Iterative Computation as Anytime Forecasting: Dense Supervision for Calibrated Trajectories in Recurrent World Models

Anonymous Author(s) [1]

Abstract

Many modern neural forecasters *iterate*: world models roll out a learned transition, recursive Transformers refine a prediction over many cycles, looped language models "think longer" before answering. We collect these systems under a single abstraction – *Iterative Neural Computations* (INCs) – and identify the standard practice of supervising only the final iterate (*endpoint supervision*) as the shared cause of two failure modes that matter directly for forecasting: (i) gradients through long rollout chains are noisy and direction-corrupted, destabilizing long-horizon training; and (ii) intermediate iterates are unconstrained, ruling out anytime prediction and horizon extrapolation. We introduce *Dense Intermediate Consistency for Endpoints* (DICE), a model-agnostic training framework that supervises every iterate through a *shared* readout head. The change is purely in the loss, adds $< 5\%$ training compute, and turns any INC into an anytime forecaster whose intermediate states are valid, calibrated predictions. We further derive a probability-space stability bound linking DICE to a decision-theoretic stopping rule, Adaptive Stability Halting. Across three INC families, DICE delivers near-perfect horizon extrapolation on prefix sums, $+7.4$ pp on long-horizon maze planning, $+3.97$ pp on bAbI temporal reasoning, and a $6.6\times$ inference-time speedup with no accuracy loss.

1. Introduction

A growing fraction of modern neural forecasters do not produce a prediction in a single forward pass. They *iterate*: a world model rolls out its transition function over a horizon (Ha & Schmidhuber, 2018; Hafner et al., 2020); a looped or recursive Transformer refines a candidate answer over many cycles (Dehghani et al., 2019; Geiping et al., 2025; Zhu et al., 2025); a Deep Thinking network repeatedly applies a shared block (Schwarzschild et al., 2021). In each case the model produces a *trajectory* of states whose decoded values can be read as forecasts at increasing horizons.

We collect these architectures under one abstraction, *Iterative Neural Computations* (INCs), and ask: when trained with the standard recipe of *endpoint supervision* – loss only on the final iterate – what kind of forecasters do we get? We identify two coupled failure modes:

(F1) Long-horizon training instability. The gradient at iteration k traverses the Jacobian product $\prod_{j=k}^{K-1} \partial f_\theta / \partial \mathbf{z}^j$. Beyond classical norm effects (Pascanu et al., 2013), the *relative variance* of this product grows as $(1+\sigma^2/\mu^2)^{K-k}-1$ under random-Jacobian assumptions (Sec. C.1). We measure directly: under endpoint supervision, early-step gradients are *anti-aligned* with the endpoint signal (cosine $\in [-0.6, -0.2]$) with mini-batch variance up to 10^4 (Sec. E).

(F2) Trajectory miscalibration / horizon foreclosure. Endpoint supervision places *no* constraint on $\mathbf{z}^1, \ldots, \mathbf{z}^{K-1}$: any trajectory producing a correct $\mathbf{z}^K$ is equally rewarded. The model can learn a fixed-horizon strategy whose intermediate states do not decode to anything meaningful. Both *anytime* forecasting (at $k < K_{\text{train}}$) and *horizon extrapolation* (at $k > K_{\text{train}}$) are foreclosed.

We propose **DICE** (Dense Intermediate Consistency for Endpoints), a remedy in the loss alone: attach an auxiliary loss at every iterate, decoded through a head *shared* with the endpoint. Short gradient paths address (F1); the shared head forces every iterate to be a decodable forecast, addressing (F2). The trajectory becomes a sequence of progressively refined forecasts rather than opaque scratch states.

This buys three forecasting-relevant properties. *Calibrated trajectories*: a small DICE loss provably bounds the ℓ_1 drift between consecutive predictive distributions (Prop. 1), a trajectory-level calibration guarantee endpoint supervision cannot provide. *Anytime prediction*: every step is a valid forecast. *Decision-theoretic stopping*: *Adaptive Stability Halting* (ASH) terminates iteration once the forecast has converged, yielding $6.6\times$ speedup at no cost in accuracy.

[1]Anonymous Affiliation. Correspondence to: Anonymous Author(s) <anon.email@example.com>.

Accepted to the Forecasting as a New Frontier of Intelligence Workshop at ICML 2026, Seoul, South Korea. PMLR 306, 2026.

Contributions. (1) We frame iterative neural systems as a unified class of forecasters and pin endpoint supervision as the cause of long-horizon instability and trajectory miscalibration. (2) DICE: a drop-in training framework with a probability-space calibration guarantee. (3) Consistent gains across three INC families plus 6.6× speedup via ASH.

