How to Train Your Energy-Based Transformer: Understanding Stability in EBMs

Anonymous authors
Paper under double-blind review

Abstract

Diffusion and autoregressive models generate images, video, and text at scale. Energy-based models (EBMs) cast generation as energy minimization. They learn a scalar energy that scores how well a candidate output matches the data, and energy-based transformers (EBTs) scale this approach by implementing the energy with a transformer. An EBT generates by refining a prediction through a sequence of gradient descent updates on the energy, starting from a noisy or corrupted input. Training differentiates through this refinement sequence, the same computation as backpropagation through time (BPTT) in recurrent networks, and it is often unstable. We analyze this instability. Each backward step multiplies the gradient by a factor set by the step size and by the curvature of the energy with respect to the prediction. We bound the resulting growth and convert the bound into a curvature score computed during training. We compare full BPTT with two lower-cost gradient estimators, truncated BPTT, which differentiates only the last few updates, and Anticipated Reweighted Truncated Backpropagation (ARTBP), which cuts the sequence at random points and reweights the surviving terms to remove the bias introduced by the cut. In image denoising with three model sizes, we find that the best estimator changes with scale. ARTBP gives the highest final training reconstruction quality with the small model, while truncation to the last three updates is best for the base and large models. With the objective and forward trajectory fixed, we find that the truncated gradient changes from closely aligned with the full gradient early in training to pointing in the opposite direction later. The curvature score separates failed from successful completed runs and predicts training stability over a horizon that shrinks with model size.

1 Introduction

Diffusion and autoregressive models generate images, video, and text at scale (Ho et al., 2020; Brown et al., 2020), and most of their computation is fixed once training ends. A complementary line of work increases the computation a model spends at inference time, letting it verify and revise a prediction before returning it, which is a basic form of reasoning (Gladstone et al., 2025). Iterative refinement of this kind requires a component that scores how good a candidate prediction is.

Energy-based models (EBMs) provide this component. An EBM learns a scalar energy that scores how well a candidate output matches the data, and it generates by lowering the energy of a candidate with gradient descent (LeCun et al., 2006; Du & Mordatch, 2019; Nijkamp et al., 2019; Song & Kingma, 2021). Inference is optimization, so additional updates can improve the prediction. Energy-based transformers (EBTs) implement the energy with a transformer and scale this approach across images, video, and text (Gladstone et al., 2025).

The obstacle is training. An EBT learns by differentiating the reconstruction loss through its own inference updates, the same computation as backpropagation through time (BPTT) in recurrent networks. Each backward step multiplies the gradient by the derivative of one update, which depends on the local curvature of the energy with respect to the prediction. When step sizes and curvature are large, repeated backward steps amplify the gradient until training collapses, and in our experiments this failure becomes harder to avoid as the model grows (Figure 1). Training stability is therefore a core challenge in scaling EBTs.

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.

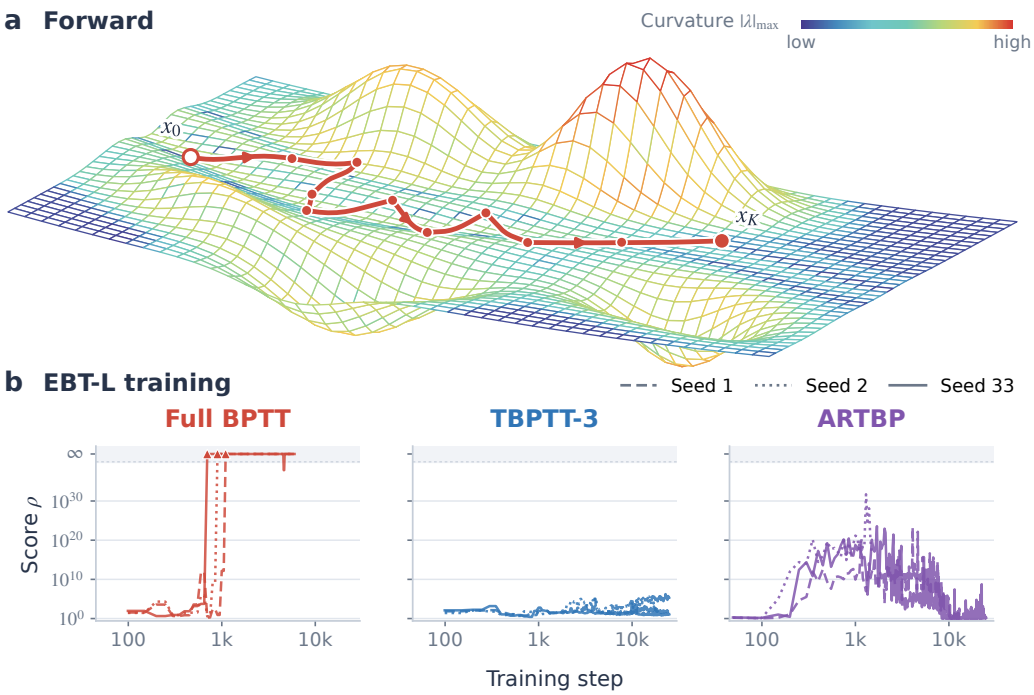


Figure 1: **Curvature geometry and measured training scores.** Above, ten descent updates on a synthetic energy surface colored by its largest absolute Hessian eigenvalue, the curvature quantity that sets backward gradient growth (Section 2). Below, the live curvature score ρ of Equation (3), recorded during training for three seeds of each EBT-L recipe under the original trainer, on shared log scales. Full BPTT uses frozen $\alpha = 300$, while TBPTT-3 and ARTBP learn α. Curves end at the last logged step, and ∞ denotes numerical overflow. Full BPTT overflows, ARTBP spikes and later falls in its longer runs, and TBPTT-3 stays lower. The ARTBP runs whose scores later fall have already failed (Table 1). Optimizer and scheduler step counts and recipe differences are detailed in Appendix B.

The choice of backward computation affects cost, gradient fidelity, and stability. Keeping more steps preserves more of the gradient, but those steps can also amplify it. Cutting earlier steps reduces computation and removes some gradient contributions. As training changes the energy and its step sizes, the contributions removed by the same cut can change. This raises three questions. Which training methods work across model sizes? How does truncation change the gradient during training? Can the curvature of the energy with respect to the prediction identify instability before collapse?

We study these questions in image denoising with small, base, and large EBT configurations. We find that the training methods change rank across sizes. In a matched comparison that fixes the objective and the forward computation at each checkpoint, the gradient retained by truncation reverses direction over training, from closely aligned with the full gradient to opposed to it. We then examine curvature as a diagnostic. Completed runs separate clearly by their retrospective curvature score. Predicting future failure requires measurements taken before collapse, so we test both advance warnings and the duration over which low-score predictions remain valid.

The paper proceeds as follows. Section 2 defines the model, the forward trajectory, and the gradient estimators, shows how each backward step depends on the curvature of the energy with respect to the prediction, and reviews related work. Section 3 analyzes the backward dynamics and converts them into the curvature score ρ and its prediction horizon. Section 4 reports training outcomes across model sizes, the fixed-objective gradient comparison, and the prediction results. Section 5 concludes.

1.1 CONTRIBUTIONS

1. **Training outcomes across model sizes.** In a comparison where only the gradient estimator differs, randomized truncation with unbiased reweighting (ARTBP) gives the highest reconstruction quality with EBT-S, truncation to the last three updates (TBPTT-3) gives the highest with EBT-B and EBT-L, and full BPTT fails at both larger sizes. We report all seed outcomes.
2. **Changes in the retained gradient.** With the loss and forward trajectory fixed within each checkpoint, the gradient terms kept after truncation change from aligned with the full gradient to opposed to it over training. No reweighting of the kept terms recovers the dropped information late in training.
3. **Curvature as a stability diagnostic and predictor.** We introduce a curvature score ρ that combines the step size, the largest Hessian eigenvalue of the energy with respect to the prediction, and the backward depth (Section 3). We evaluate it by completed-run separation, future-collapse ranking, warning time, and the duration over which a low reading remains valid, with held-out runs and matched baselines.

2 BACKGROUND

2.1 ENERGY-BASED TRANSFORMERS

An EBT learns an energy $E_\theta(x)$, a scalar function of a candidate prediction x, implemented by a transformer with parameters θ. Training shapes the energy so that accurate predictions receive low energy. We study image denoising. Let y be a clean image and let its corrupted version initialize the prediction at x_0. Inference refines the prediction through K gradient descent updates on the energy,

$$x_{k+1} = N_{k+1}(x_k - \alpha_{k+1}\nabla_x E_\theta(x_k)), \qquad k = 0, \dots, K-1, \tag{1}$$

where α_{k+1} is a step size and N_{k+1} normalizes the channels at each pixel. For a pixel vector $u \in \mathbb{R}^c$ with c channels, the normalization is $N(u) = u/\sqrt{\|u\|_2^2/c + \epsilon}$, which divides the pixel by its root mean square channel value and keeps intermediate predictions on a common scale (Appendix A.3). Normalization is usually omitted at the final output. We call $x_0, \dots, x_K$ the *forward trajectory*. Training minimizes a reconstruction loss $L = \sum_k w_k \, \ell(x_k, y)$ that compares outputs of the trajectory with the clean image, with weights w_k set by the training method.

