STATE-AWARE PROSPECTIVE RECOVERY AFTER NEURAL TRAINING DAMAGE

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural training can enter damaged states with several admissible recovery interventions, yet one action must be chosen before its future consequences are known. The central question is whether current task behavior contains enough information for that choice. In controlled digits AdamW states, 79 of 80 confirmatory matched pairs reverse the preferred retain/reset action after exact parameter restoration, showing that matched current behavior can conceal different recovery-action values. Self-Dynamics Network (SDN) is the branch-supervised recovery-decision framework used to study this problem. On held-out AdamW cohorts, state-aware scorers outperform a loss-only reference in binary retain/reset control, while state-conditioned policies have lower final-cohort cost than the lowest-cost fixed action in a four-action setting. Classical and neural scorers perform nearly identically, so these results do not support neural function-class superiority. A symmetry-compatible control identifies optimizer non-equivariance as one controlled route to divergent futures, while a separate seven-action Transformer teacher extends the decision principle under explicit future rollouts. These results support recovery-relevant training state as a source of decision value in the tested settings. Natural prevalence, large-scale competing actuation, and operational cost benefits remain open.

1 INTRODUCTION

After neural training damage, recovery may continue from the damaged state, restore a checkpoint, reset optimizer memory, or apply another admissible intervention. The choice must be made before the future consequences of those actions are observed. The central problem is action selection: the information available at decision time must distinguish states that require different recovery actions.

Partial-observation decision theory already establishes that latent states sharing an observation can require different actions, so this generic aliasing principle is not treated as new here (Kaelbling et al., 1998). Large-scale training reports document rollback and checkpoint restart as responses to instability, while recent stabilization and recovery methods use clipping, momentum reset, cautious updates, or supervisory recovery (Chowdhery et al., 2023; Zhang et al., 2022; Huang et al., 2025; Kumar et al., 2025; Or, 2026; Liang et al., 2026). Recent work shows that optimizer state can transport effects into later training or reorder prospective short-horizon choices (Xu, 2026; Guo, 2026), while optimizer-state and symmetry studies show that matched current behavior or function need not imply matched future optimization (Cattaneo and Shigida, 2025; Sevetlidis and Pavlidis, 2026; Zhao et al., 2022; Lebeurrier et al., 2026; Singh, 2026). Training-dynamics methods estimate future quantities (Wang et al., 2026; Shou et al., 2025). The unresolved recovery question is whether post-damage states matched under a restricted recovery observation can rank competing recovery interventions differently, and whether decision-time state improves the resulting held-out recovery choice.

A restricted observation contains the task-behavior information exposed to the baseline policy, whereas a state-aware observation also includes decision-time optimizer or gradient information. If two states match under the restricted observation but rank the available actions differently, no single restricted-observation decision can be optimal for both. Future prediction is useful only insofar as it helps distinguish those action values.

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.

Self-Dynamics Network (SDN) is the recovery-decision interface used here. At a damaged state, it scores the admissible recovery actions from the available information before their outcomes are known and commits one action. Candidate branches are executed only during development to provide action-specific supervision. Neural and classical scorers instantiate the same interface, allowing held-out comparisons to distinguish evidence for training-state information from evidence for scorer-class advantage.

Recovery-action aliasing matters only if matched states can develop different futures and admissible interventions can alter those futures. The experiments move from matched-state reversals and optimizer-symmetry controls to held-out binary and multi-action recovery decisions. A rollout-enabled Transformer study tests the same decision principle under a larger information budget; separate diagnostic measurements probe transfer across settings and scale.

2 PRELIMINARIES

A damaged run has a complete training state, while a recovery policy may see only part of it. Two runs can look identical on current task behavior while retaining different optimizer memories. If those memories change which recovery action is preferable, a policy that sees only current behavior must apply the same action to states with different optima.