2. Iterative Computation as Sequential Forecasting

An INC is a triple $(h_\theta, f_\theta, g_\theta)$ with shared parameters: encoder $h_\theta : \mathcal{X} \to \mathcal{Z}$ produces $\mathbf{z}^0$; block $f_\theta : \mathcal{Z} \to \mathcal{Z}$ is applied K times, giving $\mathbf{z}^k = f_\theta(\mathbf{z}^{k-1})$; readout $g_\theta : \mathcal{Z} \to \mathcal{Y}$ decodes a state to a forecast $\hat{\mathbf{y}}_k = g_\theta(\mathbf{z}^k)$. The trajectory $(\mathbf{z}^0, \ldots, \mathbf{z}^K)$ is a rollout; k is the horizon. This covers Universal Transformers, Deep Thinking, looped/recursive Transformers, DEQs, and learned world models when $\mathcal{Y}$ is observation or next-state space (Sec. B).

Endpoint supervision uses the loss $\mathcal{L}_{\text{end}}(\theta) = \ell\big(g_\theta(\mathbf{z}^K), \mathbf{y}\big)$. Two structural consequences follow: (F1) long Jacobian chains amplify gradient noise exponentially in horizon (Sec. C.1); and (F2) intermediate iterates carry no constraint, so the model is free to be miscalibrated everywhere except at $k = K$.

3. DICE

3.1. Dense Supervision through a Shared Readout

$$\mathcal{L}_{\text{dense}} = \ell(g_\theta(\mathbf{z}^K), \mathbf{y}) + \alpha \sum_{k=1}^{K-1} w_k \, \ell(g_\theta(\mathbf{z}^k), \mathbf{y}), \quad (1)$$

with $\alpha > 0$, $\sum_k w_k = 1$ (uniform, linear $w_k \propto k$, or exponential $w_k \propto 2^k$); g_θ is the *same* head at every k. Two mechanisms: **(M1) Shortened gradients.** The auxiliary loss at step k backpropagates through only k applications of f_θ, replacing one long-range direction-corrupted signal with a dense field of locally faithful ones, addressing (F1). **(M2) Trajectory-as-forecasts.** Sharing g_θ forces every iterate to decode to a valid forecast, addressing (F2). The four-way ablation in Sec. F confirms both are necessary; this is what distinguishes DICE from classical deep supervision (Lee et al., 2015), whose separate per-layer heads do not enforce trajectory consistency.

3.2. Calibration Guarantee

Let $p_k(x) = \text{softmax}(g_\theta(\mathbf{z}^k(x)))$ and let $\lambda_K = 1$, $\lambda_k = \alpha w_k$ $(k < K)$ be the effective weights in Eq. (1).

Proposition 1 (Trajectory calibration; proof in Sec. C.2)**.** *If $\mathcal{L}_{\text{dense}}(\theta) \le \delta$ under cross-entropy, for every supervised k, $\mathbb{E}_x[\|p_k(x) - e_{\mathbf{y}(x)}\|_1] \le 2\delta/\lambda_k$, and consecutive forecasts satisfy $\mathbb{E}_x[\|p_{k+1}(x) - p_k(x)\|_1] \le 2\delta(1/\lambda_{k+1} + 1/\lambda_k)$.*

Table 1. **Deep Thinking on prefix sums.** Test-time extrapolation accuracy (%) at 40 iterations.

Test bits	Endpoint	GRPO	LSRL	RLTT	DICE
64	11.70	20.10	34.25	40.08	**100.00**
72	0.12	14.64	22.09	25.19	**100.00**
128	0.00	1.49	0.56	5.33	**100.00**
256	0.00	0.00	0.03	0.00	**11.88**

Table 2. **ASH on prefix sums.** It is a pure compute saving.

Bits	Acc	Avg iters
64	100.00	15.53
72	100.00	23.37
128	100.00	47.27
256	99.95	71.97

Endpoint supervision controls only p_K and allows arbitrary intermediate forecasts; it cannot give this bound.

3.3. Adaptive Stability Halting

Prop. 1 suggests halting when the predictive distribution stops moving: $\|\text{softmax}(\hat{\mathbf{y}}_k) - \text{softmax}(\hat{\mathbf{y}}_{k-1})\|_1 < \epsilon$ for m consecutive steps. ASH treats the rollout as a probability-space fixed-point iteration – a parameter-free, decision-theoretic compute-allocation rule.