2.2 DIFFERENTIATING THROUGH INFERENCE

Every update in Equation (1) depends on θ through the energy gradient, so the loss depends on θ both directly at each supervised output and through the whole trajectory that produced it. The exact parameter gradient therefore requires the chain rule backward through all K updates. This is BPTT with inference updates in place of time steps. Its memory and computation grow with K, and, as we show, the repeated backward steps can also amplify the gradient.

2.3 PARAMETER GRADIENT ESTIMATORS

Full BPTT differentiates through all K updates. Truncated BPTT (TBPTT-M) stops the backward pass after the final M updates by detaching x_{K-M}, which drops every gradient term that flows through earlier updates. Anticipated reweighted truncated backpropagation (ARTBP) instead cuts the backward pass at random positions and divides each surviving term by its probability of surviving, which makes the estimate unbiased for the full gradient (Tallec & Ollivier, 2017). Randomized telescoping extends this bias correction to computations of unbounded length (Beatson & Adams, 2019). Appendix A states the exact gradient and the error introduced by a cut.

2.4 BACKWARD STEPS AND CURVATURE

Backpropagation applies the transpose of each update Jacobian $J_k = \partial x_{k+1}/\partial x_k$ to the gradient flowing backward. For the update in Equation (1),

$$J_k = DN_{k+1}\,(I - \alpha_{k+1}H_k), \qquad H_k = \nabla_x^2 E_\theta(x_k), \tag{2}$$

where DN_{k+1} is the derivative of the normalization and H_k is the Hessian of the energy with respect to the *prediction*. Throughout the paper, curvature refers to this prediction Hessian, not to curvature in parameter space. Its eigenvalues measure how fast the energy gradient changes along the corresponding eigenvectors, and they determine whether a backward step shrinks or amplifies the gradient.

2.5 Related Work

Differentiating through optimization also appears in learned iterative models, hyperparameter optimization, and meta-learning (Maclaurin et al., 2015; Shaban et al., 2019; Lorraine et al., 2020), and long products of update Jacobians are a known source of poorly conditioned gradients in learned optimizers (Metz et al., 2021). Several methods avoid the unrolled computation entirely. Deep equilibrium models differentiate through a fixed-point equation (Bai et al., 2019), the implicit function theorem obtains the same gradient from the sensitivity of the solution (Rajeswaran et al., 2019; Lorraine et al., 2020), and Jacobian regularization and monotone parameterizations improve the stability of these implicit gradients (Bai et al., 2021; Winston & Kolter, 2020). Predictive coding and equilibrium propagation instead update parameters from local signals at an inference equilibrium (Rao & Ballard, 1999; Scellier & Bengio, 2017; Innocenti et al., 2025). The EBT runs studied here use finite refinement sequences, and Appendix A.5 connects the finite-chain gradient to the equilibrium gradient under a stationary, contractive linearization. Closest to our analysis, the edge-of-stability literature relates parameter-space curvature to the gradient-descent step size (Cohen et al., 2021), and recent transformer work studies curvature during training (Ramasinghe et al., 2026). The energy landscape of predictive coding inference has likewise been analyzed through its saddle structure and its infinite width and depth limits (Innocenti et al., 2024; 2026), and the inference itself has been interpreted as an adaptive trust-region method (Innocenti et al., 2023).

3 Method

3.1 Analysis of Backward Dynamics

Consider descent without normalization on a locally quadratic energy, so that the analysis is exact for a convex quadratic. By Equation (2), a gradient component along an eigenvector with curvature λ is multiplied by $1 - \alpha\lambda$ at each backward step. The same factor governs the forward iterates, and gradient descent itself diverges once the step size exceeds $2/\lambda_{\max}$, where $\lambda_{\max}$ is the largest curvature. The backward component is amplified whenever $|1 - \alpha\lambda| > 1$, which happens for positive curvature once $\alpha\lambda > 2$ and for negative curvature always. Over M retained backward steps the amplification compounds geometrically. Appendix A.3 extends the multiplier to normalized updates (Equation (9)), and Equation (7) bounds the error introduced by a backward cut. Figure 1 pairs a descending forward trajectory with measured curvature-score histories during training.

The gradient describes how changing the model parameters changes the reconstruction loss. Full backpropagation adds contributions from every inference update. A backward cut keeps the contributions from recent updates and removes the earlier ones. Their directions and magnitudes depend on the learned model, so a fixed backward depth can retain different information at different points in training.

3.2 Curvature Score

The monitor estimates the largest positive Hessian eigenvalue $\lambda^{+}_{\max}$ at the final prediction using Hessian-vector products (Pearlmutter, 1994). Because $|1 - \alpha\lambda| \leq 1 + \alpha\lambda$ for $\lambda \geq 0$, a backward step can grow a positive-curvature component by at most $1 + \alpha\lambda$, and M retained steps compound this factor. With a constant step size, the score is

$$\rho = (1 + \alpha\lambda^{+}_{\max})^{M}, \tag{3}$$

with the measured curvature floored at 10^{-12}. The score upper-bounds the backward growth from positive curvature and combines the step size, the curvature, and the backward depth. When $\alpha\lambda^{+}_{\max}$ is small, each factor is close to one and ρ stays near one, so gradients pass through the backward steps without growth. When $\alpha\lambda^{+}_{\max}$ is large, the factors compound over the M retained steps and ρ grows

geometrically, the regime where backpropagated gradients can explode. We therefore evaluate ρ as an empirical predictor of training stability. With a learned schedule, it multiplies $1 + \alpha_k \lambda^+_{\max}$ over the final M entries. Step sizes have a 10^{-4} floor. ARTBP uses its rounded nominal segment length for M. The full update derivative also includes normalization and negative-curvature directions (Appendix A).

3.3 Prediction Horizon

We next define how long a low score stays informative. Suppose the monitor reads $\rho < 10^9$ at training step t in a run that is training normally, with eligibility rules in Section 4.1. We interpret this reading as a prediction that the run will not collapse during the next h training steps, and we ask for the largest h at which such predictions are reliable.

Each low reading opens a window $(t, t + h]$. A window counts as evaluated when its outcome is known, either because a collapse is recorded inside it or because the run is observed through its end. Let $n(h)$ be the number of evaluated windows of length h, and let $c(h)$ be the number that contain a collapse. The empirical survival rate and the selected horizon are

$$\widehat{S}(h) = 1 - \frac{c(h)}{n(h)}, \qquad \widehat{H}_{99} = \max\left\{h \in \mathcal{H} :\ n(h) \geq 30,\ \widehat{S}(h) \geq 0.99\right\}, \tag{4}$$

where $\mathcal{H}$ is a fixed grid of window lengths from 100 to 40,000 steps. In words, $\widehat{H}_{99}$ is the longest window over which at least 99% of evaluated low-score predictions held, with at least 30 windows required for a usable estimate. Training keeps changing the energy surface, so a reading loses predictive value over time, and $\widehat{H}_{99}$ measures how long it remains valid.

4 Experiments

4.1 Experimental Setup

We denoise ImageNet images resized to 64×64 (Deng et al., 2009), using small (EBT-S), base (EBT-B), and large (EBT-L) energy-based transformers. Their architecture settings and parameter counts are listed in Table 3.

The main comparison uses $K = 10$. The corruption setting is a diffusion-schedule fraction of 0.5. Architecture, optimization, and evaluation settings appear in Appendix B.

A training recipe specifies the gradient estimator, the placement of the smooth-ℓ_1 losses over outputs, and the step-size treatment. In the main comparison, TBPTT-3, ARTBP, and full BPTT with learned step sizes share every setting and differ only in the estimator and its per-output loss weighting. Full BPTT averages the losses over all outputs, ARTBP targets their reweighted sum, and TBPTT sums losses over its retained suffix. ARTBP's reweighting is unbiased for its target before gradient clipping. The remaining recipes are earlier fixed-step-size variants that also differ in loss placement and normalization. Results from the original ARTBP trainer, which had unequal scheduler and accumulation behavior, are reported separately in Appendix B. The gradient comparison in Section 4.3 fixes the objective and forward trajectory within each checkpoint. Exact losses, settings, and observation budgets are given in Appendix B.