Let $s_t = (\theta_t, z_t)$ denote the complete training state, with task parameters θ_t and optimizer state z_t. At decision time τ, B maps the state history to the restricted observation b_τ, while O maps it to the state-aware observation o_τ. The admissible action set is $\mathcal{A}_\tau$, and H denotes the evaluation horizon. The quantity $Q(s_\tau, a)$ is the expected future recovery cost of choosing action a from state s_τ. A deployable policy does not observe Q or future branch outcomes directly.

$$\begin{aligned} s_t &= (\theta_t, z_t), \\ b_\tau &= B(s_{0:\tau}), \\ o_\tau &= O(s_{0:\tau}), \\ Q(s_\tau, a) &= \mathbb{E}[C(s_{\tau+1:\tau+H}) \mid s_\tau, a]\,. \end{aligned}$$

For two states with the same restricted observation, a restricted policy must use one common action even when a state-aware policy would choose separately. The first term below is the best average cost when the pair is forced to share one action. The second is the average cost when each state may choose its own best action.

$$G_B(s^A, s^B) = \min_{a \in \mathcal{A}} \frac{Q(s^A, a) + Q(s^B, a)}{2} - \frac{1}{2}\left[\min_a Q(s^A, a) + \min_a Q(s^B, a)\right].$$

For the two-action reversal case, let s^A prefer a_1 and s^B prefer a_2. Writing the two positive action margins as

$$\Delta_A = Q(s^A, a_2) - Q(s^A, a_1), \qquad \Delta_B = Q(s^B, a_1) - Q(s^B, a_2),$$

the shared-action excess cost becomes

$$G_B(s^A, s^B) = \frac{1}{2}\min\{\Delta_A, \Delta_B\}.$$

Proposition 1 (Recovery-specific aliasing criterion)**.** *If $B(s^A) = B(s^B) = b$, then $G_B \geq 0$. If both states have unique optimal actions and those actions differ, then $G_B > 0$.*

If the preferred actions differ with nonzero margins, a policy constrained to the shared restricted observation cannot be optimal in both states. A positive G_B is therefore the cost of forcing one shared decision on states that prefer different actions. Here, *training-state aliasing* refers to this recovery-specific case: states share the restricted observation but not the optimal recovery action over the declared action set. The criterion specializes standard observation aliasing to recovery; it identifies when the restricted observation is insufficient without specifying how different futures arise.

One way function-preserving states can acquire different futures is for the optimizer update to fail to respect the same transformation. Let T_g denote such a transformation, F the represented function, and $\Phi(\cdot; \xi)$ one optimizer update under shared input or randomness ξ. Equation (1) gives the exclusion

case in which the optimizer does respect the transformation. For a paired trajectory with $\widetilde{s}_0 = T_g s_0$ and shared update inputs ξ_t, define $e_t = \|\widetilde{s}_t - T_g s_t\|$. Let L_t be a local Lipschitz factor satisfying $\|\Phi(\widetilde{s}_t; \xi_t) - \Phi(T_g s_t; \xi_t)\| \le L_t e_t$. With $e_0 = 0$, the equivariance defect E_g and its one-step norm δ_t give the trajectory bound

$$
\begin{aligned}
F(T_g s) &= F(s), \qquad \Phi(T_g s; \xi) = T_g \Phi(s; \xi), \\
E_g(s; \xi) &= \Phi(T_g s; \xi) - T_g \Phi(s; \xi), \\
\delta_t &= \|\Phi(T_g s_t; \xi_t) - T_g \Phi(s_t; \xi_t)\|, \\
e_{t+1} &\le L_t e_t + \delta_t, \qquad e_h \le \sum_{k=0}^{h-1} \delta_k \prod_{j=k+1}^{h-1} L_j .
\end{aligned}
\tag{1}
$$

Proposition 2 (Optimizer-equivariance exclusion criterion)**.** *Under Equation* (1)*, paired trajectories starting from s and $T_g s$ represent identical future functions at every matched step; a controller restricted to such transformations cannot change a recovery outcome defined only in function space under shared evaluation inputs.*

If the optimizer respects the transformation, function preservation extends to the matched function-space future. Without that equivariance, paired trajectories may separate. A recovery effect still requires the separation to change the relative values of the available actions.

3 Methodology

At deployment, SDN receives the information available at a damaged state, scores the admissible recovery actions, and commits one action before any candidate future is observed. Figure 1 summarizes this process. Future candidate outcomes are used only during development to train and select the scorer.

