AUXILIARY SUPERVISION SHAPES OPTIMIZATION GEOMETRY FOR RECURRENT SEQUENCE MODELS

ABSTRACT

Recurrent sequence-to-vector models are widely used for tasks such as time series forecasting and sequence classification or regression, where supervision is often applied only after the complete input sequence has been processed. This can create a temporal parameter-information bottleneck, leaving important parameter directions weakly informed and thereby causing poor conditioning, potentially slow and numerically sensitive training, and undesirable rollout behavior on unseen sequences. We show that auxiliary supervision can alleviate this bottleneck by exposing parameter information at intermediate positions before it becomes invisible or is attenuated by the recurrent dynamics. Our sensitivity-based analysis establishes that suitably chosen auxiliary observations can improve local parameter observability and prevent deterioration of generalized Gauss–Newton conditioning with increasing sequence length, thereby supporting faster and more numerically robust training. At the same time, auxiliary supervision acts as an implicit regularizer that can promote desired intermediate or rollout behavior. Moreover, we characterize the resulting primary-task performance trade-off and show that compatible auxiliary objectives need not sacrifice globally optimal primary-task performance. Analytical results and extensive experiments with modern Mamba models on the standard ListOps sequence-classification benchmark corroborate the identified mechanisms and demonstrate the practical benefits of appropriately chosen auxiliary supervision for training efficiency and primary-task prediction accuracy.

1 INTRODUCTION

Recurrent sequence-to-vector models are widely used for time-series forecasting, sequence classification and regression, system identification, and visual or acoustic sequence processing (Hewamalage et al., 2021; Salinas et al., 2020; Ismail Fawaz et al., 2019; Mohammadi Foumani et al., 2024; Forgione & Piga, 2021; Beintema et al., 2023; Zhu et al., 2024; Li et al., 2024; Cakir et al., 2017; Xu et al., 2018). Relevant architectures range from classical RNNs, LSTMs, and GRUs to modern state-space sequence models such as selective SSMs and Mamba-like architectures (Hochreiter & Schmidhuber, 1997; Cho et al., 2014; Gu & Dao, 2024; Lahoti et al., 2026). In many such tasks, supervision is applied only after the complete input sequence has been processed; we refer to this setting as *terminal supervision*.

While natural from a task perspective, terminal supervision can provide limited information about how model parameters affect the sequence internally. Parameter effects arising at intermediate positions must propagate through the subsequent recurrent dynamics before influencing the training loss. This can create a temporal parameter-information bottleneck: relevant directions may be invisible at the terminal output or strongly attenuated before reaching it, an effect closely related to the classical difficulty of learning over long temporal horizons in recurrent networks (Bengio et al., 1994; Pascanu et al., 2013; Trinh et al., 2018; Kag & Saligrama, 2021; Schiller et al., 2026; Zucchet & Orvieto, 2024). From an optimization perspective, weak parameter information can induce strongly anisotropic curvature and poor conditioning (Zhao et al., 2024), potentially resulting in slow and numerically sensitive training. At the same time, supervising only the final task output leaves intermediate and rollout behavior comparatively unconstrained, which can contribute to undesirable effects on sequences that differ from those encountered during training.

Auxiliary supervision at intermediate rollout positions provides a natural way to enrich this training signal and can additionally promote desired intermediate representations or rollout behavior

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.

(Forgione & Piga, 2021; Masti & Bemporad, 2021). Related approaches include deep supervision through intermediate prediction heads (Lee et al., 2015; Szegedy et al., 2015; Püttschneider et al., 2025), auxiliary prediction tasks in recurrent networks (Trinh et al., 2018; Kag & Saligrama, 2021), knowledge distillation (Hinton et al., 2015; Romero et al., 2015; Furlanello et al., 2018; Aguilar et al., 2020), and trajectory- or process-level supervision (Goyal et al., 2016; Malach et al., 2026), with recent applications to time-series forecasting, representation supervision, and length generalization (Sun et al., 2025; Awasthi et al., 2025; Wang et al., 2026). Despite motivations ranging from facilitating optimization and long-horizon learning to shaping intermediate representations and promoting desired rollout behavior, these approaches share a common structural feature: training is provided with additional observations of the model before the terminal output is reached.

The effectiveness of auxiliary supervision depends on its design and weighting, motivating designs based on temporally correlated task scheduling (Wu et al., 2021) or gradient-based compatibility measures (Du et al., 2018). Moreover, the special case of intermediate sub-task supervision has been shown to improve the learnability of decomposable sequence-learning problems (Wies et al., 2023). Related literature in the context of system identification and experiment design characterizes local parameter identifiability through output sensitivities, information matrices, or observability Gramians and exploits informative observations to improve parameter estimation (Van Willigenburg et al., 2022; Himpe & Ohlberger, 2014; Pronzato, 2008). Complementary theoretical analyses have linked auxiliary supervision to improved learning behavior in specific settings: representation-level supervision can introduce curvature in directions weakly constrained by next-token prediction (Zhang et al., 2026), per-layer supervision can improve the Bregman conditioning of intermediate Transformer representations (Kerce, 2026), and more informative labels can increase expected Fisher information (Saluja et al., 2026). In contrast, we focus on recurrent sequence models more generally and, in particular, on the effect of auxiliary supervision on parameter-space geometry, explicitly accounting for how parameter information propagates—and may be lost—through the recurrent dynamics.

We develop a unified sensitivity-based framework for auxiliary supervision in recurrent sequence models. It reveals two complementary mechanisms: intermediate observations can provide parameter information before it is lost through recurrent propagation, thereby reshaping the local optimization geometry, while the auxiliary objective simultaneously acts as an implicit regularizer on intermediate or rollout behavior. Our main contributions are as follows.

- **Parameter information framework.** We formulate terminal and auxiliary supervision in terms of parameter sensitivities and generalized Gauss–Newton geometry, showing how auxiliary observations can improve local parameter observability by revealing directions invisible under terminal supervision.
- **Dynamical forgetting and conditioning.** We show that, under exponential stability of the recurrent Jacobian flow and without subsequent re-excitation, parameter information available at intermediate positions can decay exponentially before reaching terminal supervision, causing exponential deterioration of generalized Gauss–Newton conditioning with sequence length. Complementary auxiliary observations can prevent this deterioration and yield uniformly bounded conditioning.
- **Primary-task performance trade-off.** We bound the primary-performance gap induced by auxiliary supervision in terms of its compatibility with primary-optimal parameterizations and characterize its dependence on the auxiliary-loss weight. Compatible auxiliary objectives can therefore reshape the optimization geometry without sacrificing globally optimal primary-task performance.

We complement the analysis with experiments on modern Mamba models that investigate the predicted mechanisms and their practical consequences.

