve the readability of the manuscript. The authors take full responsibility for the content of the paper.