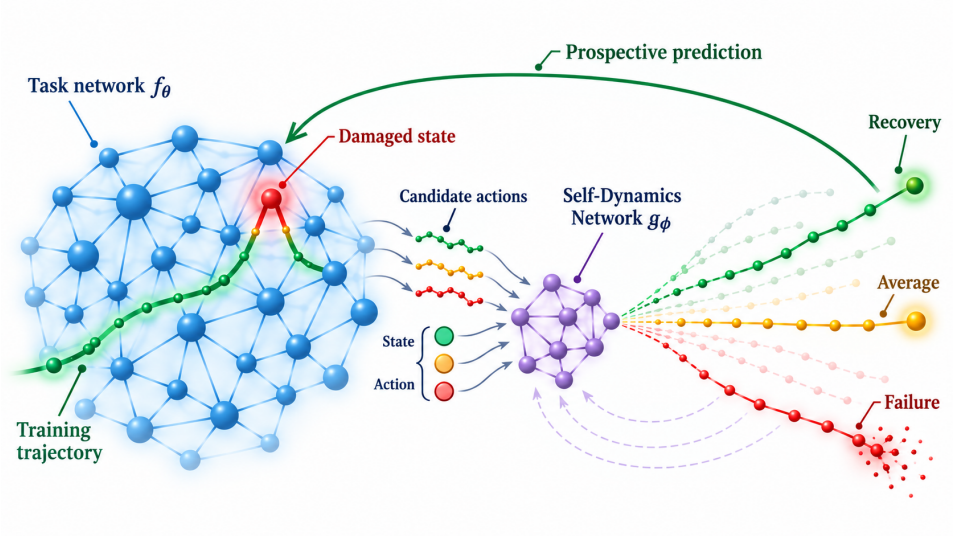

Figure 1: SDN recovery-decision interface and development/deployment separation.

Equation (2) gives the neural scorer used within the SDN decision framework; this neural realization is termed the Self-Dynamics Neural Network (SDNN). The action with the lowest predicted recovery cost is selected. Classical scorers use the same observation-action interface.

$$
\widehat{Q}_{\tau,a} = g_\phi(o_\tau, a), \qquad \widehat{a}_\tau = \arg\min_{a \in \mathcal{A}_\tau} \widehat{Q}_{\tau,a}. \tag{2}
$$

During development, each decision-time state is copied across all admissible actions and the branches are advanced under the same predefined future-input protocol. Their outcomes provide action-specific targets. The model, thresholds, and controller are fixed before the held-out final cohort is opened; final action choice then uses only the observation and admissible action set.

The neural scorer in Equation (2) is one realization of the branch-supervised framework, while classical scorers provide same-information controls. The held-out comparisons separate evidence for decision-time state from evidence for a neural function-class advantage. In the binary AdamW control, the restricted observation is a ten-step log-loss history plus current log loss; the state-aware observation adds current gradient norm, Adam first- and second-moment norms, first-moment/current-gradient cosine, preconditioned-moment norm, and the moment/gradient norm ratio. The four-action control uses layerwise versions of the gradient and Adam-state summaries.

The Transformer teacher uses a different operating regime because temporary future branches are allowed before commitment, unlike the state-only deployment policy in Equation (2). The candidate action set is predefined independently of the scorer. The evaluated actions include continuation, checkpoint restoration, optimizer-memory retention or reset, layerwise reset, scalar reciprocal rescaling, ReLU rescaling, and function-preserving attention-gauge transformations. Automatic action discovery is not evaluated.

4 EXPERIMENTS

In the digits AdamW construction, each pair is cloned from a common trained state. One clone receives one actual AdamW optimizer-state write using the true labels and the other a write using 50% corrupted labels; task parameters are then restored exactly. At the decision point, task parameters, represented function, current true-task loss, and pre-decision loss history match while optimizer memory differs. Recovery compares KEEP with RESET, which clears the Adam first- and second-moment buffers while preserving the optimizer step counter. Matched damaged states can prefer different recovery actions even when the restricted observation is the same. The sine momentum-SGD and digits AdamW constructions both compare retaining versus resetting optimizer memory after the predefined matched quantities are restored. Reversals occur in both settings, including 79/80 AdamW pairs (Table 1). These controlled constructions establish that the aliasing condition can occur; they do not estimate its natural prevalence.