4. Experiments

We evaluate DICE on three INC families; architecture, optimizer, and data are unchanged across baseline and DICE. Each setting probes a forecasting-relevant property: *horizon extrapolation* (prefix sums), *long-horizon planning* (maze), *multi-step temporal reasoning* (bAbI). Baselines: Endpoint, GRPO (Shao et al., 2024), LSRL (Ren, 2025), RLTT (Williams & Tureci, 2026); configs in Sec. D and K.

Horizon extrapolation (Deep Thinking, prefix sums). Trained on 32-bit prefix-sums, evaluated at 64/72/128/256 bits with 40 test iterations. Endpoint supervision achieves only 11.70% at 64 bits and collapses at longer horizons. DICE (linear schedule) reaches **100%** at 64/72/128 bits (Tab. 1); ASH halts in far fewer than 80 iterations on easier widths, giving compute savings "for free." The shared readout converts each step into an explicit incremental forecast, so additional steps continue the refinement.

Long-horizon planning (Recursive Transformer, mazes). On 30 × 30 mazes (Lehnert et al., 2024) (shortest path > 110 steps), the base model is the Tiny Recursive Model with $K = 12$. DICE reaches **81.0%** exact-match vs. 73.6% for endpoint supervision (+7.4 pp), and matches the baseline's best using 75% of the training budget (Tab. 3). ASH averages 2.43 outer-refinement steps with a marginal accu-

Table 3. **Recursive Transformer on maze solving.** *Top:* Exact accuracy (%) at each evaluation epoch. DICE produces dramatically faster learning and higher final accuracy. *Bottom:* Comparison of training strategies on exact-match accuracy.

Configuration	Ep. 1k	Ep. 2k	Ep. 3k	Ep. 4k
Endpoint	0.0	27.1	59.7	73.6
DICE, w=0.1	3.8	36.7	69.6	74.7
DICE, w=0.3	0.7	68.8	66.4	73.3
DICE, w=1.0	0.7	**69.7**	**78.9**	**81.0**

(a) Per-epoch evaluation.

Method	**Acc.**	**vs ep.**
Endpoint	73.6	–
GRPO	74.3	+0.7 pp
LSRL	74.8	+1.2 pp
RLTT	75.0	+1.4 pp
DICE	**81.0**	**+7.4 pp**

(b) Comparison vs. baselines.

racy *improvement* (81.05 vs. 80.96): 6.6× inference-time speedup at no cost in forecast quality (Tab. 4).

Table 4. **Adaptive halting on TRM (maze, epoch 4k).** DICE improves training-time accuracy over the endpoint baseline; applying prediction-stability halting on the outer N_{sup} refinement loop yields an additional 6.6× inference speedup with no accuracy loss.

Configuration	Outer steps	Exact acc.	Avg. steps	Speedup
Endpoint	16 (fixed)	73.63%	16.00	1.00×
DICE (w.o. adaptive halt)	16 (fixed)	80.96%	16.00	1.00×
DICE (w. adaptive halt)	$\leq$ 16	**81.05%**	**2.43**	6.58×

Multi-step temporal reasoning (Universal Transformer, bAbI). DICE reaches **93.69%** vs. 89.72% for endpoint supervision (+3.97 pp), outperforming all baselines (Tab. 5).

Table 5. **Universal Transformer on bAbi benchmark.** Comparison of training strategies.

Method	**Test accuracy**	**vs endpoint**
Endpoint	0.8972	–
GRPO	0.8988	+0.16 pp
LSRL	0.8982	+0.10 pp
RLTT	0.8987	+0.15 pp
DICE	0.9369	+3.97 pp

Why it works. Sec. E measures gradient direction and variance along the rollout: endpoint supervision yields anti-aligned early-step gradients with mini-batch variance up to 10^4; DICE realigns direction and compresses variance by 6–8 orders of magnitude. Sec. F confirms both (M1) and (M2) are individually necessary. Sec. H additionally shows DICE eliminates a catastrophic mid-training collapse in DEQs, the INC with the longest rollout chain.

5. Discussion

Connection to world models. The INC abstraction – a learned transition f_θ iterated to produce forecasts – underlies learned world models. Endpoint supervision is the rollout analogue of supervising only the final step: gradients traverse the full horizon while intermediate predictions go unconstrained. DICE's mechanisms import directly: supervising every rollout step against ground-truth observations through a shared observation head. Prop. 1 then bounds how confidently intermediate rollouts can be used for planning. We see this as the immediate next step for the framework.

ASH as compute-budget allocation. The shared-readout property turns INCs into anytime systems; ASH is the simplest decision-theoretic policy on top – "stop when the forecast stops moving." Calibration- or risk-aware halting policies are immediate extensions.