2 Auxiliary supervision in recurrent sequence models

2.1 Problem setup

We consider the general problem of learning a recurrent sequence-to-vector model, that is, a map from a d_x-dimensional input sequence $X \in \mathbb{R}^{d_x \times N}$ of length N to a d_y-dimensional output $Y \in \mathbb{R}^{d_y}$:

$$X \mapsto Y. \tag{1}$$

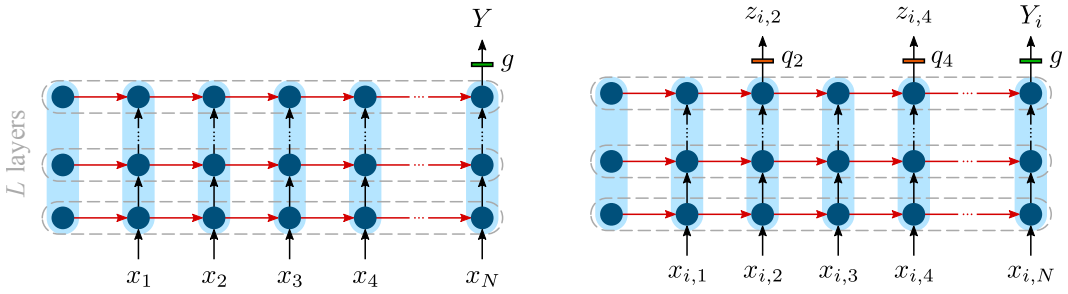


Figure 1: General architecture of a causal recurrent sequence-to-vector model, unfolded in depth and time. The model consists of L recurrent layers and a terminal readout map g. Black arrows indicate within-time, layer-wise processing, while red arrows indicate temporal recurrent propagation. The arrows may represent learnable linear or nonlinear projections, normalization layers, activations, selective gating mechanisms, and skip connections. Left: During inference, the model processes an input sequence $X = (x_1, \dots, x_N)$ and produces the terminal output Y. Right: During training with auxiliary supervision, auxiliary readout maps q_r generate causal observations $z_{i,r}$ at selected times $r \in \mathcal{R}$ along the rollout of training sample i.

Throughout, recurrent sequence models are understood broadly as causal state-space architectures that process the input sequence through a hidden-state recursion. We collect the learnable parameters of the inference model in $\theta \in \mathbb{R}^o$ and denote by $Y(X;\theta)$ the model output after processing the input sequence $X = (x_1, \dots, x_N)$. The terminal output is generated by a readout map g applied after the final recurrent block; see Figure 1 (left) for an illustration.

To determine the model parameters θ, we assume that there is a given dataset $D = \{(X_i, \tilde{Y}_i)\}_{i=1}^B$ consisting of B pairs of input sequences $X_i = (x_{i,1}, \dots, x_{i,N})$ of common length N and corresponding targets $\tilde{Y}_i$, $i \in [1, B]$. The primary training goal is to learn the sequence-to-vector mapping in (1) by matching the model output $Y(X_i;\theta)$ to the corresponding target $\tilde{Y}_i$. We therefore define the primary task loss under *terminal supervision* as

$$P(\theta) := \frac{1}{B} \sum_{i=1}^{B} l^{\mathrm{p}}\big(Y(X_i;\theta), \tilde{Y}_i\big). \tag{2}$$

Here, l^{p} denotes a task-specific loss evaluated after the complete input sequence has been processed, such as the mean-squared error for regression or the cross-entropy loss for classification.

In addition to terminal supervision, we allow for auxiliary supervision at selected temporal positions. Let $\emptyset \neq \mathcal{R} \subseteq [1, N-1]$ denote the set of auxiliary supervision times. For each $r \in \mathcal{R}$, let $z_r(X^r;\theta,\xi)$ denote the causal input–output map induced by the model rollout up to time r and the auxiliary readout q_r, where $\xi \in \mathbb{R}^{o_{\mathrm{a}}}$ collects optional auxiliary-only parameters (e.g., of an intermediate superclass-classification head (Alsallakh et al., 2018)) that may be discarded at inference. Its evaluation on training sample i is denoted by

$$z_{i,r} := z_r(X_i^r;\theta,\xi), \qquad X_i^r := (x_{i,1}, \dots, x_{i,r}), \qquad r \in \mathcal{R}, \quad i \in [1, B]. \tag{3}$$

The general model architecture during training with auxiliary supervision is illustrated in Figure 1 (right). The auxiliary observations $z_{i,r}$ may, for example, represent intermediate predictions, auxiliary task outputs, teacher-provided features, or supervised projections of intermediate representations.

Given corresponding auxiliary targets $\tilde{z}_{i,r}$ and loss functions l_r^{a}, we define

$$A(\theta,\xi) := \frac{1}{B|\mathcal{R}|} \sum_{i=1}^{B} \sum_{r \in \mathcal{R}} l_r^{\mathrm{a}}\big(z_r(X_i^r;\theta,\xi), \tilde{z}_{i,r}\big). \tag{4}$$

The resulting training objective is

$$J_\lambda(\theta,\xi) = P(\theta) + \lambda A(\theta,\xi), \tag{5}$$

where $\lambda \geq 0$ controls the strength of auxiliary supervision.

The problem setup and its analysis can be extended in several directions; see Appendix A.1 for details.

Remark 1 (Implicit regularization)**.** *Let $\lambda > 0$ and suppose that $(\theta^\star_\lambda, \xi^\star_\lambda)$ minimizes J_λ in (5). Defining $\varepsilon := A(\theta^\star_\lambda, \xi^\star_\lambda)$, the same pair satisfies $(\theta^\star_\lambda, \xi^\star_\lambda) \in \arg\min_{\theta,\xi} \{P(\theta) \mid A(\theta, \xi) \le \varepsilon\}$. Hence, auxiliary supervision can equivalently be interpreted as restricting the primary optimization problem to models satisfying an auxiliary-loss constraint, providing an implicit regularization mechanism whose meaning depends on the chosen auxiliary objective. Classical parameter regularization is recovered as a special case by choosing the auxiliary output directly as θ.*

We next analyze how the additional observation channels introduced by auxiliary supervision affect local parameter observability and the conditioning of the training problem.

2.2 Auxiliary supervision shapes optimization geometry

For the theoretical analysis, we make the recurrent state-space realization underlying the sequence-to-vector model explicit. Specifically, we consider

$$h_t = f(h_{t-1}, x_t; \theta), \quad t \in [1, N], \qquad h_0 = 0, \tag{6}$$

$$z_r(X^r; \theta, \xi) = q_r(h_r, x_r; \theta, \xi), \quad r \in \mathcal{R}, \tag{7}$$

$$Y(X; \theta) = g(h_N, x_N; \theta). \tag{8}$$

Here, $h_t \in \mathbb{R}^{d_h}$ denotes the recurrent state at temporal position t; for multilayer architectures, h_t may collect the recurrent states of all layers. The maps f and g characterize the recurrent dynamics and primary readout, respectively, while q_r, $r \in \mathcal{R}$, denotes the auxiliary readout introduced in Section 2. For notational simplicity, we suppress the explicit dependence on X, X^r, θ, and ξ whenever it is clear from context and write simply Y and z_r in the following. We assume that all auxiliary outputs have a common dimension d_z; allowing for a dimension depending on r requires only minor notational modifications.