Table 1: Matched-state interventions.

System	Matched quantities	Actions	Different preferred action	State-by-action interaction
Sine, momentum SGD	θ, f, L, H_L	retain/reset memory	33/80	50/80 positive; median 33.12 [2.61, 235.05]
Digits, AdamW	θ, f, L, H_L	retain/reset memory	79/80	80/80 positive; median 61.18 [60.31, 63.14]

θ: task parameters; f: represented function; L: current loss; H_L: pre-decision loss history. Interaction: positive pairs; median [bootstrap 95% CI]. Parameters are restored before action selection.

Action reversal under a matched observation matters only if the paired states can develop different futures. Under the same ReLU rescalings, Adam shows a nonzero one-step defect and later path divergence, whereas the symmetry-compatible Path-SGD control remains at the numerical-precision floor (Table 2). Optimizer non-equivariance is thus one controlled route to different futures, although separation alone says nothing about whether an intervention is beneficial.

Table 2: Optimizer-symmetry comparison.

Check	Adam scaling	Symmetry-compatible control	Outcome
Function preservation	132/132; 1.14×10^{-15}	8.38×10^{-16}	Off-grid passed
One-step defect D_1	Median 0.02325; max 0.09431	Max 9.34×10^{-16}	Separated
Path divergence R_H	Max 0.18080	Max 1.59×10^{-15}	Collapsed
D_1 to R_H	$\rho = 0.863$ [0.780, 0.932]	Numerical floor	Associated
$\lvert\Delta T_{\mathrm{rec}}\rvert$	Max 308 steps	0 steps	Collapsed

D_1: one-step equivariance defect; R_H: path divergence; ΔT_{rec}: recovery-time difference. Brackets for ρ: bootstrap 95% CI. Path-SGD is a symmetry control, not an absolute optimizer-performance comparison.

Different futures matter for recovery only if an admissible action can change what happens next. In the scalar-factorization probe, identity fails on six held-out states and nonidentity choices recover all six, while the learned value selector misses its prespecified gate. The result establishes local action authority, but not successful learned selection.

Table 3: Scalar internal actuation.

Decision rule	Recovery	Identity failures recovered	Utility	Δ vs fixed
Identity	2/8	0/6	0.00000	
Best fixed action	8/8	6/6	0.71534	0.00000
Short horizon	8/8	6/6	0.71101	-0.00433
Value model	8/8	6/6	0.71967	+0.00433
Oracle (future-aware maximum)	8/8	6/6	0.72401	+0.00867

Recovery: 8 held-out states; identity fails on 6. Utility = rescue reward - harm penalty - rescaling penalty; Δ is vs. best fixed. Oracle is retrospective. The learned selector missed the prespecified $\Delta U \geq 0.05$ gate.

The binary retain/reset task provides the cleanest comparison between restricted and state-aware observations because both policies act on the same two-action problem. Its untouched final cohort preserves the matched-observation ambiguity: the restricted loss observation is identical within each clean/contaminated pair, while the preferred retain/reset action differs in 78 of 80 final pairs. The four-action task tests whether state-conditioned selection can outperform all fixed internal actions on the untouched final cohort. Table 4 reports both held-out comparisons.

Table 4: State-aware prospective control.

Policy	Mean cost	Δ cost vs. reference [95% CI]
Binary retain/reset		
loss-only	64.6614	reference
state logistic/HGB	50.0788	-14.5827 [-16.1913, -13.0271]
SDNN neural scorer	50.2302	-14.4313 [-16.0326, -12.8675]
future-aware minimum	50.0160	infeasible
Four-action control		
final-cohort best fixed reset	31.3847	reference
HGB	28.4242	-2.9604 [-3.8371, -2.1185]
SDNN action scorer	28.4340	-2.9507 [-3.8521, -2.1031]
future-aware minimum	28.1596	infeasible

HGB: histogram gradient boosting. Binary: 80 final pairs / 160 states; 60-step cumulative true-task loss normalized by decision-time loss; pair-bootstrap CIs. Four-action: 140 final events; 40-step cumulative true-task loss normalized by decision-time loss; bootstrap CIs. Both final cohorts were generated after model selection was frozen. Future-aware minimum: infeasible retrospective lower bound.