Limitations. Optimal α and schedule depend on which failure mode dominates (Sec. J); adaptive schedules are a natural next step. DICE assumes every step decodes to a meaningful forecast, which may not hold for multi-phase computations. Extending the calibration bound to continuous outputs (regression, density estimation) remains open.

6. Related Work

Iterative forecasters. Universal Transformers (Dehghani et al., 2019), DEQs (Bai et al., 2019; 2020), Deep Thinking (Schwarzschild et al., 2021), looped Transformers (Giannou et al., 2023; Zhu et al., 2025; Geiping et al., 2025), and learned world models (Ha & Schmidhuber, 2018; Hafner et al., 2020) all instantiate INC. **Deep / process supervision.** Classical deep supervision (Lee et al., 2015; Szegedy et al., 2015) uses per-layer heads; INCs share parameters, so DICE reshapes *dynamics* rather than independent features (Sec. F). Verifier-based process supervision (Lightman et al., 2024; Uesato et al., 2022) acts at inference; RL step rewards (Ren, 2025; Williams & Tureci, 2026) act post-training. DICE shapes dynamics from the ground up. **Gradient flow.** Skip connections (He et al., 2016), clipping (Pascanu et al., 2013), and normalization (Ba et al., 2016) target gradient *magnitude*; our diagnosed pathology is *directional* (Sec. G).

7. Conclusion

We frame iterative neural computations as a unified class of forecasters and identify endpoint supervision as the shared cause of long-horizon instability and trajectory miscalibration. DICE addresses both with a single loss change, with a trajectory-calibration guarantee and a 6.6× inference-time speedup via Adaptive Stability Halting. We see DICE as a step toward iterative neural systems whose every step is a usable, calibrated forecast.

Acknowledgements

Anonymous.

References

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Bai, S., Kolter, J. Z., and Koltun, V. Deep equilibrium models. *Advances in neural information processing systems*, 32, 2019.

Bai, S., Koltun, V., and Kolter, J. Z. Multiscale deep equilibrium models. *Advances in neural information processing systems*, 33:5238–5250, 2020.

Dehghani, M., Gouws, S., Vinyals, O., Uszkoreit, J., and Kaiser, Ł. Universal transformers. In *International Conference on Learning Representations (ICLR)*, 2019.

Dieng, A. B., Ranganath, R., Altosaar, J., and Blei, D. Noisin: Unbiased regularization for recurrent neural networks. In *International Conference on Machine Learning*, pp. 1252–1261. PMLR, 2018.

Geiping, J., McLeish, S., Jain, N., Kirchenbauer, J., Singh, S., Bartoldson, B. R., Kailkhura, B., Bhatele, A., and Goldstein, T. Scaling up test-time compute with latent reasoning: A recurrent depth approach. *arXiv preprint arXiv:2502.05171*, 2025.

Giannou, A., Rajput, S., Sohn, J.-y., Lee, K., Lee, J. D., and Papailiopoulos, D. Looped transformers as programmable computers. In *International Conference on Machine Learning (ICML)*, 2023.

Ha, D. and Schmidhuber, J. World models. *arXiv preprint arXiv:1803.10122*, 2(3):440, 2018.

Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., and Davidson, J. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations (ICLR)*, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

Lamb, A., Goyal, A., Zhang, Y., Zhang, S., Courville, A., and Bengio, Y. Professor forcing: A new algorithm for training recurrent networks. In *Advances in Neural Information Processing Systems (NIPS)*, volume 29, pp. 4601–4609, 2016.

Lamb, A., Binas, J., Goyal, A., Subramanian, S., Mitliagkas, I., Bengio, Y., and Mozer, M. State-reification networks: Improving generalization by modeling the distribution of hidden representations. In *International Conference on Machine Learning*, pp. 3622–3631. PMLR, 2019.

Lee, C.-Y., Xie, S., Gallagher, P., Zhang, Z., and Tu, Z. Deeply-supervised nets. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 562–570, 2015.

Lehnert, L., Sukhbaatar, S., Su, D., Zheng, Q., Mcvay, P., Rabbat, M., and Tian, Y. Beyond a*: Better planning with transformers via search dynamics bootstrapping, 2024.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. In *International Conference on Learning Representations (ICLR)*, 2024.

Pascanu, R., Mikolov, T., and Bengio, Y. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning (ICML)*, pp. 1310–1318, 2013.

Ren, H. Lsrl: Process-supervised grpo on latent recurrent states improves mathematical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 12534–12545, 2025.