2.2.1 Sensitivity-based optimization geometry

Throughout this section, we analyze the optimization geometry with respect to the inference parameters θ and treat auxiliary-only parameters ξ as fixed. If ξ is jointly optimized, the quantities considered below correspond to the θ-block of the respective joint parameter-space quantities. Moreover, for notational simplicity, we consider a single input–target sample ($B = 1$), comprising an input sequence X together with the corresponding primary target $\tilde{Y}$ and auxiliary targets $\tilde{z}_r$, $r \in \mathcal{R}$. We therefore omit the sample index i in the following; the extension to multiple samples follows by averaging the corresponding sample-wise contributions, compare Example 1 below.

Define the primary and auxiliary parameter sensitivities as

$$S^{\mathrm{p}}(\theta) := \frac{\partial Y}{\partial \theta}(\theta) \in \mathbb{R}^{d_y \times o} \qquad S^{\mathrm{a}}_r(\theta, \xi) := \frac{\partial z_r}{\partial \theta}(\theta, \xi) \in \mathbb{R}^{d_z \times o},\ r \in \mathcal{R} \tag{9}$$

and the Hessians of the primary and auxiliary loss functions as

$$H^{\mathrm{p}}(\theta) = \nabla^2_Y l^{\mathrm{p}}(Y, \tilde{Y}) \in \mathbb{R}^{d_y \times d_y}, \qquad H^{\mathrm{a}}_r(\theta, \xi) = \nabla^2_z l^{\mathrm{a}}_r(z_r, \tilde{z}_r) \in \mathbb{R}^{d_z \times d_z},\ r \in \mathcal{R}. \tag{10}$$

Moreover, we introduce

$$G_\lambda(\theta) := G^{\mathrm{p}}(\theta) + \lambda G^{\mathrm{a}}(\theta, \xi), \quad G^{\mathrm{p}}(\theta) := {S^{\mathrm{p}}}^\top H^{\mathrm{p}} S^{\mathrm{p}}, \quad G^{\mathrm{a}}(\theta, \xi) := \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} {S^{\mathrm{a}}_r}^\top H^{\mathrm{a}}_r S^{\mathrm{a}}_r. \tag{11}$$

The matrix G_λ is the generalized Gauss–Newton (GGN) matrix associated with the training objective; for squared-error losses, it reduces to the standard Gauss–Newton matrix. It provides a local measure of parameter sensitivity weighted by the curvature of the output losses and therefore admits a parameter-information interpretation. For likelihood-based losses, it is also closely related to Fisher information (Saluja et al., 2026).

Due to the recurrent dynamics (6), the sensitivities obey a recursive update law. To this end, define the sensitivities of the recurrent dynamics:

$$A_t = \frac{\partial f}{\partial h}(h_{t-1}, x_t; \theta), \qquad B_t = \frac{\partial f}{\partial \theta}(h_{t-1}, x_t; \theta), \qquad t \in [1, N], \tag{12}$$

the primary readout:

$$C^{\mathrm{p}} = \frac{\partial g}{\partial h}(h_N, x_N; \theta), \qquad D^{\mathrm{p}} = \frac{\partial g}{\partial \theta}(h_N, x_N; \theta), \tag{13}$$

and the auxiliary readout:

$$C_r^{\mathrm{a}} = \frac{\partial q_r}{\partial h}(h_r, x_r; \theta, \xi), \qquad D_r^{\mathrm{a}} = \frac{\partial q_r}{\partial \theta}(h_r, x_r; \theta, \xi), \qquad r \in \mathcal{R}. \tag{14}$$

Then, the primary and auxiliary sensitivities in (9) can be rewritten as

$$S^{\mathrm{p}} = C^{\mathrm{p}}\Gamma_N + D^{\mathrm{p}}, \qquad S_r^{\mathrm{a}} = C_r^{\mathrm{a}}\Gamma_r + D_r^{\mathrm{a}}, \quad r \in \mathcal{R}, \tag{15}$$

where the time-varying matrix $\Gamma_t \in \mathbb{R}^{d_h \times o}$ is defined as

$$\Gamma_t := \sum_{k=1}^{t} \phi(t,k) B_k, \quad \phi(t,k) := A_t A_{t-1} \dots A_{k+1}, \quad \phi(t,t) := I \tag{16}$$

for all $t \in [1, N]$ and $\Gamma_0 := 0$.

The representation in (15) separates direct parameter effects through D^{p} and D_r^{a} from parameter effects propagated through the recurrent dynamics via Γ_t. This reveals two distinct mechanisms by which terminal supervision may lose parameter information. First, parameter directions visible through intermediate readouts may be unobservable from the terminal readout. Second, parameter sensitivity accumulated earlier in the sequence may be attenuated by the recurrent dynamics before reaching the terminal output. The following subsections analyze these two mechanisms. Throughout, we impose a mild regularity condition to ensure that the required derivatives are well-defined.

Assumption 1. *The functions f, g, and q_r, $r \in \mathcal{R}$, as well as the loss functions l^{p} and l_r^{a}, $r \in \mathcal{R}$ are sufficiently smooth for all derivatives used throughout this work to exist.*

2.2.2 Parameter observability

The direct auxiliary parameter sensitivities D_r^{a} in (14) available at intermediate observations do not enter the terminal sensitivity S^{p}; see (15). Hence, even without dynamical attenuation, a parameter direction may be locally invisible at the terminal output while being visible through an auxiliary readout. This effect is particularly relevant when auxiliary and primary readouts share parameters. To formalize this observation, let $\mathcal{R} = \{r_1, \dots, r_{|\mathcal{R}|}\}$ and define the stacked sensitivity matrix

$$\mathcal{S}(\theta) := \begin{bmatrix} S^{\mathrm{p}}(\theta)^\top & S_{r_1}^{\mathrm{a}}(\theta)^\top & \cdots & S_{r_{|\mathcal{R}|}}^{\mathrm{a}}(\theta)^\top \end{bmatrix}^\top. \tag{17}$$

The kernel of $\mathcal{S}$ characterizes the locally unobservable parameter subspace associated with the combined primary and auxiliary observations. This provides a direct basis for comparing parameter observability under terminal-only and auxiliary supervision.

Proposition 1 (Parameter observability under auxiliary supervision)**.** *Let Assumption 1 hold. Then,*

$$\ker(\mathcal{S}) = \ker(S^{\mathrm{p}}) \cap \bigcap_{r \in \mathcal{R}} \ker(S_r^{\mathrm{a}}) \subseteq \ker(S^{\mathrm{p}}). \tag{18}$$

The claim follows directly from the definition of $\mathcal{S}$ in (17). Hence, auxiliary supervision can reduce, but never enlarge, the locally unobservable parameter subspace. The inclusion in (18) is strict whenever at least one auxiliary sensitivity reveals a parameter direction contained in $\ker(S^{\mathrm{p}})$.

This effect is particularly relevant when intermediate and terminal readouts share parameters, including the case $q_r \equiv g$. Since the readouts are evaluated at different hidden states and inputs, a parameter direction v may have a nonzero direct contribution $D_r^{\mathrm{a}} v$ at an intermediate time r, while remaining invisible through $D^{\mathrm{p}} v$ at the terminal output. Such behavior can arise, for example, with shared nonlinear or attention-based readouts, tied output embeddings, and input-dependent selective readouts as used in SSMs such as Mamba (Gu & Dao, 2024; Lahoti et al., 2026). A particular example based on a selective SSM readout where this effect occurs is provided in Appendix A.2. It is deliberately kept simple to permit an explicit analytical characterization; however, the same mechanism can arise more generally whenever the available terminal observations do not sufficiently excite all relevant parameter directions.