On the held-out binary cohort, state-aware selection lowers realized cost relative to the loss-only reference while the matched-observation ambiguity persists in the same fresh cohort. The state-aware classical and SDNN scorers agree with the future-aware action on 98.75% and 98.125% of final binary states, respectively. In the four-action cohort, both state-conditioned policies have lower mean cost than the lowest-cost fixed action on that final cohort. Classical and neural scorers are nearly tied in both settings, so neither result supports neural function-class superiority.

The Transformer extension evaluates the same decision principle with a larger exact action bank and explicit future branches before commitment. This operating regime is not deployment-equivalent to the state-only SDN policy. Seven candidate actions are advanced on the same fixed future minibatch sequence, and the teacher selects the branch with the lowest 500-step shadow score in Equation (3). Let $L_{t+r}^{(a)}$ denote the task loss on branch a and $B_{\text{score}} > 0$ the fixed normalization baseline.

$$J_h(a; s_t) = \sum_{r=1}^{h} \log\left(\frac{L_{t+r}^{(a)}}{B_{\text{score}}}\right), \qquad a_h^*(s_t) = \arg\min_a J_h(a; s_t). \tag{3}$$

The selected branch then continues for 1500 steps. Seven 500-step shadows plus this continuation require 5000 recovery-update equivalents versus 2000 for one identity trajectory. This is raw-update accounting before parallelism; wall-clock time, energy, and I/O were not measured.

The teacher and action bank are fixed before evaluation on 48 held-out damaged states. Let B be the median pre-event loss. Sustained recovery begins at T_{50} when evaluation loss falls below $2B$ and remains below that threshold for 50 consecutive evaluations before the 2000-step horizon; $V_{50} = \max\{0, 1 - T_{50}/2000\}$, with zero assigned if sustained recovery is not reached. Table 5 shows higher mean V_{50} for the teacher than for identity and the predeclared fixed gauge (fixed4). The binary recovery comparison is not significant versus identity at the two-sided 5% level and is significant versus the predeclared fixed gauge.

Table 5: Counterfactual Transformer teacher.

Policy	Recovery	Mean V_{50}	Paired ΔV_{50}
500-step teacher	23/48	0.208	
Identity	17/48	0.143	+0.0658 [0.0198, 0.1141]
Predeclared fixed gauge (fixed4)	15/48	0.127	+0.0810 [0.0386, 0.1261]
Hindsight best fixed gauge (fixed1)	22/48	0.189	
Any-action oracle	29/48		

ΔV_{50} is teacher minus comparator; brackets are seed-bootstrap 95% CIs. Identity and fixed4 are deployable comparators; fixed4 was predeclared from development, whereas fixed1 is hindsight descriptive and the oracle is infeasible. Exact two-sided McNemar tests ($n = 48$): $p = 0.0703$ vs. identity, $p = 0.0215$ vs. fixed4.

Accurate future prediction alone does not imply useful recovery choice. Equation (4) trains a future quantity without comparing recovery actions, where y_{t+h} is the future target, sg denotes stop-gradient, and λ weights the auxiliary loss.

$$\mathcal{L}_{\text{dyn}} = \ell(g_\phi(o_t), \text{sg}[y_{t+h}]), \qquad \mathcal{L}_{\text{joint}} = \mathcal{L}_{\text{task}} + \lambda \mathcal{L}_{\text{dyn}}. \tag{4}$$

Future quantities are predictable in these settings, yet larger auxiliary weights in digits are associated with higher 500-step minimum loss (Table 6). Prediction accuracy alone does not establish recovery-action value.

Table 6: Short-horizon self-prediction.

System	λ	Dynamics correlation	Task observation
Sine regression	0.1	0.962	Final loss 0.002555; task-only 0.002569
Digits	0.1	0.991	500-step minimum 0.000275; task-only 0.000118
Digits	0.3	0.988	500-step minimum 0.000544
Digits	1.0	0.982	500-step minimum 0.000979

More state variables are also insufficient by themselves. In the continuation-versus-checkpoint comparison, both learned policies recover all 80 states, yet the complete-feature state tree has higher mean cost than the loss-only linear policy (Table 7 and Figure 2). Because the estimator also changes, this comparison does not isolate a feature effect; it only shows that the complete-feature policy was not the lower-cost policy in this protocol.