Schwarzschild, A., Borgnia, E., Gupta, A., Huang, F., Vishkin, U., Goldblum, M., and Goldstein, T. Can you learn an algorithm? generalizing from easy to hard problems with recurrent networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.

Uesato, J., Kushman, N., Kumar, R., Song, F., Siegel, N., Wang, L., Creswell, A., Irving, G., and Higgins, I. Solving math word problems with process- and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Williams, J. and Tureci, E. Prioritize the process, not just the outcome: Rewarding latent thought trajectories improves reasoning in looped language models. *arXiv preprint arXiv:2602.10520*, 2026.

Zhu, R.-J., Wang, Z., Hua, K., Zhang, T., Li, Z., Que, H., Wei, B., Wen, Z., Yin, F., Xing, H., et al. Scaling latent reasoning via looped language models. *arXiv preprint arXiv:2510.25741*, 2025.

Table 6. **Diverse architectures, one abstraction.** Every system is a loop over a shared f_θ.

Architecture	Transition f_θ	Readout g_θ	K	Domain
Universal Transformer	Transformer block	Output projection	Adaptive (ACT)	Language
Deep Thinking	Recurrent block	Linear head	40 (test)	Algorithmic
Recursive Transformer	Transformer cycle	LM head	12	Structured

A. Impact Statement

This work introduces a training-time framework for iterative neural computation. Its societal consequences depend on the downstream systems it enables. On the positive side, dense supervision can reduce inference-time compute via adaptive halting, lowering energy use; it can support more reliable iterative reasoners in scientific discovery, formal verification, planning, and education, where step-by-step refinement is more trustworthy than single-pass prediction. The anytime property exposes intermediate predictions to inspection, a useful primitive for interpretability. On the negative side, any technique that makes reasoning models more capable potentially amplifies broader risks of advanced AI systems, including misuse for disinformation and automation of harmful tasks. We encourage pairing such methods with investment in safety and responsible deployment.

B. INC Instances: Encoder, Block, Readout, and K

Universal Transformers reuse a Transformer block across depth, iterating it K times. $\mathbf{z}^k$ represents all token embeddings; h_θ initializes them with positional encoding; g_θ maps selected tokens to outputs.

Deep Thinking networks are convolutional/convolutional-recurrent models where h_θ embeds the input into a spatial feature map, f_θ propagates information across it, and g_θ decodes per-position predictions. Designed for algorithmic extrapolation – increasing K at test time should enable larger instances.

Recursive Transformers (TRM) apply a Transformer block recursively to a hidden state representing a partial solution, conditioned on the input each step. Each application of f_θ refines a candidate output rather than tying fixed layers.

Deep Equilibrium Models replace explicit unrolling with a fixed point $\mathbf{z}^\star = f_\theta(\mathbf{z}^\star)$ found via a root-finding solver (e.g., Broyden), with gradients through the implicit function theorem.

Learned world models instantiate f_θ as a state-transition function and g_θ as an observation or reward head; rolling out K steps produces a trajectory of forecasts over a horizon.

C. Theoretical Evidence

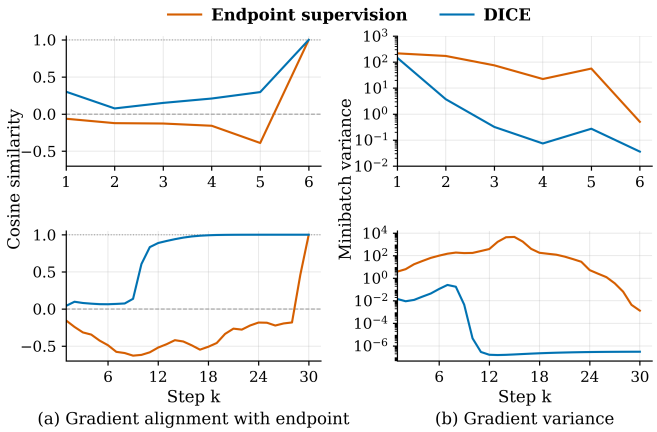
C.1. Relative Gradient Noise under Endpoint and Dense Supervision

Consider $z_{k+1} = f_\theta(z_k)$, $k = 0, \dots, K-1$. The contribution of a length-L backprop path contains a multiplicative factor $P_L = \prod_{j=1}^{L} A_j$, where A_j models the random Jacobian factor at step j. We use *relative variance* $\mathrm{RV}(P_L) := \mathrm{Var}(P_L)/\mathbb{E}[P_L]^2$.