Hence, auxiliary supervision can reveal parameter directions that are invisible at the terminal readout. The following subsection considers the complementary mechanism, in which parameter information is initially present but subsequently attenuated by the recurrent dynamics.

2.2.3 DYNAMICAL FORGETTING AND CONDITIONING

We next analyze dynamical forgetting. A parameter direction that is informative at some intermediate time $K < N$ may become weakly visible at the terminal output if its effect is subsequently attenuated by the recurrent dynamics and not re-excited later in the sequence. In particular, if $B_t v^\star = 0$ for all $K < t \leq N$ for some $v^\star \neq 0$, the definition of Γ_t in (16) yields

$$\Gamma_t v^\star = \phi(t, K)\Gamma_K v^\star, \qquad t \in [K, N]. \tag{19}$$

In the absence of subsequent re-excitation, decay of the state-transition sensitivity $\phi(t, K)$ directly induces a loss of parameter sensitivity toward the terminal output. We emphasize that such decay typically manifests as part of the classical vanishing gradient phenomenon in recurrent neural network training and is therefore highly relevant in practice (Bengio et al., 1994; Pascanu et al., 2013).

Theorem 1 (Dynamical forgetting of parameter information)**.** *Let Assumption 1 hold and suppose that the Hessians defined in (10), evaluated at the considered parameterization, are positive semidefinite. Moreover, suppose that there exist a fixed $K \in \mathbb{N}$, a unit parameter direction $v^\star \in \mathbb{R}^o$, and constants $C > 0$, $\rho \in (0, 1)$, $L_C > 0$, and $L_H > 0$, all independent of N, such that, for all $N \geq K$,*

$$\|\phi(t, K)\| \leq C\rho^{t-K}, \qquad t \in [K, N], \tag{20}$$

$$B_t v^\star = 0, \qquad t \in [K+1, N], \tag{21}$$

$$D^{\mathrm{p}} v^\star = 0, \tag{22}$$

$$\|C^{\mathrm{p}}\| \leq L_C, \qquad \|H^{\mathrm{p}}\| \leq L_H. \tag{23}$$

Then there exists $c > 0$, independent of N, such that

$$v^{\star\top} G^{\mathrm{p}} v^\star \leq c\rho^{2(N-K)}. \tag{24}$$

Thus, the parameter information available to terminal supervision in direction $v^\star$ decays exponentially with the distance $N - K$.

If, additionally, $K \in \mathcal{R}$ and there exists $\mu > 0$ such that

$$v^{\star\top} S_K^{\mathrm{a}\top} H_K^{\mathrm{a}} S_K^{\mathrm{a}} v^\star \geq \mu, \tag{25}$$

then

$$v^{\star\top} G_\lambda v^\star \geq \frac{\lambda\mu}{|\mathcal{R}|}. \tag{26}$$

Hence, if $|\mathcal{R}|$ is uniformly bounded, auxiliary supervision preserves a non-vanishing amount of parameter information in direction $v^\star$.

[Proof in Appendix B.1.] Theorem 1 separates the two mechanisms leading to information loss. Specifically, condition (20) formalizes forgetting through the recurrent dynamics (compare the discussion below (19)), whereas (21) excludes subsequent re-excitation of the considered parameter direction. Terminal supervision therefore observes the information available at time K only after its propagation to time N, whereas auxiliary supervision can access it before it is attenuated. We note that condition (21) can be relaxed to sufficiently decaying subsequent excitation.

The time K characterizes the information-loss mechanism and need not be known a priori in practice. Auxiliary supervision at multiple or task-informed temporal positions can instead provide observation channels before such information is lost. If $|\mathcal{R}|$ increases with N, the aggregate contribution of multiple auxiliary observations can analogously provide non-vanishing parameter information.

The positive-semidefiniteness condition on the loss Hessians and the uniform Hessian bound required above are satisfied for common choices such as the MSE and standard binary or multiclass cross-entropy losses formulated in terms of logits.

The information loss characterized by Theorem 1 directly affects the conditioning of the training objective. In particular, if G coincides with or suitably approximates the Hessian, a large spectral condition number

$$\kappa(G) := \lambda_{\max}(G)/\lambda_{\min}(G), \quad G \succ 0, \tag{27}$$

indicates strongly anisotropic curvature and can lead to slow convergence of gradient-based optimization algorithms; see Appendix A.3 for details. The following corollary shows that the conditioning under terminal supervision can deteriorate exponentially with increasing sequence length.

Corollary 1 (Ill-conditioning with increasing sequence length)**.** *Suppose that the conditions of Theorem 1 hold and that $G^{\mathrm{p}} \succ 0$ for all sufficiently large N. If there exists $\beta > 0$, independent of N, such that $\lambda_{\max}(G^{\mathrm{p}}) \geq \beta$, then*[1]

$$\kappa(G^{\mathrm{p}}) \geq \frac{\beta}{c}\rho^{-2(N-K)} \in \Omega(\rho^{-2N}) \qquad \text{as } N \to \infty. \tag{28}$$

[Proof in Appendix B.2.] Thus, if at least one parameter direction remains uniformly visible while another is forgotten according to Theorem 1, terminal supervision becomes exponentially ill-conditioned with increasing sequence length. To translate the directional mechanism into a global conditioning result, we consider a weak-information subspace $\mathcal{W}$ containing the parameter directions that are insufficiently observed under terminal supervision.

Proposition 2 (Conditioning under complementary auxiliary supervision)**.** *Suppose that there exists a subspace $\mathcal{W} \subseteq \mathbb{R}^o$ and constants $m, \mu > 0$, independent of N, such that*

$$G^{\mathrm{p}} \succeq m\Pi_{\mathcal{W}^\perp}, \qquad G^{\mathrm{a}} \succeq \mu\Pi_{\mathcal{W}}, \tag{29}$$

where $\Pi_{\mathcal{W}}$ and $\Pi_{\mathcal{W}^\perp}$ denote the orthogonal projections onto $\mathcal{W}$ and $\mathcal{W}^\perp$, respectively. Then, for every $\lambda > 0$,

$$\lambda_{\min}(G_\lambda) \geq \min\{m, \lambda\mu\}. \tag{30}$$

If, additionally, there exists $L_G > 0$, independent of N, such that $\|G_\lambda\| \leq L_G$, then

$$\kappa(G_\lambda) \leq L_G/\min\{m, \lambda\mu\}. \tag{31}$$

Hence, the conditioning of G_λ remains uniformly bounded with increasing sequence length.

[Proof in Appendix B.3.] Thus, auxiliary supervision can prevent the deterioration in conditioning with increasing sequence length when the added parameter information complements directions that are weakly visible under terminal supervision. In particular, together with Corollary 1, this yields the qualitative contrast

$$\kappa(G^{\mathrm{p}}) \in \Omega(\rho^{-2N}), \qquad \kappa(G_\lambda) \in \mathcal{O}(1), \tag{32}$$

provided that the respective assumptions hold. The following example illustrates the main implications of Theorem 1, Corollary 1, and Proposition 2 in a deliberately simple setting that permits an explicit analytical characterization; benchmark experiments are presented in Section 3.