The fixed-seed comparison contains 69 runs across 23 configurations, with three seeds per configuration. One EBT-L run, interrupted nonfailed by a cluster time limit at about one third of its budget, is excluded from means and failure counts (Appendix B). The monitoring cohort contains 72 runs, including 37 from the fixed-seed comparison, and was locked before the matched-estimator reruns, so no prediction result depends on them. It contains 31 failed and 41 nonfailed runs.

We use peak signal-to-noise ratio (PSNR), a reconstruction measure for which larger values are better. A run fails if its final training PSNR is below $8\,\mathrm{dB}$ or more than $3\,\mathrm{dB}$ below its previous maximum. Collapse times are defined by a sustained PSNR drop, as specified in Appendix C. Prediction windows start only after past PSNR has exceeded $9\,\mathrm{dB}$, while current PSNR remains at least $8\,\mathrm{dB}$ and within $3\,\mathrm{dB}$ of its past maximum. All predictors use measurements available by the prediction time. Training-batch PSNR diagnoses training behavior, and held-out image quality is reported separately in Appendix B.

Table 1: **Training outcomes across model sizes.** Entries give mean final training PSNR in dB and failed runs among three seeds. Means include failures. The upper rows share identical settings and differ only in the gradient estimator and its per-output loss weighting. The lower rows are earlier fixed-step-size recipes that also differ in loss placement and normalization. "Final norm." retains normalization at the last output. ‡The third seed was interrupted nonfailed at about one third of its budget and is excluded. Per-seed outcomes, original-trainer ARTBP results, and observation budgets appear in Appendix B.

Recipe	Step sizes	EBT-S	EBT-B	EBT-L
TBPTT-3	Learned	8.3 / 1	**15.7 / 0**	**15.9 / 0**
ARTBP	Learned	**16.4 / 0**	9.8 / 2	6.7 / 2‡
Full BPTT	Learned	13.4 / 0	6.2 / 3	5.9 / 3
TBPTT-1	Fixed	8.5 / 0	14.7 / 0	—
TBPTT-5	Fixed	8.5 / 1	6.6 / 3	6.9 / 3
Full BPTT, final norm.	Fixed	11.7 / 0	8.2 / 2	6.2 / 3
Full BPTT	Fixed	6.2 / 3	6.2 / 3	6.0 / 3

4.2 Estimator Performance Across Model Sizes

Table 1 compares training recipes across model sizes. Its upper block holds every setting fixed except the gradient estimator. We find that the estimators change rank with scale. ARTBP gives the highest mean training PSNR with EBT-S and avoids failure in all three seeds. TBPTT-3 gives the highest mean with EBT-B and EBT-L, again with no failures. ARTBP fails in two of three seeds with EBT-B and in both completed EBT-L runs, and full BPTT trains with EBT-S but fails in every run at the two larger sizes. An estimator that works well with the small model can therefore be a poor choice for the larger models.

The amount of retained history and the step-size treatment both have large effects. With EBT-B, TBPTT-1 and TBPTT-3 avoid failure in every seed, while TBPTT-5 fails in all three. With EBT-S, letting full BPTT learn its step sizes raises it from 6.2 with three failures to 13.4 with none, yet the same change leaves it failing in every EBT-B and EBT-L run. Final-output normalization also improves fixed-step-size full BPTT, especially with EBT-S, but fails in every EBT-L run. More backward steps, learned step sizes, and final normalization are therefore insufficient for a stable recipe across sizes.

Some failures follow an extended period of normal training. A monitored EBT-B TBPTT-5 run collapses after about 18,000 training steps, and two EBT-L runs collapse after about 7,800. One matched-recipe full BPTT run with EBT-B reaches 16.3 dB before collapsing near 26,000 steps. Short pilot runs can miss these changes. The next section examines what happens to the gradient when training continues. Per-seed outcomes and the measured computation costs are in Appendix B.

4.3 Gradient Changes During Training

We compare two checkpoints of one EBT-B model, at 5,000 and 50,000 training steps. Within each checkpoint, every gradient uses the same 12-update forward trajectory and the same loss on the last three outputs. We vary only the retained backward contributions. The comparison uses eight paired validation batches. A contribution's age indexes its update backwards from the final update, and the three youngest form the truncated gradient.

We measure alignment by the cosine between the truncated and full gradients. A cosine of one means the same direction, zero means perpendicular directions, and minus one means opposite directions. The mean changes from 0.971 early in training to -0.693 later (Figure 2). All eight batches show this change. The same three-update window therefore changes from a close directional approximation to a gradient that points against the full gradient.

We also test whether changing the weights of the three retained contributions can recover the full gradient. The best weights for each batch give mean relative error 0.104 at 5k and 0.638 at 50k. Relative error divides the reconstruction error by the full-gradient norm, so zero is exact recovery. We then fit weights on seven batches and test the excluded batch. The errors are 0.110 and 0.662,

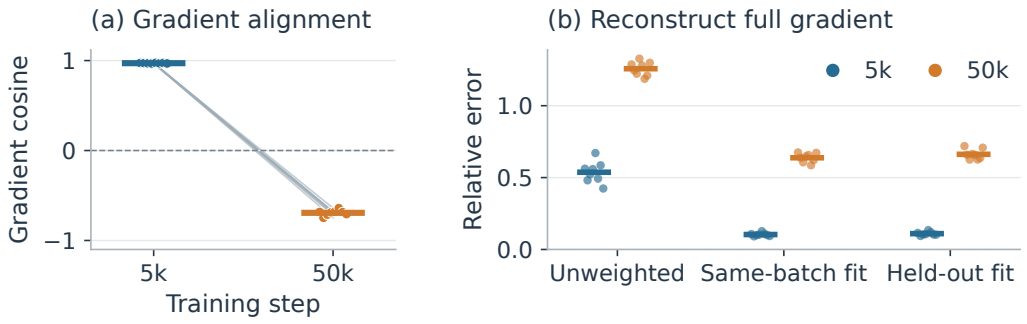


Figure 2: **Short backward paths change during training.** Points show eight paired batches at 5k and 50k steps, and horizontal bars show means. Left: cosine between the full gradient and the sum of the three youngest contributions, with matched batches connected. Right: relative reconstruction error using unit weights, best-fitting weights for the same batch, or weights fitted on the other seven batches. The objective and forward trajectory are fixed within each checkpoint.

respectively. Both tests show that reweighting the retained contributions leaves much more error later in training. The saved gradient inner products, fitting procedure, and decomposition checks are given in Appendix D.

Learned Step Sizes. Training also changes how far each inference update moves. In one EBT-B ARTBP run, eight of ten step sizes reach the 10^{-4} floor by step 10,000, while the first and last remain large. TBPTT-3 leaves the seven entries outside its backward window at initialization. These schedules show that methods with the same nominal inference length can learn different update magnitudes. The complete trajectories and the derivative of the step-size clamp appear in Appendix D.

4.4 Predicting Training Failure

We next test whether ρ read during training, the live score, ranks future failures at a fixed observation time. At step 1,000, 55 runs have all required measurements and an observable next-5,000-step outcome, and 21 collapse within that interval. We compare ρ, nominal backward depth, curvature alone, and lower current PSNR on these identical runs. Area under the receiver operating characteristic curve (AUC) measures how often a collapsing run receives a higher risk score than a noncollapsing run, and 0.5 corresponds to chance.

At this observation time, ρ and nominal backward depth rank future failures comparably, with AUC 0.728 and 0.732, and both exceed curvature alone at 0.599 and current PSNR at 0.571 (Figure 3). Resampling complete size–configuration groups gives a 95% interval of $[0.558, 0.876]$ for ρ. The stronger test is incremental. We fit a fixed regularized logistic regression on model size and backward depth, then add $\log \rho$, holding out entire size–configuration groups and fitting preprocessing only on the training groups. Adding the score raises out-of-fold AUC from 0.598 to 0.686 and lowers Brier error from 0.247 to 0.239. Lower Brier error means better probability predictions. Additional observation times, the full protocol, and paired uncertainty are in Appendix C.

To measure advance warning, we test the pre-specified threshold $\rho > 10^{16}$ at each eligible live reading. It warns before 18 of 29 collapses and produces four false alarms among 37 nonfailed runs. Only five collapses have at least 250 steps of warning, four have at least 500, and three have at least 1,000. Most detections occur close to the recorded collapse. Advance warning in this cohort is therefore limited to short leads, and its 29 collapses include only three after step 5,000.

Retrospective Stability Boundaries. The largest retrospective controller score over a completed run separates the 31 failed from the 41 nonfailed runs. The nonfailed maximum is 3.77×10^{13} and the failed minimum is 4.48×10^{27}. This score uses a smoothed controller state, described in Appendix C. Four matched-estimator reruns monitored after the cohort was locked give an out-of-sample check.