Proposition 2 (Exponential growth of relative gradient noise)**.** *Let $A_1, \dots, A_L$ be independent with $\mathbb{E}[A_j] = \mu \neq 0$ and $\mathrm{Var}(A_j) = \sigma^2$. Then $\mathrm{RV}(P_L) = (1 + \sigma^2/\mu^2)^L - 1$.*

Proof. By independence, $\mathbb{E}[P_L] = \mu^L$ and $\mathbb{E}[P_L^2] = \prod_j \mathbb{E}[A_j^2] = (\mu^2 + \sigma^2)^L$. So $\mathrm{Var}(P_L) = (\mu^2 + \sigma^2)^L - \mu^{2L}$ and $\mathrm{RV}(P_L) = ((\mu^2 + \sigma^2)/\mu^2)^L - 1 = (1 + \sigma^2/\mu^2)^L - 1$. □

Implications. For endpoint supervision, the gradient at step t contains $K - t$ Jacobian factors, giving $\mathrm{RV}_{\mathrm{end}}(t) = (1 + \sigma^2/\mu^2)^{K-t} - 1$: noise grows exponentially with distance from the endpoint. Dense supervision contributes a path of length $m - t < K - t$ via the auxiliary loss at step m, giving $\mathrm{RV}_{\mathrm{dense}}(t, m) = (1 + \sigma^2/\mu^2)^{m-t} - 1 < \mathrm{RV}_{\mathrm{end}}(t)$ whenever $\sigma^2 > 0$. Dense supervision replaces a single long-range, noise-amplified signal with a collection of shorter-path signals having exponentially smaller relative noise.



(a) Gradient alignment with endpoint (b) Gradient variance

Figure 1. **Gradient pathology under endpoint supervision.** (a) Cosine similarity between step-k and endpoint gradients: endpoint supervision yields anti-aligned early iterates ($\cos < 0$), snapping to 1.0 only at $k = K$, while DICE aligns within a few steps. (b) Per-step gradient variance across mini-batches (log scale). Top is Universal Transformers, bottom is Deep Thinking model.

C.2. Proof of Proposition 1 (Prediction-Space Stability)

For a one-hot target e_y, $\|p - e_y\|_1 = 2(1 - p[y]) \le 2(-\log p[y]) = 2\,\ell_{\mathrm{CE}}(p, y)$, using $-\log u \ge 1 - u$ for $u \in (0, 1]$. Because all terms in $\mathcal{L}_{\mathrm{dense}}$ are nonnegative, $\lambda_k\,\mathbb{E}_x[\ell_{\mathrm{CE}}(p_k, \mathbf{y})] \le \mathcal{L}_{\mathrm{dense}} \le \delta$, so $\mathbb{E}_x[\ell_{\mathrm{CE}}(p_k, \mathbf{y})] \le \delta/\lambda_k$ and $\mathbb{E}_x[\|p_k - e_{\mathbf{y}}\|_1] \le 2\delta/\lambda_k$. The consecutive bound follows from the triangle inequality: $\mathbb{E}_x[\|p_{k+1} - p_k\|_1] \le \mathbb{E}_x[\|p_{k+1} - e_{\mathbf{y}}\|_1] + \mathbb{E}_x[\|p_k - e_{\mathbf{y}}\|_1] \le 2\delta(1/\lambda_{k+1} + 1/\lambda_k)$. Endpoint supervision controls only p_K and is consistent with arbitrary p_k for $k < K$, hence with arbitrary inter-step drift.

D. Baselines

Endpoint: standard endpoint supervision; the primary baseline. **GRPO** (Shao et al., 2024): group-normalized policy gradient on the terminal iterate; outcome-reward only at the endpoint. **LSRL** (Ren, 2025): dense intermediate supervision via RL, decoding each iterate through the shared readout, with an intermediate verifier assigning per-step rewards (optimized with GRPO). **RLTT** (Williams & Tureci, 2026): distributes the outcome-based reward across iterations via a weighted sum of per-step log-probabilities under a shared readout; trajectory-level credit assignment without verifiers. All baselines use the same training data per task.

E. Gradient-Pathology Diagnostic

Setup. A Universal Transformer is trained on bAbI 10k with $K = 6$ and a Deep Thinking model on prefix sums with $K = 30$. At each step we log two diagnostics for every iterate k: (i) **Gradient alignment**: $\cos(g_k, g_K)$, where $g_k = \nabla_\theta \ell(g_\theta(\mathbf{z}^k), \mathbf{y})$ is the gradient produced by reading out at k, and g_K is the endpoint gradient – the direction of the learning signal at k relative to what endpoint supervision uses. (ii) **Mini-batch gradient variance**: trace of the per-example covariance, $\mathrm{Var}_i[g_k^{(i)}]$.