Table 7: Continuation and checkpoint restoration.

Policy	Recovery	Mean cost	Restorations
State tree (complete features)	80/80	73.75	59
Loss linear (loss-only)	80/80	50.00	40
Fixed rollback	80/80	100.00	80
Random	76/80	173.75	59
Fixed wait	62/80	450.00	0
Oracle (future-aware)	80/80	22.50	18
State tree minus loss linear: +23.75; bootstrap 95% CI [15.00, 33.75]			

Policy names match Figure 2; the oracle uses future outcomes retrospectively and is not deployable.

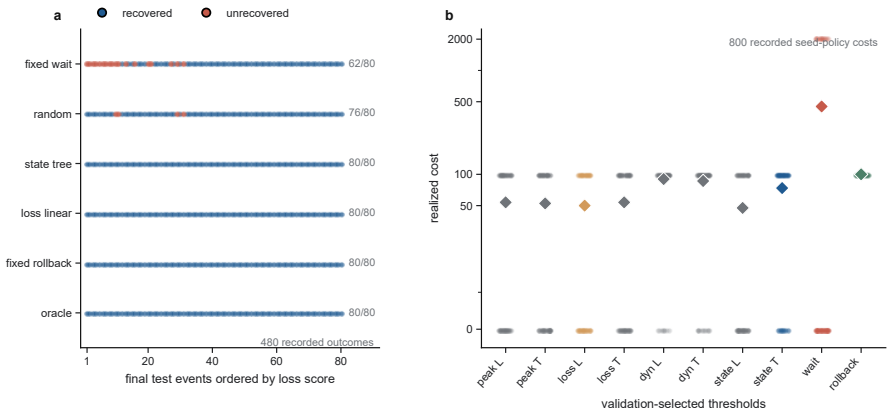

Figure 2: Continuation and checkpoint-restoration outcomes.

Recovery-time prediction in Figure 3 is accurate on 120 held-out events, but it does not compare competing recovery actions and cannot establish action value.

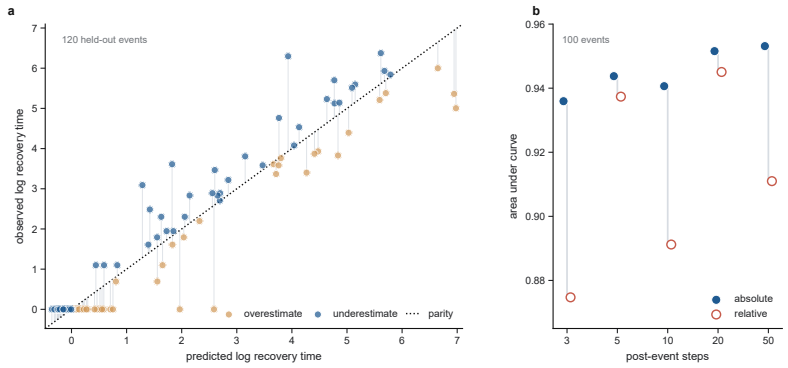

Figure 3: Recovery-time prediction.

Figures 4 and 5 ask how far the associated measurements transfer across settings and scale. Because competing recovery branches were not executed at the larger scales, these observations remain associational rather than tests of recovery selection.

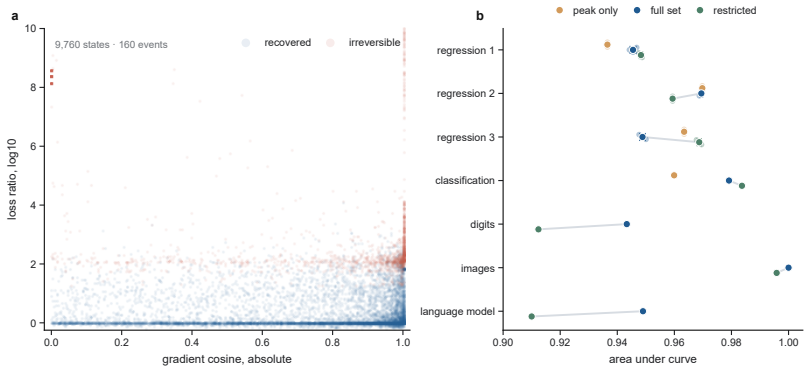

Figure 4: Outcome structure.