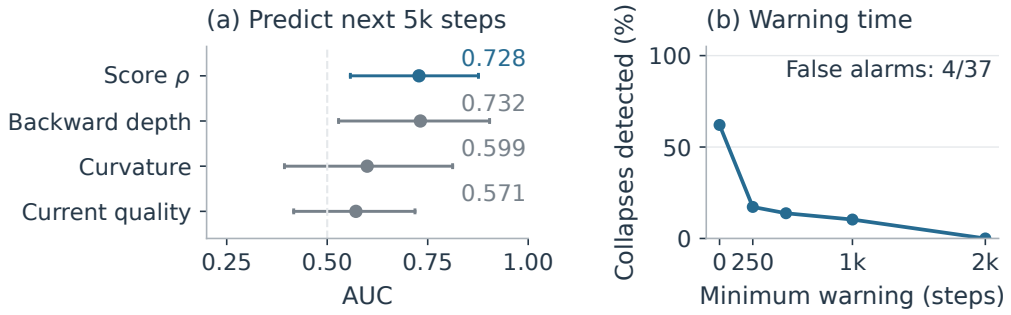


Figure 3: **Future-collapse prediction and warning time.** Left: AUC at training step 1,000 for collapse within the next 5,000 steps, using the same 55 runs for every predictor. Bars are 95% intervals from resampling size–configuration groups. Right: the fraction of 29 observed collapses preceded by $\rho > 10^{16}$, as the required warning time increases. There are four false alarms among 37 nonfailed runs.

The two runs reading below 10^9 satisfied the low-score rule and survived, but one surviving run reached 4.3×10^{24}, above the nonfailed ceiling, so this boundary is specific to the cohort and is measured only after outcomes are known (Appendix C.3). Prediction must instead use the live score recorded before each outcome, which we evaluate next.

4.5 Prediction Horizons

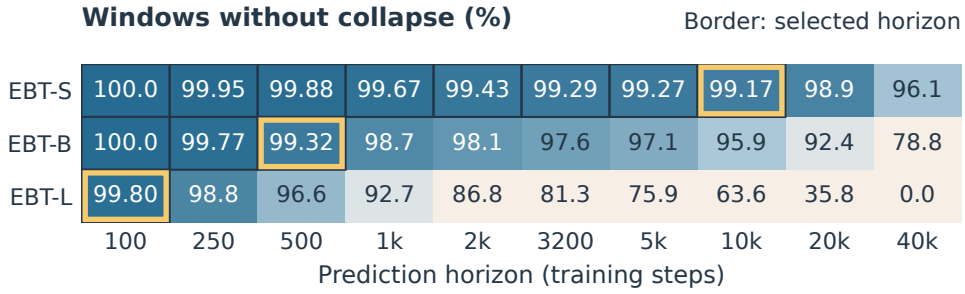


Figure 4: **Validity duration of low-score predictions.** Cells give the percentage of evaluated low-score windows without collapse. Dark borders mark cells meeting the 99% target with at least 30 windows, and gold borders mark the largest qualifying horizon. Rows use the same live score and threshold, $\rho < 10^9$. Longer windows can have different contributing runs because follow-up requirements change.

The selected horizons are **10,000, 500, and 100 training steps** for EBT-S, EBT-B, and EBT-L models (Figure 4). At these horizons, 86 of 10,413, 107 of 15,778, and 4 of 1,971 windows contain a collapse. They come from 19, 34, and 13 distinct runs, including 4, 15, and 2 runs that contribute collapse-containing windows.

For validation, we exclude one run, select the horizon on the remaining runs of the same size, and test the excluded run. Pooled window survival remains above 99% at all sizes (Table 2). The comparison without score filtering uses the same eligibility rules and fold-selected horizons. Filtering improves survival at each size. It also extends the fitted EBT-S horizon from 2,000 to 10,000 steps.

Window weighting determines the horizon. Long surviving runs contribute many windows, so the per-window estimate describes logged windows, while equal-run weighting gives more influence to short runs that later collapse and selects 100 steps at every size, with survival of 78.95%, 79.49%,

Table 2: **Validation of low-score prediction horizons.** Horizons are measured in training steps. Held-out survival pools test windows after selecting the horizon without the evaluated run. The no-score comparison uses the same fold-selected horizons and eligibility rules. The last column selects the horizon with equal weight for each contributing run. Complete-run bootstrap intervals appear in Appendix C.

Size	Fitted $\widehat{H}_{99}$	Held-out survival	No-score survival	Equal-run $\widehat{H}_{99}$
EBT-S	10,000	99.17%	98.63%	100
EBT-B	500	99.32%	99.07%	100
EBT-L	100	99.49%	98.70%	100

and 99.64% at the longer per-window horizons. The two weightings answer different questions, per-window and per-run reliability, and we report both. Uncertainty is estimated by resampling complete runs, with intervals and full counts in Appendix C.

4.6 Exploratory Interventions

Targeted interventions test whether perturbing the inference updates changes the observed outcomes. With EBT-B, a recipe that perturbs step sizes, adds noise, and repeats the final update leaves two of three TBPTT-5 runs nonfailed, while all seven deterministic TBPTT-5 runs in the broader comparison fail. Three stochastic TBPTT-3 runs finish near $14.1\,\mathrm{dB}$, below the deterministic mean of $15.7\,\mathrm{dB}$. The perturbations prevent failure for TBPTT-5 and slightly reduce final quality for TBPTT-3. Settings and individual outcomes are in Appendix D.

5 Discussion and Conclusion

Training recipes for EBTs should be evaluated across model sizes and training stages. In the matched comparison, ARTBP gives the highest reconstruction quality with EBT-S, truncation to three updates gives the highest with EBT-B and EBT-L, and full BPTT fails at both larger sizes even with learned step sizes. The fixed-objective probe explains why truncation-based recipes require this evaluation. The gradient terms kept by a short backward window change from closely aligned with the full gradient to opposed to it during training, and no reweighting of the kept terms recovers the dropped information. The retrospective curvature score separates completed outcomes, the live score supports measurable low-score prediction horizons, and the low-score screen transfers to held-out reruns. Advance warning of late collapse remains open, since most detections arrive near the collapse itself. Curvature monitors should therefore be judged by prediction horizon and warning time in addition to their separation of completed outcomes.

AI USE STATEMENT

The research questions, hypotheses, experimental design, and conclusions are the authors' own. Under the authors' direction, AI tools assisted in implementing parts of the analysis and plotting code, provided feedback during research ideation, assisted in proving mathematical claims formulated by the authors, and drafted and polished manuscript text from the authors' outlines and results. We did not use these tools to generate synthetic data sets or for literature retrieval. The authors verified all derivations manually, checked AI-assisted code against logged experiment records, traced every reported number to its run records, and take responsibility for the final content.

REPRODUCIBILITY STATEMENT

Appendix B specifies all architecture, optimization, data, and recipe settings, with per-seed outcomes and observation budgets in Table 5. Appendix C defines the score computation, collapse times, eligibility rules, and the selection and validation protocol behind every prediction number. Appendix A states all assumptions and derivations, and Appendix D details the gradient probes and interventions.

REFERENCES

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. Deep equilibrium models. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Shaojie Bai, Vladlen Koltun, and J. Zico Kolter. Stabilizing equilibrium models by jacobian regularization. In *International Conference on Machine Learning*, 2021.

Alex Beatson and Ryan P. Adams. Efficient optimization of loops and limits with randomized telescoping sums. In *International Conference on Machine Learning*, 2019.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Jeremy M. Cohen, Simran Kaur, Yuanzhi Li, J. Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations*, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Yilun Du and Igor Mordatch. Implicit generation and modeling with energy-based models. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Alexi Gladstone, Ganesh Nanduru, Md Mofijul Islam, Peixuan Han, Hyeonjeong Ha, Aman Chadha, Yilun Du, Heng Ji, Jundong Li, and Tariq Iqbal. Energy-based transformers are scalable learners and thinkers. *arXiv preprint arXiv:2507.02092*, 2025.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Francesco Innocenti, Ryan Singh, and Christopher L. Buckley. Understanding predictive coding as an adaptive trust-region method. *arXiv preprint arXiv:2305.18188*, 2023.

Francesco Innocenti, El Mehdi Achour, Ryan Singh, and Christopher L. Buckley. Only strict saddles in the energy landscape of predictive coding networks? In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Francesco Innocenti, El Mehdi Achour, and Christopher L. Buckley. μPC: Scaling predictive coding to 100+ layer networks. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Francesco Innocenti, El Mehdi Achour, and Rafal Bogacz. On the infinite width and depth limits of predictive coding networks. In *International Conference on Machine Learning*, 2026.

Yann LeCun, Sumit Chopra, Raia Hadsell, Marc'Aurelio Ranzato, and Fu Jie Huang. A tutorial on energy-based learning. In *Predicting Structured Data*. MIT Press, 2006.

Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *International Conference on Artificial Intelligence and Statistics*, 2020.

Dougal Maclaurin, David Duvenaud, and Ryan P. Adams. Gradient-based hyperparameter optimization through reversible learning. In *International Conference on Machine Learning*, 2015.