Example 1. Consider the scalar recurrent model $h_t = ah_{t-1} + bx_t$, $y_t = h_t + c$, with $a \in (0, 1)$, and two sequences whose only nonzero input occurs at time K. For $\theta = [b, c]^\top$, terminal supervision yields $\kappa(G^p) = a^{-2(N-K)}$, whereas an auxiliary observation at K gives $\kappa(G_\lambda) \leq \frac{1+\lambda}{\lambda}$. Thus, the example explicitly exhibits the exponential loss of terminal parameter information and its prevention by auxiliary supervision. Details are given in Appendix A.4.

2.3 Primary-task performance trade-off

The preceding analysis shows that auxiliary supervision can improve parameter observability and conditioning. However, because it modifies the training objective, it may also shift the optimum away from a minimizer of the primary objective. We next quantify this trade-off.

Let $P^\star := \inf_\theta P(\theta)$, and, for every $\epsilon > 0$, $\Theta_{\mathrm{p}}^\epsilon := \{\theta \mid P(\theta) \leq P^\star + \epsilon\}$. We emphasize that this formulation does *not* require the infimum $P^\star$ to be attained and thus also covers cases where $P^\star$ may only be approached as $\|\theta\| \to \infty$. This occurs, for example, in unregularized classification with cross-entropy loss on separable data (Soudry et al., 2018). Accordingly, define $A_{\mathrm{p}}^\star := \lim_{\epsilon\to 0^+} \inf_{\theta\in\Theta_{\mathrm{p}}^\epsilon,\xi} A(\theta, \xi)$. Therefore, $A_{\mathrm{p}}^\star$ denotes the smallest auxiliary loss achievable along parameter choices that become arbitrarily close to optimal for the primary task.

Proposition 3 (Primary-task performance trade-off)**.** *Let $A \geq 0$ and suppose that, for some $\lambda > 0$, the objective J_λ admits a global minimizer $(\theta_\lambda^\star, \xi_\lambda^\star)$. Then,*

$$0 \leq P(\theta_\lambda^\star) - P^\star \leq \lambda A_{\mathrm{p}}^\star. \tag{33}$$

In particular, if $A_{\mathrm{p}}^\star = 0$, auxiliary supervision does not increase the optimal primary loss.

Moreover, for $0 < \lambda_1 < \lambda_2$, let $(\theta_{\lambda_j}^\star, \xi_{\lambda_j}^\star)$ be global minimizers of J_{λ_j} for $j \in \{1, 2\}$. Then,

$$A(\theta_{\lambda_2}^\star, \xi_{\lambda_2}^\star) \leq A(\theta_{\lambda_1}^\star, \xi_{\lambda_1}^\star), \qquad \text{while} \qquad P(\theta_{\lambda_2}^\star) \geq P(\theta_{\lambda_1}^\star). \tag{34}$$

[1] We use standard Landau notation Ω for asymptotic lower bounds.

[Proof in Appendix B.4.] Proposition 3 characterizes how the weight λ balances primary and auxiliary performance. Indeed, increasing λ favors solutions with smaller auxiliary loss, but can increase the corresponding primary loss. At the same time, (33) shows that the potential degradation of the primary objective is controlled by the compatibility of the auxiliary objective with parameterizations that are arbitrarily close to primary-optimal. In particular, if $A^{\star}_{\mathrm{p}} = 0$, auxiliary supervision may reshape the optimization geometry without sacrificing primary-task performance.

3 EXPERIMENTS

We investigate the mechanisms underlying auxiliary supervision and its practical consequences on *ListOps* (Nangia & Bowman, 2018), a standard task for hierarchical sequence processing that was subsequently included in the Long Range Arena benchmark (Tay et al., 2021). ListOps is a 10-way sequence classification task in which the model predicts the value of a hierarchically nested expression, making it particularly suitable for our setting: whenever a proper subexpression is completed during a causal rollout, its value is known and provides a natural intermediate supervision target. Following the original task definition, we generate ListOps instances with prescribed sequence lengths.

We consider a causal six-layer Mamba-2 sequence-to-vector model (Dao & Gu, 2024). The primary output predicts the value of the complete expression, while auxiliary outputs predict the values of completed proper subexpressions. Unless stated otherwise, auxiliary predictions at intermediate positions use the terminal classification readout mapping. Further details on the data generation, architecture, and training setup are provided in Appendix C.

The experiments address four complementary questions motivated by the preceding analysis: whether parameter information available at intermediate positions is attenuated during recurrent propagation (Theorem 1), whether auxiliary observations reveal complementary parameter-sensitivity directions (Proposition 1), whether the resulting information enrichment is accompanied by faster primary-task optimization (Proposition 2), and how the design of auxiliary supervision affects the resulting performance.

Dynamical forgetting. To isolate the effect of dynamical forgetting, we consider a frozen terminal-only model trained at $N = 72$ and focus on the component of the parameter sensitivity carried by the recurrent state, excluding direct parameter dependence of the readout. For a completed proper subexpression at position r, we define the local sensitivity information as $I_{\mathrm{loc}}(r) := \|C^{\mathrm{a}}_r \Gamma_r\|_F^2$ and compare it with the sensitivity information remaining after propagation to the terminal output, $I_{\mathrm{prop}}(r) := \|C^{\mathrm{p}} \phi(N, r) \Gamma_r\|_F^2$. We evaluate these quantities at four uniformly sampled proper subexpression positions in each of the first 1,000 sequences of the fixed held-out test set. To visualize their dependence on temporal distance, we partition the distances into eight equal-width intervals and report the median sensitivity within each interval. Figure 2 (left) shows that the local sensitivity information remains substantial, whereas the propagated information is increasingly attenuated for positions farther from the terminal output. This illustrates the dynamical-forgetting mechanism characterized in Theorem 1. In Appendix C, we additionally report the relative propagated sensitivity for all intermediate positions, which exhibits the same distance-dependent attenuation.

Parameter observability. We next investigate whether auxiliary observations reveal parameter directions that are only weakly represented by terminal observations. Using the same frozen terminal-only model, we construct the terminal sensitivity matrix S^p in (9) and the augmented sensitivity matrix $\mathcal{S}$ in (17). To distinguish the effect of intermediate observations from the mere increase in the number of matrix rows, we additionally construct a row-count-matched terminal sensitivity matrix by augmenting S^p with terminal sensitivities from fresh held-out sequences. Figure 2 (middle) compares the corresponding normalized singular-value spectra σ_j/σ_1. While increasing the number of observations lifts the spectrum relative to the terminal sensitivity S^{p}, the augmented sensitivity matrix $\mathcal{S}$ exhibits a slower spectral decay than the row-count-matched terminal sensitivity matrix. Hence, intermediate observations provide complementary parameter-sensitivity information that is not recovered simply by adding the same number of terminal observations, consistent with Proposition 1. A quantitative comparison in terms of the relative numerical rank is provided in Appendix C.