Results. Under endpoint supervision, $\cos(g_k, g_K)$ is *negative* for the first two-thirds of the chain ($[-0.6, -0.2]$), collapsing to 1.0 at $k = K$ by construction: the optimizer is updating f_θ based on a signal pointing nearly opposite to what the endpoint loss prefers, for most iterations. Under DICE the auxiliary gradient is well-aligned with the endpoint within $\sim$10 of 30 steps in Deep Thinking and never becomes anti-aligned. Mini-batch variance at intermediate steps reaches 10^2–10^4 under endpoint supervision; DICE compresses this by 6–8 orders of magnitude past step 10 (down to $\sim 10^{-7}$). Together these confirm that endpoint supervision delivers a gradient that is wrong in direction and noisy in magnitude at most early iterations.

F. Four-Way Mechanism Ablation

We decompose DICE into **gradient shortening**, **representation constraint**, and **additional regularization** on the Universal Transformer: (1) **Endpoint** baseline. (2) **Final-state repeated-loss**: same number of auxiliary terms and total weight, all applied to $\mathbf{z}^K$ – controls for extra supervision/regularization without shortening the gradient path. (3) **Intermediate losses, separate heads**: supervision at intermediate steps with independent g_k per step – preserves shortened gradients but removes the shared-decodable-representation constraint. (4) **DICE**: intermediate losses with a shared head.

The comparison (3)>(2) isolates gradient shortening; (4)>(3) isolates the shared-representation constraint; (4)>(1) is the combined effect. Together with Sec. E and Prop. 2, this completes a single causal account.

Table 7. **Four-way ablation on the Universal Transformer (bAbI 10k).** The improvement decomposes into a gradient-shortening effect (#3 vs. #2) and a shared-representation effect (#4 vs. #3).

Mode	**Mean acc.**	**Δ vs. previous**
(1) Baseline (endpoint)	0.8972	—
(2) Final-repeated (extra regularization only)	0.9155	+1.83 pp
(3) Separate heads (short gradients only)	0.9311	+1.56 pp
(4) Shared head (full DICE)	**0.9369**	+0.58 pp

G. DICE vs. Post-Hoc Stabilization

Since the pathology is one of *direction*, post-hoc techniques targeting gradient norm should provide only marginal relief. On bAbI 1k with the Universal Transformer (Tab. 8), skip connections, state normalization, and gradient clipping combined raise accuracy from 51.25% to 54.75%. DICE reaches 57.64%, $+2.9$ pp over the best combined baseline. Injecting short-range gradient signals at every iteration is fundamentally more effective than post-hoc stabilization of a single long, noisy gradient path.

Table 8. **DICE vs. post-hoc stabilization on bAbI 1k.** Mean accuracy across tasks. Standard stabilization techniques target gradient magnitude; DICE targets the underlying long-range gradient signal.

Method	Transition skip	State norm	Grad clip norm	Mean acc.
Baseline (None)	No	No	–	0.5125
Norm only	No	Yes	–	0.5153
Skip only	Yes	No	–	0.5365
Clip only	No	No	1.0	0.5437
Skip + Norm + Clip	Yes	Yes	1.0	0.5475
DICE (ours)	No	No	–	**0.5764**

H. Catastrophic Training Collapse in Deep Equilibrium Models

DEQs replace explicit unrolling with $\mathbf{z}^\infty = f_\theta(\mathbf{z}^\infty)$, found via Broyden or Anderson acceleration. The Broyden solver has the longest iteration chain among our systems ($K \approx 27$), where the directional pathology of Sec. E is most severe.

Setup. MDEQ-Large (Bai et al., 2020) on CIFAR-10 (Krizhevsky & Hinton, 2009), 60 epochs, ~ 27 Broyden iterations. We supervise $K = 5$ evenly-spaced solver iterates with the classification head, exponential schedule.

Problem and solution. At epoch 13 the baseline collapses: top-1 drops from $\sim 85\%$ to 39.65% before recovering to a best of 91.08% (Tab. 10). Perturbations in the solver trajectory at early steps are amplified through 27 Jacobian products, pushing parameters into an unstable region. DICE passes through epoch 13 stably at 85.17%, reaches $\mathbf{91.65}\%$ best ($+0.57$ pp), and hits 90% at epoch 32 vs. 40 for the baseline (20% faster).

I. Constraining Intermediate States via a Shared Readout

Prior work regularizes intermediate representations indirectly. **Noisin** (Dieng et al., 2018) injects noise into hidden states and maximizes expected likelihood, encouraging smoothness but not semantic consistency. **Professor Forcing** (Lamb et al.,

Table 9. **DEQ on CIFAR-10.** Best top-1 accuracy (%). DICE with exponential schedule and K=5 improves over endpoint baseline.