Luke Metz, C. Daniel Freeman, Samuel S. Schoenholz, and Tal Kachman. Gradients are not all you need. *arXiv preprint arXiv:2111.05803*, 2021.

Erik Nijkamp, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. Learning non-convergent non-persistent short-run MCMC toward energy-based model. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Barak A. Pearlmutter. Fast exact multiplication by the hessian. *Neural Computation*, 6(1):147–160, 1994.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Aravind Rajeswaran, Chelsea Finn, Sham M. Kakade, and Sergey Levine. Meta-learning with implicit gradients. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Sameera Ramasinghe, Ajanthan Thalaiyasingam, Hadi Mohaghegh Dolatabadi, Chamin Hewa Koneputugodage, Gil Avraham, Violetta Shevchenko, Yan Zuo, Karol Pajak, and Alexander Long. Taming curvature: Architecture warm-up for stable transformer training. *arXiv preprint arXiv:2606.16768*, 2026.

Rajesh P. N. Rao and Dana H. Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2(1):79–87, 1999.

Benjamin Scellier and Yoshua Bengio. Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. *Frontiers in Computational Neuroscience*, 11:24, 2017.

Amirreza Shaban, Ching-An Cheng, Nathan Hatch, and Byron Boots. Truncated back-propagation for bilevel optimization. In *International Conference on Artificial Intelligence and Statistics*, 2019.

Yang Song and Diederik P. Kingma. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*, 2021.

Corentin Tallec and Yann Ollivier. Unbiasing truncated backpropagation through time. *arXiv preprint arXiv:1705.08209*, 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.

Ezra Winston and J. Zico Kolter. Monotone operator equilibrium networks. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

A THEORY

A.1 FINITE-CHAIN GRADIENTS

Write one inference update as $x_{k+1} = F_k(x_k, \theta)$ and its reconstruction objective as $L = \sum_{k=1}^{K} w_k \ell(x_k, y)$. Here k indexes inference updates and t indexes training steps. Define $J_k = \partial x_{k+1}/\partial x_k$, $C_k = \partial x_{k+1}/\partial \theta$ with x_k fixed, and $v_k = \nabla_{x_k} \ell(x_k, y)$. The adjoint b_k is the derivative of the remaining objective with respect to x_k. For fixed loss weights and an initializer independent of θ, it satisfies

$$b_K = w_K v_K, \qquad b_k = w_k v_k + J_k^\top b_{k+1}, \qquad \nabla_\theta L = \sum_{k=0}^{K-1} C_k^\top b_{k+1}, \tag{5}$$

with $w_0 v_0 = 0$. This follows by substituting $dx_{k+1} = J_k dx_k + C_k d\theta$ into dL and collecting the coefficient of each state differential backwards. A loss or initializer with direct parameter dependence adds its corresponding derivative.

The weights determine which outputs receive supervision, and detachment determines which dependencies are differentiated. In the recorded recipes, TBPTT-1 uses only the final loss, TBPTT-M uses the last M losses for $M > 1$, full BPTT averages all losses, and ARTBP targets their sum. These are different objectives even when they share a forward trajectory.

A.2 TRUNCATION ERROR FOR A FIXED OBJECTIVE

Proposition 1 (A cut on a finite trajectory)**.** *For a suffix objective $L = \sum_{s=q+1}^{K} w_s \ell(x_s, y)$, let $S_q = dx_q/d\theta$ through the prefix. Let g differentiate L through the complete chain and g_{cut} differentiate the same L with x_q detached. Then*

$$g - g_{\mathrm{cut}} = S_q^\top b_q, \tag{6}$$

$$\|g - g_{\mathrm{cut}}\|_2 \le \|S_q\|_2 \sum_{s=q+1}^{K} |w_s| \|v_s\|_2 \prod_{j=q}^{s-1} \|J_j\|_2. \tag{7}$$

Proof. The suffix depends on parameters both directly and through $x_q(\theta)$. Detachment removes only the latter derivative, $S_q^\top \nabla_{x_q} L$. Expanding the adjoint gives $b_q = \sum_{s=q+1}^{K} w_s (J_{s-1} \cdots J_q)^\top v_s$. The triangle inequality and submultiplicativity give the bound. □

For a terminal loss, if the last $M = K - q$ Jacobians satisfy $\|J_j\|_2 \le \gamma < 1$, the bound becomes $\|S_q\|_2 \gamma^M \|v_K\|_2$. This statement requires the actual update Jacobians and controlled prefix sensitivity. Changing the supervised outputs adds an objective difference that is absent from this identity.

A.3 CURVATURE AND NORMALIZATION

For plain descent followed by normalization, let $u_{k+1} = x_k - \alpha_{k+1} \nabla_x E_\theta(x_k)$ and $x_{k+1} = N_{k+1}(u_{k+1})$. Holding the scalar step size fixed with respect to the state,

$$J_k = DN_{k+1}(u_{k+1})(I - \alpha_{k+1} H_k), \qquad H_k = \nabla_x^2 E_\theta(x_k). \tag{8}$$

For symmetric H and $\alpha > 0$, orthogonal diagonalization gives

$$\|I - \alpha H\|_2 = \max\{|1 - \alpha \lambda_{\min}(H)|, |1 - \alpha \lambda_{\max}(H)|\} \le 1 + \alpha r, \tag{9}$$

where $r = \max_i |\lambda_i(H)|$. Negative curvature gives a multiplier above one, while positive curvature expands only when $\alpha\lambda > 2$. Thus a path norm is bounded by the product of $\|DN_{k+1}\|_2(1 + \alpha_{k+1} r_k)$. The bound uses curvature and normalization at every update, while the monitored score uses endpoint curvature.

For a pixel vector $u \in \mathbb{R}^c$, channel RMS normalization has

$$N(u) = u/s, \qquad s = \sqrt{\|u\|_2^2/c + \epsilon}, \qquad DN(u) = s^{-1} I - \frac{u u^\top}{c s^3}. \tag{10}$$

Its radial eigenvalue is ϵ/s^3, and its orthogonal eigenvalues are $1/s$. It can amplify perturbations when $s < 1$. In the usual recipe, $N_K = I$, so the final readout differs from the normalized interior updates. For a detached positive diagonal preconditioner D, the unnormalized derivative becomes $I - \alpha DH$. Similarity to $I - \alpha D^{1/2} H D^{1/2}$ preserves eigenvalues, but not Euclidean operator norms.

A.4 Random Truncation and Gradient Variance

Condition on fixed forward randomness and deterministic gradient contributions a_i with $g = \sum_i a_i$. For inclusion indicators I_i, probabilities $p_i > 0$, and joint probabilities p_{ij}, the inverse-probability estimate $\widehat{g} = \sum_i I_i a_i / p_i$ obeys

$$\mathbb{E}\widehat{g} = g, \qquad \mathbb{E}\|\widehat{g} - g\|_2^2 = \sum_{i,j} \left(\frac{p_{ij}}{p_i p_j} - 1 \right) \langle a_i, a_j \rangle. \tag{11}$$

Both identities follow by expanding the expectation and allow dependent inclusion events. ARTBP uses conditional survival compensation (Tallec & Ollivier, 2017). Sampled paths may exceed the nominal segment length. For finite paths with finite contributions and positive inclusion probabilities, variance is finite but can be large. Multiplying a loss by c scales variance by c^2, and nonlinear clipping generally changes the mean.

A.5 The Equilibrium Limit

Near a stationary point, a finite gradient can approach the implicit gradient (Shaban et al., 2019; Lorraine et al., 2020). At an unnormalized energy stationary point, assume a nonsingular symmetric Hessian H, fixed $\alpha > 0$, and constant linearization $A = I - \alpha H$. Let $B = \partial_\theta \nabla_x E \in \mathbb{R}^{d \times p}$ and $v = \nabla_x \ell$. For a terminal loss and detached initial state,

$$g_M = -\alpha B^\top \sum_{j=0}^{M-1} A^j v = -B^\top H^{-1} (I - A^M) v, \qquad g_{\mathrm{IFT}} = -B^\top H^{-1} v. \tag{12}$$

The finite sum uses $I - A = \alpha H$, and differentiating stationarity gives the implicit gradient. Their difference is $B^\top H^{-1} A^M v$, which decays if $0 < \alpha \lambda_i(H) < 2$ for every eigenvalue. For a normalized fixed point $x^* = F_\theta(x^*)$, implicit differentiation solves $(I - J_*)^\top q = v$, followed by $g = C_*^\top q$, assuming invertibility (Bai et al., 2019). Our finite-chain model uses a constant learned conditioning embedding and receives the corrupted observation through initialization.