Optimization. Here, we investigate whether the additional parameter information provided by auxiliary supervision translates into improved numerical optimization behavior. To this end, we compare terminal-only training and training under auxiliary supervision at $N = 72$ over five paired

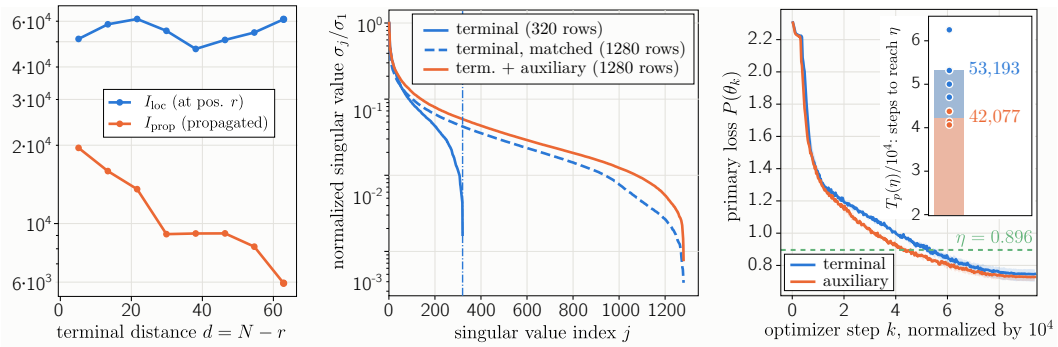


Figure 2: Numerical evaluation on ListOps. *Left:* Local parameter information I_{loc} and the corresponding propagated information I_{prop} at the terminal position. Curves show medians over eight equal-width distance intervals. *Middle:* Normalized singular-value spectra σ_j/σ_1 of the terminal sensitivity matrix S^p, the row-count-matched terminal sensitivity, and the terminal-plus-auxiliary sensitivity matrix $\mathcal{S}$. *Right:* Primary loss during training under terminal and auxiliary supervision. Curves show the median over five paired random seeds, with shaded regions indicating the interquartile range. The inset reports the number of optimizer updates $T_P(\eta)$ required to reach the reference loss level $\eta = 0.896$ for each seed (dots) and on average (bars).

random seeds using identical optimization budgets. Figure 2 (right) shows the evolution of the primary loss, evaluated periodically on a fixed subset of the training data at the current parameter iterate θ_k. We find that auxiliary supervision consistently reduces this loss faster than terminal supervision. To quantify this observation, we use the per-minibatch primary loss recorded during training, smoothed with a centered moving average over 100 optimizer steps. We define $\eta = 0.896$ as the mean smoothed terminal-supervision loss after 60% of the training budget and let $T_P(\eta)$ denote the first optimizer step at which the corresponding smoothed loss reaches this level. On average, terminal supervision requires 53,193 updates, compared with 42,077 updates under auxiliary supervision, corresponding to approximately 21% fewer updates. This empirical speedup is consistent with the optimization implications characterized in Proposition 2.

Ablations. Appendix C further investigates the robustness and design of auxiliary supervision. Varying the auxiliary-loss weight λ reveals the trade-off analyzed in Propositions 2 and 3: moderate weights yield the strongest optimization and validation performance, whereas excessively large weights reduce the benefit of auxiliary supervision. Supervising only 25–50% of the available positions retains much of the optimization gain. True subexpression targets outperform repeated terminal labels, while the shared-readout configuration performs better than the separate-head variant, suggesting that both target information and architectural integration matter.

Summary. Overall, the numerical results illustrate the mechanisms developed above: parameter sensitivity can be attenuated before reaching the terminal output, while auxiliary observations can expose weakly represented parameter directions. In training, auxiliary supervision yields faster optimization and improved validation performance. The ablations further show that these benefits depend on weighting, placement, and target information, with sparse informative supervision reducing the effort required to reach a given primary-loss level at essentially no additional computational overhead.

4 Conclusions

This paper developed a sensitivity-based framework for understanding auxiliary supervision in recurrent sequence models. We showed that parameter information available at intermediate positions can be attenuated by the recurrent dynamics, whereas suitable auxiliary observations can expose weakly observed parameter directions and improve the conditioning of the training problem. At the same time, the auxiliary objective introduces a trade-off with primary-task performance that depends on its compatibility and weighting. Numerical experiments support these mechanisms and demonstrate faster optimization and improved predictive performance with appropriately designed auxiliary supervision.

AI USE STATEMENT

In preparing this work, we used OpenAI's ChatGPT and Anthropic's Claude to support the development, implementation, analysis, and presentation of the work. Specifically, ChatGPT was used to discuss and refine research ideas and the conceptual and theoretical framing, support the formulation and presentation of mathematical claims and proof arguments, provide feedback on the experimental methodology, assist in auditing experimental implementations and result artifacts, discuss the interpretation and presentation of experimental findings, identify potentially relevant literature, and assist with the structure, drafting, and language editing of the manuscript. Claude was used to support the implementation and auditing of the experimental code and associated result artifacts. Generative AI was not used to generate the datasets or autonomously execute the reported experiments. All AI-assisted content was reviewed and revised by the authors, who take full responsibility for the final manuscript and its scientific claims.

REFERENCES

Gustavo Aguilar, Yuan Ling, Yu Zhang, Benjamin Yao, Xing Fan, and Chenlei Guo. Knowledge distillation from internal representations. *Proc. AAAI Conf. Artif. Intell.*, 34(05):7350–7357, 2020. doi: 10.1609/aaai.v34i05.6229.

Bilal Alsallakh, Amin Jourabloo, Mao Ye, Xiaoming Liu, and Liu Ren. Do convolutional neural networks learn class hierarchy? *IEEE Trans. Vis. Comput. Graphics*, 24(1):152–162, 2018. doi: 10.1109/tvcg.2017.2744683.

Pranjal Awasthi, Anupam Gupta, and Ravi Kumar. Length generalization via auxiliary tasks. In *Adv. Neur. Inf. Process. Syst. (NeurIPS)*, 2025. URL https://openreview.net/forum?id=QL3J1fyAFv.

Gerben I. Beintema, Maarten Schoukens, and Roland Tóth. Deep subspace encoders for nonlinear system identification. *Automatica*, 156:111210, 2023. doi: 10.1016/j.automatica.2023.111210.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.*, 5(2):157–166, 1994. doi: 10.1109/72.279181.

Emre Cakir, Giambattista Parascandolo, Toni Heittola, Heikki Huttunen, and Tuomas Virtanen. Convolutional recurrent neural networks for polyphonic sound event detection. *IEEE/ACM Trans. Audio, Speech, Language Process.*, 25(6):1291–1303, 2017. doi: 10.1109/taslp.2017.2690575.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proc. Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, pp. 1724–1734, 2014. doi: 10.3115/v1/d14-1179.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *Proc. 41st Int. Conf. Mach. Learn. (ICML)*, volume 235 of *PMLR*, pp. 10041–10071, 2024. URL https://proceedings.mlr.press/v235/dao24a.html.

Yunshu Du, Wojciech M. Czarnecki, Siddhant M. Jayakumar, Mehrdad Farajtabar, Razvan Pascanu, and Balaji Lakshminarayanan. Adapting auxiliary losses using gradient similarity. *arXiv:1812.02224*, 2018. doi: 10.48550/arxiv.1812.02224.