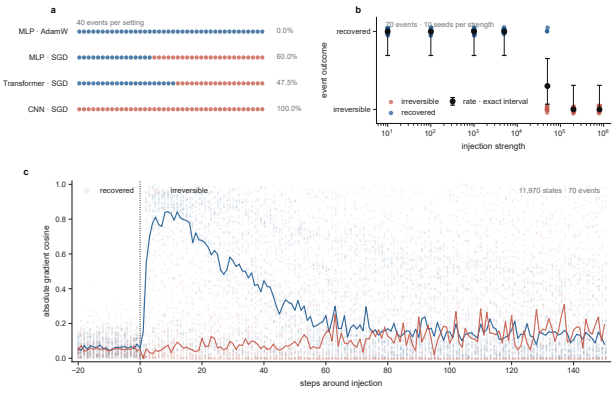

Figure 5: Transfer and stress measurements.

Table 8 collects additional diagnostic, compression, and transfer results. They provide context for the main experiments but do not evaluate prospective action selection.

Table 8: Scope diagnostics, compression, and unresolved tests.

Measurement	Observation	Interpretation
Diagnostics		
Recovery-time prediction	$r = 0.947$, $\rho = 0.888$ on 120 events	predictive diagnostic
Outcome prediction	AUC 0.925-1.000 across evaluated settings	measurement transfer
0.85B LM injection	AUC 0.949; recovery falls with stronger damage	cross-scale association
Released K2-65B log (Liu et al., 2025)	27/27 observed recoveries	observed association
Compression		
200-step student	16/20 teacher-action agreement; regret 4.68	development-only compression
Direct 200-step loss	8/20 agreement; regret 31.81	compression baseline
Scope tests		
MNIST matched-state test	0/30 sine-direction reversals; opposite interaction sign	construction not reproduced
Fresh sine controllers	all three neural-vs-fixed 95% CIs included zero	no controller gain detected
Neural vs classical	HGB/logistic match or slightly exceed neural in final cohorts	neural-class gain not detected
Large-scale competing recovery actions	not executed	branch control untested
Wall-clock / energy / I/O	not measured	operational cost unmeasured

5 Discussion

The central finding is about decision information, not damage measurement. The confirmatory matched-state constructions show that restricted task behavior can conceal different recovery-action rankings. A fresh binary final cohort preserves that ambiguity, with identical restricted observations but different preferred actions in 78 of 80 pairs, and state-aware selection lowers prospective recovery cost on the same cohort. The four-action result separately shows that state-conditioned selection can improve on a fixed internal action. Current task behavior can be an incomplete basis for recovery decisions in the tested settings. The controlled construction does not establish how often such ambiguity arises naturally.

Optimizer non-equivariance provides one controlled mechanism by which function-preserving states can acquire different futures. The symmetry-compatible control isolates that mechanism, and the actuation probe shows that an admissible intervention can matter. These results establish one route to recovery-relevant divergence, but do not imply that every recovery failure or action reversal has the same mechanism.

Prediction and state richness are not interchangeable with action-value information. In the continuation/checkpoint comparison, the complete-feature policy is not lower-cost; because representation and estimator change together, the result does not isolate the effect of additional state variables. The near-parity of classical and neural state-aware scorers points to decision-time information within the SDN interface, rather than an SDNN function-class advantage, as the source of the observed gain.

The Transformer teacher extends the decision principle to a larger exact action bank, but requires seven explicit future shadow rollouts before commitment. This places it under a larger information and compute budget than the state-only policy. Natural prevalence, large-scale competing actuation, and operational cost benefits remain open.

6 Conclusion

In the tested settings, restricted current task behavior is not sufficient to determine the best recovery action within the evaluated action sets. Matched states can prefer different interventions, and decision-relevant training state can reduce realized selection cost on held-out cohorts. Optimizer non-equivariance provides one controlled route to the divergent futures that make such distinctions possible. The evidence does not establish neural scorer superiority, natural prevalence, large-scale competing actuation, or operational cost benefits.