Configuration	Acc.	Δ
Baseline (endpoint)	91.08	—
DICE (exp., K=5)	**91.65**	+0.57

Table 10. **Training stability at DEQ epoch 13.** The baseline collapses to near-random; DICE passes stably.

Configuration	Acc. at ep. 13 (%)
Baseline (endpoint)	39.65
DICE	**85.17**

2016) aligns training and generation distributions via an adversarial discriminator. **State-Reification** (Lamb et al., 2019) projects states onto a learned manifold via a denoising autoencoder. The latter two require auxiliary networks.

On bAbI 10k with the Universal Transformer (Tab. 11), Noisin provides a small gain over baseline ($+1.5$ pp); DICE achieves $+4.0$ pp without stochasticity or auxiliary networks. The distinction is that prior methods shape the *distribution* of hidden states; DICE constrains their *function* – every state must be decodable by the same head – shifting from regularization to representation design.

Table 11. **Constraining intermediate states on bAbI 10k.** Mean accuracy across tasks.

Method	Mean acc.	vs. baseline
Baseline	0.8972	—
Noise injection (Noisin)	0.9109	+1.5 pp
DICE (ours)	**0.9369**	+4.0 pp

J. Which Weight Schedule, and Why

The optimal schedule depends on which failure mode dominates (Tab. 12). **Exponential** suits systems where (F1) dominates (long K, early iterations unreliable) – consistent with DEQs (Sec. H), where exponential outperforms linear and uniform. **Linear** suits (F2) (extrapolation requires progressive improvement) – consistent with prefix sums (Tab. 1), where linear yields perfect extrapolation to 128 bits. **Uniform** suits short-K systems where each cycle contributes comparably.

Training longer (maze). We ran the maze baseline for 7,000 epochs (75% longer than 4,000). Best accuracy was 79.5%, still below DICE's 81.05% obtained in fewer epochs – longer training does not close the gap.

Gradient clipping on DEQ. Clipping at $\{1.0, 5.0, 10.0\}$ still admits the epoch-13 collapse (accuracy dips to $\sim 55\%$ instead of 39.65%). DICE eliminates the collapse entirely. Clipping treats the symptom (large norms); DICE treats the cause (noisy long-range gradient signal).

K. Implementation Details

Universal Transformer. Hidden 128, 4 heads, max 10 ACT steps. Adam, lr 10^{-3}, batch 64, 100 epochs, patience 15. Shared output projection at each ACT step. Sweep: 3 schedules $\times$ 4 α values $\times$ 3 seeds $=$ 36 runs/task.

DEQ. MDEQ-Large, 3 resolution streams. SGD with momentum 0.9, initial lr 0.1, cosine annealing, weight decay 10^{-4}, batch 128, 60 epochs. Broyden solver: 27 steps train and test. Intermediate readout on $K = 5$ evenly-spaced iterates with the shared classification head, exponential schedule.

Deep Thinking. Recurrent architecture, hidden 256. Adam, lr 10^{-3}, decay $\times 0.1$ at epoch 60, batch 64, 80 epochs. Linear schedule.

Recursive Transformer. $L = 2$, $H = 3$, $L_c = 4$. Adam, lr 10^{-4}, weight decay 1.0, cosine schedule, global batch 384, EMA, 4,000 epochs, eval every 1,000. Shared LM head at each H-cycle; gradient checkpointing over H-cycles.

Constraining intermediate states (Sec. I). Noisin std 0.25. 100 epochs, patience 20, 5 bAbI tasks, 3 seeds/task, 10k dataset. Best-checkpoint test accuracy.

bAbI post-hoc (Sec. G). Exponential schedule with $\alpha = 0.5$.

Adaptive Stability Halting (TRM). $\epsilon = 5 \times 10^{-3}$, patience $m = 1$.

Table 12. **Best DICE configuration across systems.** The optimal schedule correlates with the primary problem being solved.

System	Best schedule	α	K	Rationale
DEQ (MDEQ)	Exponential	2.0	$\sim$27	Long chain; emphasis on later (better) iterates
Universal Transformer	Exponential	0.5	Adaptive (ACT)	Early iterates poor; weight later halts
Deep Thinking	Linear	1.0	40	Extrapolation needs progressive improvement
Recursive TRM	Uniform	1.0	12	Few cycles; each must contribute equally

Compute. All experiments were run on NVIDIA GeForce RTX 2080 Ti GPUs and NVIDIA A100 GPUs.