Marco Forgione and Dario Piga. Continuous-time system identification with neural networks: Model structures and fitting criteria. *Eur. J. Control*, 59:69–81, 2021. doi: 10.1016/j.ejcon.2021.01.008.

Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *Proc. Int. Conf. Mach. Learn. (ICML)*, volume 80 of *PMLR*, pp. 1607–1616, 2018. URL https://proceedings.mlr.press/v80/furlanello18a.html.

Anirudh Goyal, Alex Lamb, Ying Zhang, Saizheng Zhang, Aaron Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. In *Adv. Neur. Inf. Process. Syst. (NeurIPS)*, volume 29, 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/16026d60ff9b54410b3435b403afd226-Paper.pdf.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv:2312.00752*, 2024. URL https://openreview.net/forum?id=AL1fq05o7H.

Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara. Recurrent neural networks for time series forecasting: Current status and future directions. *Int. J. Forecasting*, 37(1):388–427, 2021. doi: 10.1016/j.ijforecast.2020.06.008.

Christian Himpe and Mario Ohlberger. Cross-gramian-based combined state and parameter reduction for large-scale control systems. *Math. Probl. Eng.*, 2014(1), 2014. doi: 10.1155/2014/843869.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *NIPS 2014 Deep Learning Workshop*, 2015. doi: 10.48550/arxiv.1503.02531.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data Min. Knowl. Discov.*, 33(4): 917–963, 2019. ISSN 1573-756X. doi: 10.1007/s10618-019-00619-1.

Anil Kag and Venkatesh Saligrama. Training recurrent neural networks via forward propagation through time. In *Proc. 38th Int. Conf. Mach. Learn. (ICML)*, volume 139 of *PMLR*, pp. 5189–5200, 2021. URL https://proceedings.mlr.press/v139/kag21a.html.

James Clayton Kerce. Stream separation improves Bregman conditioning in transformers. *arXiv:2603.21317*, 2026. doi: 10.48550/arxiv.2603.21317.

Aakash Lahoti, Kevin Li, Berlin Chen, Caitlin Wang, Aviv Bick, J. Zico Kolter, Tri Dao, and Albert Gu. Mamba-3: Improved sequence modeling using state space principles. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2026. URL https://openreview.net/forum?id=HwCvaJOiCj.

Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-Supervised Nets. In *Proc. 18th Int. Conf. Artif. Intell. Stat.*, volume 38 of *PMLR*, pp. 562–570, 2015. URL https://proceedings.mlr.press/v38/lee15a.html.

Kunchang Li, Xinhao Li, Yi Wang, Yinan He, Yali Wang, Limin Wang, and Yu Qiao. VideoMamba: State space model for efficient video understanding. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pp. 237–255, 2024. doi: 10.1007/978-3-031-73347-5_14.

Eran Malach, Omid Saremi, Sinead Williamson, Arwen Bradley, Aryo Lotfi, Emmanuel Abbe, Joshua M. Susskind, and Etai Littwin. To infinity and beyond: Tool-use unlocks length generalization in state space models. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2026. URL https://openreview.net/forum?id=sSfep4udCb.

Daniele Masti and Alberto Bemporad. Learning nonlinear state-space models using autoencoders. *Automatica*, 129:109666, 2021. doi: 10.1016/j.automatica.2021.109666.

Navid Mohammadi Foumani, Lynn Miller, Chang Wei Tan, Geoffrey I. Webb, Germain Forestier, and Mahsa Salehi. Deep learning for time series classification and extrinsic regression: A current survey. *ACM Computing Surveys*, 56(9):1–45, 2024. doi: 10.1145/3649448.

Nikita Nangia and Samuel Bowman. ListOps: A diagnostic dataset for latent tree learning. In *Proc. 2018 Conf. N. Am. Chapter Assoc. Comput. Linguist.: Stud. Res. Workshop*, pp. 92–99, 2018. doi: 10.18653/v1/n18-4013.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, NY, USA, 2006. doi: 10.1007/978-0-387-40065-5.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *Proc. Int. Conf. Mach. Learn. (ICML)*, volume 28 of *PMLR*, pp. 1310–1318, 2013. URL https://proceedings.mlr.press/v28/pascanu13.html.

Luc Pronzato. Optimal experimental design and some related control problems. *Automatica*, 44(2): 303–325, 2008. doi: 10.1016/j.automatica.2007.05.016.

Jens Püttschneider, Simon Heilig, Asja Fischer, and Timm Faulwasser. Towards the optimal control perspective of ResNet training. In *HiLD 2025: 3rd Workshop on High-dimensional Learning Dynamics*, 2025. URL https://openreview.net/forum?id=xMXpVetlhj.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints for thin deep nets. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015. URL https://openreview.net/forum?id=SWN-Izy8Sl9.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *Int. J. Forecasting*, 36(3):1181–1191, 2020. doi: 10.1016/j.ijforecast.2019.07.001.

Rachit Saluja, Asli Cihangir, Ruining Deng, Johannes C. Paetzold, Fengbei Liu, and Mert R. Sabuncu. BackSplit: The importance of sub-dividing the background in biomedical lesion segmentation. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 8492–8502, 2026.

Julian D. Schiller, Malte Heinrich, Victor G. Lopez, and Matthias A. Müller. Tuning the burn-in phase in training recurrent neural networks improves their performance. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2026. doi: 10.48550/arXiv.2602.10911. URL https://openreview.net/forum?id=jwkdKpioHJ.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *J. Mach. Learn. Res.*, 19(70):1–57, 2018. URL http://jmlr.org/papers/v19/18-188.html.

Yanru Sun, Zongxia Xie, Dongyue Chen, Emadeldeen Eldele, and Qinghua Hu. Hierarchical classification auxiliary network for time series forecasting. *Proc. AAAI Conf. Artif. Intell.*, 39(19): 20743–20751, 2025. doi: 10.1609/aaai.v39i19.34286.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015. doi: 10.1109/cvpr.2015.7298594.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long Range Arena: A benchmark for efficient transformers. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021. URL https://openreview.net/forum?id=qVyeW-grC2k.

Trieu H. Trinh, Andrew M. Dai, Minh-Thang Luong, and Quoc V. Le. Learning longer-term dependencies in RNNs with auxiliary losses. In *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, volume 80 of *PMLR*, pp. 4965–4974, 2018. doi: 10.48550/arxiv.1803.00144.

L. G. Van Willigenburg, J. D. Stigter, and J. Molenaar. Sensitivity matrices as keys to local structural system properties of large-scale nonlinear systems. *Nonlinear Dynam.*, 107(3):2599–2618, 2022. doi: 10.1007/s11071-021-07125-4.

Jiacheng Wang, Liang Fan, Baihua Li, and Luyan Zhang. Forecasting with guidance: Representation-level supervision for time series forecasting. *arXiv preprint*, 2026. doi: 10.48550/arXiv.2603.24262.

Noam Wies, Yoav Levine, and Amnon Shashua. Sub-task decomposition enables learning in sequence to sequence tasks. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2023. URL https://openreview.net/forum?id=BrJATVZDWEH.