Reproducibility Statement

The underlying research record includes freeze and result snapshots for the reported AdamW claims, the claim-evidence ledger, the negative-result record, the ReLU optimizer-symmetry reports, and the Stage8-v3 Transformer source. For Table 4, binary model selection uses 160 fit pairs and 80 validation pairs before an 80-pair final cohort; the four-action study uses 280 development events before a fresh 140-event final cohort. The initially structured four-action neural model failed its validation gate; the

replacement flat model was selected by development-only grouped cross-validation and frozen before final generation. Confirmatory final cohorts are excluded from subsequent tuning. The accompanying source package contains the manuscript source, style files, and figures. Action branches within one damaged seed are treated as potential outcomes; the damaged seed is the inferential unit.

AI DISCLOSURE

Generative AI tools were used for literature search and synthesis; refinement of hypotheses and conceptual framing; feedback on methodology and experimental design; assistance with mathematical formulation, code, and technical tasks; interpretation checks; reviewer-style critique; manuscript organization; and language editing. All AI-assisted outputs were checked against the research record and source literature, and the authors take responsibility for the final scientific claims, experiments, analyses, interpretations, and manuscript.

REFERENCES

Matias D. Cattaneo and Boris Shigida. How memory in optimization algorithms implicitly modifies the loss. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-5215.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1-113, 2023.

Jinhui Guo. Delayed optimizer-state transport shapes short-horizon training decisions. arXiv preprint arXiv:2608.24593, 2026.

Tianjin Huang, Ziquan Zhu, Gaojie Jin, Lu Liu, Zhangyang Wang, and Shiwei Liu. Spam: Spike-aware adam with momentum reset for stable llm training. In *International Conference on Learning Representations*, 2025. arXiv:2501.06842.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1–2):99–134, 1998. doi: 10.1016/S0004-3702(98)00023-X.

Abhay Kumar, Louis Owen, Nilabhra Roy Chowdhury, and Fabian Gura. Zclip: Adaptive spike mitigation for llm pre-training. arXiv preprint arXiv:2504.02507, 2025.

Arthur Lebeurrier, Titouan Vayer, and Remi Gribonval. Path-conditioned training: A principled way to rescale relu neural networks. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of Proceedings of Machine Learning Research, 2026. arXiv:2602.19799.

Kaizhao Liang, Lizhang Chen, Bo Liu, and Qiang Liu. Cautious optimizers: Improving training with one line of code. In *International Conference on Learning Representations*, 2026. arXiv:2411.16085.

Zhengzhong Liu, Bowen Tan, Hongyi Wang, et al. LLM360 K2: Building a 65B 360-open-source large language model from scratch. arXiv preprint arXiv:2501.07124, 2025.

Barak Or. Automatic stability and recovery for neural network training. arXiv preprint arXiv:2601.17483, 2026.

Vasileios Sevetlidis and George Pavlidis. Process-tensor tomography of sgd: Measuring non-markovian memory via back-flow of distinguishability. In *Proceedings of the 29th International Conference on Artificial Intelligence and Statistics*, volume 300 of Proceedings of Machine Learning Research, pp. 2521-2529, 2026.

Xiao Shou, Yanna Ding, and Jianxi Gao. Gradient flow matching for learning update dynamics in neural network training. arXiv preprint arXiv:2505.20221, 2025.

Devender Singh. The loss does not see the basis, but adam does. arXiv preprint arXiv:2608.05136, 2026.

Fanqi Wang, Weisheng Tang, Landon Harris, Hairong Qi, Dan Wilson, and Igor Mezic. Predictive differential training guided by training dynamics. In *International Conference on Learning Representations*, 2026.

Qinyang Xu. Stored in optimizer state, valued by later training: A causal account of subliminal trait transfer. arXiv preprint arXiv:2608.20442, 2026.

Susan Zhang, Stephen Roller, Naman Goyal, et al. Opt: Open pre-trained transformer language models. arXiv preprint arXiv:2205.01068, 2022.

Bo Zhao, Nima Dehmamy, Robin Walters, and Rose Yu. Symmetry teleportation for accelerated optimization. In *Advances in Neural Information Processing Systems*, volume 35, 2022.