B Experimental Setting

B.1 Training Setting

Table 3: Model configurations and parameter counts in millions. EBT-S/B/L denote our small, base, and large models, with depth, width, and head counts matching the DiT reference (Peebles & Xie, 2023). The EBT column counts trainable parameters in our checkpoints, which use a feed-forward expansion ratio of one. The reference column gives the original DiT/2 counts, with expansion ratio four.

Model	Blocks	Width	Heads	EBT (M)	DiT/2 reference (M)
EBT-S	12	384	6	21.9	33
EBT-B	12	768	12	86.8	130
EBT-L	24	1024	16	305.2	458

The fixed-seed comparison has 69 runs across 23 configurations. It shares 37 runs with the 72-run monitored cohort, all from the original 51. The matched-estimator reruns postdate the cohort lock, so no prediction analysis depends on them. Table 5 lists the seeds and monitoring coverage. Means include every run observed to at least 90% of its budget or ended in an established failure, at its last available diagnostic. One run, EBT-L ARTBP seed 2, was interrupted nonfailed by a cluster time limit near 8,800 of 25,000 steps and is listed but excluded from means and failure counts.

Table 4: Deterministic training settings shared across model sizes, except where specified.

Setting	Value
Data and model	ImageNet 64×64, values in $[-1, 1]$; bidirectional image EBT; patch 16; FFN multiplier 1.
Optimizer	AdamW, $\beta = (0.9, 0.999)$, weight decay 0.01; peak LR 3×10^{-4}; 1k warmup; cosine decay to one tenth.
Batch and budget	Batch 128 and 50k trainer steps. EBT-L microbatch 32, accumulation setting 4, 25k trainer steps.
Numerics	FP32 tensors, TF32 matrix multiplication; norm clipping 1.0, or 0.5 for endpoint-normalized full BPTT.
Inference	Nominal $K = 10$; channel RMS, $\epsilon = 10^{-6}$; last output unnormalized unless specified.
Step sizes	Learned rows (TBPTT-3, ARTBP, full BPTT learned): initialized at 300, LR multiplier 2000, floor 10^{-4}. Fixed rows: TBPTT-1 500, TBPTT-5 100, full BPTT 300.

Table 5: Fixed-seed outcomes, giving the last training-diagnostic PSNR in seed order. "Norm." applies normalization at the final output and "free" omits it. "Orig." marks the original ARTBP trainer with unequal scheduler and accumulation behavior. Coverage counts monitored traces. Traces of matched-estimator reruns postdate the cohort lock and enter no prediction analysis. [i]Run interrupted with its collapse established, budget in the text. [c]Run interrupted nonfailed at about one third of budget, excluded from means.

Size	Recipe	Seeds	PSNR (dB)	Coverage
EBT-S	TBPTT-3	1,2,33	9.0, 8.8, 7.0	3/3
	ARTBP	33,1,2	16.5, 16.5, 16.3	3/3
	ARTBP, orig.	1,2,33	16.1, 15.8, 16.0	3/3
	TBPTT-1	1,2,33	8.4, 8.5, 8.5	3/3
	TBPTT-5	1,2,33	9.7, 9.0, 6.8	3/3
	Full BPTT, learned	33,1,2	16.6, 11.8[i], 11.8	3/3
	Full BPTT, norm.	1,2,33	11.8, 11.7, 11.6	3/3
	Full BPTT, free	1,2,33	6.6, 5.8, 6.1	3/3
EBT-B	TBPTT-3	33,1,2	15.8, 15.4, 15.8	0/3
	ARTBP	33,1,2	16.0, 6.6[i], 6.8	0/3
	ARTBP, orig.	33,1,2	16.0, 7.3, 7.3	0/3
	TBPTT-1	33,1,2	14.6, 15.2, 14.4	0/3
	TBPTT-5	33,5,6	6.7, 6.2, 7.0	2/3
	Full BPTT, learned	33,1,2	5.8, 6.7[i], 6.0[i]	0/3
	Full BPTT, norm.	33,1,2	11.7, 6.5, 6.5	0/3
	Full BPTT, free	33,1,2	6.2, 6.2, 6.0	3/3
EBT-L	TBPTT-3	33,1,2	15.6, 16.3, 15.8	3/3
	ARTBP	33,1,2	6.9[i], 6.5[i], 11.5[c]	3/3
	ARTBP, orig.	33,1,2	6.9, 6.5, 6.4	2/3
	TBPTT-5	33,1,2	6.9, 6.8, 6.9	3/3
	Full BPTT, learned	33,1,2	5.6, 6.1[i], 6.0[i]	3/3
	Full BPTT, norm.	33,1,2	6.3, 6.2, 6.2	3/3
	Full BPTT, free	33,1,2	6.2, 5.5, 6.2	3/3

Thirteen checkpoint optimizer states reveal unequal update schedules in the original trainer. At 50k trainer steps, inspected EBT-S/B TBPTT-3 checkpoints have 50k Adam updates and 50,001 scheduler steps, while original-trainer ARTBP has 50k updates but 100,001 scheduler steps. For EBT-L, three TBPTT-3 checkpoints have 25k updates at trainer step 25k, while original-trainer ARTBP seed 33 has 99,994 updates and 124,995 scheduler steps there. Its manual-gradient branch bypasses accumulation and also advances the scheduler.

The ARTBP and learned-step-size full BPTT rows of Table 1 use a corrected trainer whose manual-gradient path matches the automatic path in optimizer stepping, scheduler advancement, and accumulation. These reruns share every setting with TBPTT-3 at the same size and seed except the estimator and its per-output loss weighting. The original-trainer ARTBP outcomes remain in Table 5 and show the same ranking pattern across sizes. The interrupted runs are as follows. EBT-B full BPTT seed 2 ends at a NaN guard near 7,100 of 50,000 steps. EBT-L ARTBP seeds 33 and 1 end at NaN guards

Table 6: **EBT-B validation quality.** Each row evaluates one checkpoint on eight batches of four sampled ImageNet validation images. Higher PSNR and SSIM and lower LPIPS are better. FD denotes finite-difference derivatives. ARTBP lengths are nominal segment lengths. Stochastic rows use the combined recipe in Table 13. These checkpoint evaluations are separate from the replicated training-outcome comparison.

Checkpoint	PSNR (dB) ↑	SSIM ↑	LPIPS ↓
Full BPTT, endpoint norm.	11.63	0.272	0.430
TBPTT-1	14.23	0.213	0.545
TBPTT-3, FD	14.74	0.184	0.692
ARTBP, length 2	15.00	0.228	0.460
ARTBP, length 3	15.90	0.268	0.499
Full BPTT, endpoint norm., stochastic	11.60	0.271	0.433
TBPTT-3, stochastic	13.71	0.231	0.456
TBPTT-5, stochastic	13.24	0.245	0.448
TBPTT-3, annealed noise	15.63	0.276	0.556
TBPTT-5, annealed noise	13.68	0.211	0.728
TBPTT-3, randomized length	15.04	0.234	0.534

Table 7: **Single-pass computation cost.** Measurements use EBT-B, batch size eight, and $K = 10$. Memory is peak allocated GPU memory. Time covers one forward and backward pass, excluding optimizer state and updates. Each ARTBP measurement uses one sampled truncation pattern.

Estimator	Peak memory (GiB)	Time (s)
TBPTT-1	0.677	0.237
TBPTT-3	1.076	0.337
TBPTT-5	1.342	0.444
ARTBP, length 3	1.406	0.620
ARTBP, length 2	1.821	0.661
Full BPTT	2.005	1.346

near 20,200 and 2,100 of 25,000. EBT-L full BPTT seed 1 ends at a logging failure near 5,400 of 25,000 after collapsing within its first 2,500 steps, and seed 2 ends at a NaN guard near 8,100. EBT-S full BPTT seed 1 ends nonfailed at a logging failure near 45,700 of 50,000. Every interrupted failure holds its collapse through its last observation.

Training samples one of 100 indices in an underlying 200-step diffusion schedule: $x_0 = \sqrt{\bar{a}_\tau} y + \sqrt{1 - \bar{a}_\tau}\epsilon$, $\epsilon \sim \mathcal{N}(0, I)$. The noise setting 0.5 specifies the fraction of the diffusion schedule used for corruption. The training diagnostic uses maximum configured corruption on the current training batch. It denormalizes and clips pixels to $[0, 255]$, then computes $10 \log_{10}(255^2/(\mathrm{MSE}_{\mathrm{batch}} + 10^{-12}))$. Validation clips to $[0, 1]$, computes PSNR for each image, and averages the results.

B.2 Validation Quality

Validation uses SSIM (Wang et al., 2004) and AlexNet LPIPS (Zhang et al., 2018). Endpoint-normalized full BPTT has lower PSNR than ARTBP but higher SSIM and lower LPIPS. Each row evaluates one checkpoint, and the ranking depends on the image-quality metric.

B.3 Memory and Computation

For EBT-B, batch eight, and $K = 10$, the single-pass benchmark reports peak allocated memory / time of 0.68 GiB / 0.24 s for TBPTT-1, 1.08 / 0.34 for TBPTT-3, 1.34 / 0.44 for TBPTT-5, 1.41 / 0.62 for one ARTBP draw, and 2.01 / 1.35 for full BPTT. The harness clears gradients, resets the CUDA peak counter, and synchronizes after backward. The measurement covers one forward and backward pass, with optimizer state and updates excluded.

Table 8: Monitored cohort. NF means no observed failure at the last evaluation. Timing analyses require a peak above 9 dB, separating deterioration from persistently low quality.

Size	Runs	NF	Failed	Peak $>$ 9 dB
EBT-S	24	19	5	19
EBT-B	34	19	15	34
EBT-L	14	3	11	13
Total	72	41	31	66

C Monitoring and Prediction

C.1 Monitored Runs

The monitored cohort contains 33 configurations, including stochastic and inner-preconditioning studies. The numerical failure rule is final PSNR below 8 dB or more than 3 dB below the run's peak. Four NF and two failed runs never exceed 9 dB. The remaining 66 records contain 37 NF runs and 29 timed collapses, and 26 collapse before 5k trainer steps. The three later collapses are TBPTT-5 runs, EBT-B seed 5 at 18,299 and EBT-L seeds 33 and 2 at 7,799. A run contributes only windows covered by its observed follow-up.

C.2 Score Definitions and Aggregation

The live monitor performs Hessian-vector products and ten Lanczos iterations at the endpoint of a deterministic chain on the current training batch. It uses conditioning index zero. Current code floors effective step sizes at 10^{-4} and curvature at 10^{-12}, then multiplies $(1 + \alpha_k \lambda^{+}_{\mathrm{end}})$ over the final $\min(M, n_\alpha)$ entries of a length-n_α schedule. A parametric schedule is materialized at nominal K, and a length-one stored schedule contributes one factor. Full BPTT uses nominal K for M, and ARTBP uses its rounded nominal segment length, at least one. Preconditioned runs use the spectrum of $D^{1/2} H D^{1/2}$. This score summarizes endpoint curvature and the stored step-size schedule.

We use the recorded live series throughout the main analysis. Some recorded scores used raw step sizes before clamping. We exclude negative, subunit, and NaN scores. Positive infinity is retained as an alarm and capped at $\log_{10} \rho = 38$ for ranking.

The retrospective controller score is a different quantity, $s_j = (1 + \widetilde{\alpha}_j \Lambda_j)^M$, where Λ_j aggregates curvature and $\widetilde{\alpha}_j$ is internal controller state. In monitor-only mode, that state still changes without changing the model's step sizes. The retrospective aggregation uses this series when available and converts overflow to 10^{38}. The controller score describes its internal state. Repeated cached entries retain the last available value. Its completed-run maxima reproduce the NF ceiling 3.769×10^{13} and failure floor 4.482×10^{27} of the completed-run analysis, often using observations after deterioration.

C.3 Out-of-Sample Check

Four matched-estimator EBT-S reruns carry monitor traces recorded after the cohort was locked. They enter no prediction analysis. Reading each at step 2,000 with the locked protocol gives 2.2×10^6 and 5.4×10^7 for the two ARTBP runs. Both read below the threshold and both survive. The two full BPTT runs read 3.9×10^{11} and 4.3×10^{24} and also survive. The larger reading stays above 10^{17} for the entire run, and its completed-run maximum of 4.29×10^{24} exceeds the locked cohort's nonfailed ceiling of 3.77×10^{13}. The retrospective separation in Section 4.4 is therefore specific to the locked cohort, while the low-score rule transfers to these runs.

C.4 Collapse Times and Eligible Windows

For a failed run with peak $P_{\max} > 9$, let T_{col} be the earliest recorded time after which every remaining PSNR is below $\max(8, P_{\max} - 3)$. This is an outcome label using the full record. Eligibility at time t uses only past information. The observed peak must exceed 9 dB, the latest PSNR must be at least $\max(8, P_{\max,\le t} - 3)$, and the run must not already have collapsed. A window $(t, t + h]$ is violated when $t < T_{\mathrm{col}} \le t + h$, and a nonviolated window requires observed follow-up through

Table 9: Live-score horizon uncertainty. Bootstrap intervals resample complete runs 2,000 times with seed zero, holding the selected horizon fixed. Event runs contribute at least one violated window. Equal-run survival first averages windows within each run.

Size	Horizon	Event runs/runs	95% interval (%)	Equal-run (%)
EBT-S	10,000	4/19	$[97.844, 99.934]$	78.95
EBT-B	500	15/34	$[98.651, 99.668]$	79.49
EBT-L	100	2/13	$[99.286, 100]$	99.64

Table 10: Retrospective-controller comparison only. Horizons are selected on the retrospective aggregation, separately from the live-score analysis in Section 4.4. All violations at each selected nonzero-event horizon come from one run.

Size	Controller H_{99}	Violations/windows	Runs	Held-out survival (%)
EBT-S	40,000	0/1,113	13	100.00
EBT-B	3,200	64/6,557	19	97.58
EBT-L	250	5/662	4	91.54

$t + h$. Repeated logged evaluations limit the resolution of this event time. Horizon windows start at native valid-score timestamps. PSNR is the latest entry at or before that timestamp. Cached values retain their recorded timestamps, so consecutive windows can overlap and reuse an evaluation.

C.5 Horizon Sensitivity

The main horizon analysis uses eligible live readings below 10^9 and the grid $\{100, 250, 500, 1000, 2000, 3200, 5000, 10000, 20000, 40000\}$. For each model size, it selects the largest horizon with at least 30 evaluable windows and at least 99% observed survival. The selected values are 10k, 500, and 100 trainer steps. Leaving each run out of horizon selection and pooling its eligible test windows gives survival of 99.174%, 99.322%, and 99.493%, respectively. Equal run weighting changes all three selected horizons to 100 steps. The main estimate weights every eligible logged window equally, while the sensitivity analysis gives each run equal weight.

The intervals in Table 9 preserve within-run dependence but condition on horizon selection. A run-held-out fold selects its horizon using only the remaining same-size runs, and test windows are then pooled across folds. The fold-matched no-score control uses the same eligible timestamps without the $\rho < 10^9$ filter.

The retrospective-controller horizons in Table 10 answer the same window question for a different series. Removing TBPTT-5 eliminates their violations and moves the selected horizons to observable limits of 40k, 40k, and 10k. The live and controller scores therefore give different horizon estimates.

C.6 Forecast Protocol and Warning Lead Times

At trainer steps 500, 1k, 2k, and 5k, the forecast replay takes the latest valid live score no more than 100 steps old and predicts collapse within the next 5k steps. Eligibility and censoring follow the previous subsection. Each run contributes once at each landmark, and comparisons use the same eligible rows. The four event/run counts are 24/54, 21/55, 8/41, and 2/35. Corresponding live-score AUCs are 0.653, 0.728, 0.689, and 0.970, and nominal-depth AUCs are 0.730, 0.732, 0.786, and 0.773. The last landmark has only two events from one configuration. Uncertainty resamples size–recipe groups while retaining all seeds within each group.

The incremental test at 1k holds out each of 28 complete size–recipe groups. Both logistic models use three fixed size indicators and backward depth, and the extended model adds $\log_{10} \rho$, capped at 38. Numeric features are standardized using training folds only. We fix ℓ_2 regularization at $C = 1$ and use L-BFGS with at most 5,000 iterations. Class weights are equal, and hyperparameters are fixed. Out-of-fold AUC is 0.598 for size/depth and 0.686 with the score. Paired group resampling of the saved predictions gives AUC difference of 0.088 with 95% interval $[0.002, 0.186]$. The Brier difference

Table 11: **Failure prediction on a common evaluation set.** All rows use the same 55 runs observed at step 1,000, of which 21 collapse within the next 5,000 steps. Upper rows rank runs directly by the named signal. Intervals resample complete size–configuration groups. Both directions of gradient norm are reported. Lower rows evaluate logistic regressions with complete size–configuration groups held out. Higher AUC and lower Brier error are better.

Predictor	AUC	95% interval	Brier error
Live curvature score ρ	0.728	$[0.558, 0.876]$	—
Backward depth	0.732	$[0.528, 0.904]$	—
Endpoint curvature	0.599	$[0.394, 0.812]$	—
Lower current PSNR	0.571	$[0.417, 0.718]$	—
PSNR drop from past maximum	0.528	$[0.391, 0.649]$	—
Higher gradient norm	0.361	$[0.182, 0.566]$	—
Lower gradient norm	0.639	$[0.434, 0.818]$	—
Model size and backward depth	0.598	—	0.247
Model size, backward depth, and $\log_{10} \rho$	0.686	—	0.239

is -0.008, with interval $[-0.044, 0.032]$. The 2,000 bootstrap draws reuse the fitted predictions. The Brier-score difference interval includes zero.

The pre-specified threshold 10^{16} alarms before 18 of 29 collapses and on four of 37 NF runs. Only five collapses have at least 250 steps of lead, four have 500, three have 1k, and none have 2k. Maximum recorded lead is 1,750 steps. These leads refer to the logged collapse time, and cached PSNR values can lag actual deterioration. Repeating the alarm test with the retrospective controller series yields no advance alarm, with only three failures having an eligible controller reading.

D Mechanism Probes and Intervention Details

D.1 Learned Step-Size Trajectories

The representative ARTBP EBT-B trajectory has seven entries at the 10^{-4} floor by 5k steps and eight by 10k. The eight remain there through 50k, while the first and last reach approximately 1431 and 523. In the corresponding TBPTT-3 trajectory, the seven entries outside the differentiated window retain their initialization. Below the hard clamp, the local derivative with respect to a raw step parameter is zero. Optimizer momentum or weight decay can still move that parameter, and persistence here is established by saved checkpoints. Normalization remains active at updates with small step sizes.

D.2 Gradient Comparisons

The fixed-objective probe underlying Figure 2 uses two checkpoints of one EBT-B run trained with randomized inference length, eight batches of eight validation draws, and the unchanged last-three-output loss. The deterministic probe has 12 updates. One-hot backward weights isolate each local parameter contribution, and their sum reconstructs the full gradient with relative error below 8.3×10^{-8}. The vectors include transformer parameters, the learned conditioning embedding, and four shared parametric step-size coefficients. Frozen parameters contribute zero blocks.

For contribution matrix U and full gradient g, the best same-batch scalar reweighting satisfies

$$\min_{w} \|Uw - g\|_2^2 = \|g\|_2^2 - c^\top Q^\dagger c, \qquad Q = U^\top U, \quad c = U^\top g. \tag{13}$$

This is the orthogonal-projection identity. Saved per-batch Gram matrices supply all terms. Oracle three-contribution relative errors are 0.104 at 5k and 0.638 at 50k, and ordinary TBPTT-3/full cosines change from 0.971 to -0.693. The Gram matrices combine energy and step-size parameters in one Euclidean gradient.

Table 12: **Reconstruction with longer inference.** Entries are validation PSNR in dB at the maximum training corruption setting, using eight batches of two sampled ImageNet validation images. All checkpoints use EBT-B and TBPTT-3 after 50k training steps. K is nominal evaluation depth. Fixed-length training executes K evaluation updates, and randomized-length recipes execute $K + 2$. Each row uses one training seed. Both training length and step-size parameterization vary between recipes.

Training length	Step sizes	Mean train length	$K = 10$	$K = 20$	$K = 100$
Fixed, 10	Learned	10	15.40	10.29	7.97
Randomized, 10–30	Fixed	20	14.04	14.39	13.71
Randomized, 10–30	Parametric	20	14.38	15.22	13.55

For excluded batch b, fit weights on the other seven batches by minimizing their mean relative squared error:

$$w_{-b} = \left(\sum_{i \neq b} Q_i / n_i \right)^{\dagger} \left(\sum_{i \neq b} c_i / n_i \right), \qquad n_i = \|g_i\|_2^2. \tag{14}$$

We use float64 arithmetic and a fixed relative singular-value cutoff of 10^{-10}. All three-contribution Gram matrices have full rank. Mean excluded-batch errors are 0.110 and 0.662, and mean coefficient ℓ_1 norms are 52.31 and 8.34. The eight batches are paired across checkpoints of the same trained model.

D.3 Randomized-Length Training and Evaluation

Two EBT-B variants repeat each of ten nominal update indices independently one, two, or three times, giving training lengths 10–30 with mean 20. One freezes $\alpha = 300$, and the other uses a positive exponential polynomial in normalized progress with a separate final-step multiplier. At nominal evaluation depth 100, the implementation retains two final-index repetitions, so these variants execute 102 updates. Existing single-checkpoint validation sweeps retain 13.71 and 13.55 dB there, versus 7.97 dB for a fixed-length checkpoint at 100 updates. Reconstruction quality peaks before the maximum evaluated depth. This experiment increases the number of inference updates at evaluation. The prediction horizon measures future training steps.

D.4 Other Interventions

Full S multiplies step sizes by $U[1/2, 2]$, adds pre-gradient Gaussian perturbations with standard deviation $0.05 \cdot 0.85^k$, and repeats the final nominal update 1–3 times during training. Deterministic evaluation uses 12 updates, while ARTBP-S and noise-only variants use ten. For EBT-B, TBPTT-5 full S has two NF attempts out of three, whereas seven deterministic attempts across step sizes 30, 100, and 300 fail. One noise-only record reaches 14.2 dB through 30k steps. Full-S TBPTT-3 finishes near 14.1 dB across three runs, while the noise-only record reaches about 15.6 dB. The treatments differ in both their components and observed training duration.

Curvature feedback, gradient gating, and inference preconditioning produced no consistent improvement in the recorded attempts. One gated 2k-step record and a plain-descent preconditioning control reaching about 14.9 dB through 10k are censored observations. More than 30 implicit-gradient configurations yield no successfully trained model under their tested inner-solver and linear-solve budgets. The attempted solvers operated on the finite-chain model, where the endpoints were still changing and normalization altered the fixed-point equation.

Table 13: **Combined stochastic updates with EBT-B.** All runs reach 50k trainer steps. PSNR columns give final training PSNR in dB. Means include failed runs. ARTBP uses the original manual-gradient trainer. Final normalization is applied at the last output.

Recipe	Seed 33	Seed 1	Seed 2	Mean	Failed
TBPTT-1	10.94	11.65	12.03	11.54	0/3
TBPTT-3	14.10	14.05	14.05	14.07	0/3
TBPTT-5	13.40	13.80	6.24	11.15	1/3
ARTBP	6.75	13.77	15.07	11.86	1/3
Full BPTT, final norm.	11.82	11.84	11.83	11.83	0/3

Table 14: **Additional EBT-B training studies.** Counts summarize recorded attempts across their available observation budgets. They overlap with the fixed-seed and monitored cohorts. Nonfailed means that the run does not meet the failure rule at its last observation. ARTBP rows use the original manual-gradient trainer.

Study	Attempts	Nonfailed	Observation
ARTBP	9	3	Additional deterministic seeds.
TBPTT-3	4	4	Final PSNR 15.4–16.0 dB.
TBPTT-5	7	0	Multiple fixed step sizes.
Inner-update screen	6	1	Plain-descent control has 10k steps of follow-up.
ARTBP preconditioning	2	0	Both runs end near 6.97 dB after 25k steps.

Table 15: **Individual EBT-B controls and interventions.** Entries give final training PSNR at the stated observation budget. Each row is one run. The learned-step-size full-BPTT run reaches 16.30 dB before collapsing around 26k steps. Different budgets and recipe settings prevent pooling these rows into one estimator ranking. ARTBP and gradient-gating rows use the original manual-gradient trainer.

Configuration	Training steps	PSNR (dB)
Full BPTT, learned step sizes	50,000	5.82
TBPTT-3, fixed $\alpha = 100$	25,000	14.25
ARTBP, fixed $\alpha = 100$	25,000	6.42
TBPTT-5, fixed $\alpha = 300$	25,000	13.02
TBPTT-3, randomized step sizes	50,000	14.67
TBPTT-3, annealed noise	50,000	15.60
TBPTT-5, annealed noise	30,000	14.22
TBPTT-3, randomized length, learned steps	50,000	15.11
TBPTT-3, randomized length, fixed steps	50,000	14.25
Gradient gate, batch 128, initial $\alpha = 300$	6,000	5.82
Full normalization reference, batch 128, initial $\alpha = 300$	6,000	11.09

Table 16: **Implicit-differentiation experiments.** The table groups 31 runs with solver diagnostics. The class-conditional studies are separate from the denoising comparison. All runs use the manual-gradient trainer, so their training outcomes inherit its schedule and accumulation differences. The tested configurations yield no successfully trained model.

Study	Runs	Observed outcome
Class-conditional damping, prior, and budget sweep	11	Last solver readings report nonconvergence. Final validation loss is 0.601–0.627, against an initial value of 0.639.
Class-conditional solver sweep	9	Anderson and Broyden variants end with validation loss 0.606–0.642 and report nonconvergence at the last read.
Class-conditional convergence sweep	3	Longer inner budgets and changes to step size and damping do not produce successful training.
Denoising implicit-gradient baseline	2	Reconstruction does not improve beyond the initial PSNR of about 7.58 dB.
Denoising derivative-backend sweep	6	Changing the inner and outer derivative backends does not produce successful training. Completed runs end near 5.2–6.0 dB.