Xueqing Wu, Lewen Wang, Yingce Xia, Weiqing Liu, Lijun Wu, Shufang Xie, Tao Qin, and Tie-Yan Liu. Temporally correlated task scheduling for sequence learning. In *Proc. Int. Conf. Mach. Learn. (ICML)*, volume 139 of *PMLR*, pp. 11274–11284, 2021. URL https://proceedings.mlr.press/v139/wu21e.html.

Yong Xu, Qiuqiang Kong, Wenwu Wang, and Mark D. Plumbley. Large-scale weakly supervised audio classification using gated convolutional neural network. In *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, pp. 121–125, 2018. doi: 10.1109/ICASSP.2018.8461975.

Xiangdong Zhang, Debing Zhang, Shaofeng Zhang, Xiaohan Qin, Yu Cheng, and Junchi Yan. NITP: Next implicit token prediction for LLM pre-training. In *Proc. Int. Conf. Mach. Learn. (ICML)*, 2026. URL https://openreview.net/forum?id=oPQmCKS1tV.

Jim Zhao, Sidak Pal Singh, and Aurelien Lucchi. Theoretical characterisation of the Gauss-Newton conditioning in neural networks. In *Adv. Neur. Inf. Process. Syst. (NeurIPS)*, 2024. URL https://openreview.net/forum?id=fpOnUMjLiO.

Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision Mamba: Efficient visual representation learning with bidirectional state space model. In *Proc. 41st Int. Conf. Mach. Learn. (ICML)*, volume 235 of *PMLR*, pp. 62429–62442, 2024. URL https://proceedings.mlr.press/v235/zhu24f.html.

Nicolas Zucchet and Antonio Orvieto. Recurrent neural networks: vanishing and exploding gradients are not the end of the story. In *Adv. Neur. Inf. Process. Syst. (NeurIPS)*, volume 38, 2024. URL https://openreview.net/forum?id=46Jr4sgTWa.

A Additional remarks and illustrations

A.1 Extensions of the framework

The problem setup described in Section 2.1 admits several straightforward extensions:

Sample-dependent sequence lengths. One may consider input sequences X_i whose lengths N_i vary across the training samples $i \in [1, B]$, together with $\mathcal{R}_i \subseteq [1, N_i - 1]$ and, if required, sample-dependent auxiliary loss functions $l^{\mathrm{a}}_{i,r}$.

Layer-wise auxiliary supervision. Auxiliary readouts may also be indexed jointly by temporal position and layer in a multilayer recurrent sequence model. Such layer-wise auxiliary supervision is closely related to deep supervision in neural networks (Lee et al., 2015; Szegedy et al., 2015; Püttschneider et al., 2025). The observability arguments in Section 2.2 then extend analogously.

Bidirectional sequence models. Bidirectional recurrent and state-space architectures are widely used for sequence-level representation and classification tasks; see, e.g., Zhu et al. (2024). The sensitivity-based framework can be adapted by accounting for both forward and backward recurrent states in the primary and auxiliary readouts, while the analysis of dynamical forgetting would need to distinguish the two propagation directions.

We restrict the main exposition in Section 2 to common sequence lengths, temporal auxiliary supervision, and causal recurrent models to keep the notation and analysis concise.

A.2 Illustration of Proposition 1

Example 2. Consider a SISO selective state-space model typically underlying Mamba, with $d_x = d_y = 1$ and SSM state dimension d_h. A characteristic feature of Mamba is that the readout matrix C depends on the current input through a learned affine projection (Gu & Dao, 2024). Accordingly, consider

$$g(h, x; \theta) = C(x; \theta)h,$$

where

$$C(x; \theta) = (W_C x + b_C)^\top,$$

with $W_C, b_C \in \mathbb{R}^{d_h}$ denoting the weight and bias of the projection generating the input-dependent readout. We assume that W_C and b_C are contained in the overall parameter vector θ and enter only the readout, i.e., they are not shared with the recurrent dynamics.

For any time $j \in [1, N]$, the corresponding direct parameter sensitivities are

$$\frac{\partial g}{\partial W_C}(h_j, x_j; \theta) = x_j h_j^\top, \qquad \frac{\partial g}{\partial b_C}(h_j, x_j; \theta) = h_j^\top. \tag{35}$$

Hence, if $x_N = 0$, terminal supervision contains no direct information about W_C. More precisely, consider a parameter direction $v^\star \in \mathbb{R}^o$ supported only on the W_C block. Since W_C does not enter the recurrent dynamics, $\Gamma_N v^\star = 0$, while $x_N = 0$ implies $D^{\mathrm{p}} v^\star = 0$. Thus,

$$S^{\mathrm{p}} v^\star = 0.$$

Consequently, terminal supervision leaves the parameters collected in W_C unconstrained, although they can become active along unseen rollouts with $x_N \neq 0$ and thereby affect the resulting predictions.

Now choose the auxiliary readouts as $q_r = g$, $r \in \mathcal{R}$, and suppose that there exists $K \in \mathcal{R}$ with $x_K \neq 0$ and $h_K \neq 0$. Choosing, for example, $v^\star$ with W_C-component $h_K / \|h_K\|$ and all other components zero and using (35) yields

$$S_K^{\mathrm{a}} v^\star = D_K^{\mathrm{a}} v^\star = \frac{\partial g}{\partial W_C}(h_K, x_K; \theta) \frac{h_K}{\|h_K\|} = x_K \|h_K\| \neq 0.$$

Thus,

$$v^\star \in \ker(S^{\mathrm{p}}) \qquad \text{but} \qquad v^\star \notin \ker(\mathcal{S}),$$

and therefore

$$\ker(\mathcal{S}) \subset \ker(S^{\mathrm{p}}).$$

Hence, an auxiliary observation at a time with nonzero input can reveal input-dependent readout parameters that are locally unobservable under terminal supervision alone.

A.3 Conditioning and gradient-based optimization

The condition number characterizes the anisotropy of the local objective curvature and is closely related to the convergence properties of gradient-based optimization. To illustrate this connection explicitly, consider a strongly convex quadratic objective $J(\theta)$ with Hessian $G \succ 0$ and spectral condition number

$$\kappa(G) := \frac{\lambda_{\max}(G)}{\lambda_{\min}(G)}.$$

Let $\theta^\star$ denote its unique minimizer. For steepest descent with exact line search, the weighted error $\|\theta^k - \theta^\star\|_G^2 = (\theta^k - \theta^\star)^\top G (\theta^k - \theta^\star)$ satisfies

$$\|\theta^{k+1} - \theta^\star\|_G^2 \leq q^2 \|\theta^k - \theta^\star\|_G^2, \qquad q := \frac{\lambda_{\max}(G) - \lambda_{\min}(G)}{\lambda_{\max}(G) + \lambda_{\min}(G)} = \frac{\kappa(G) - 1}{\kappa(G) + 1} \in [0, 1), \tag{36}$$

showing that the convergence rate deteriorates as $\kappa(G)$ increases (Nocedal & Wright, 2006, Th. 3.3).

An analogous local result holds for sufficiently smooth nonlinear objectives near a minimizer with positive-definite Hessian (Nocedal & Wright, 2006, Th. 3.4). In particular, for any $r \in (q, 1)$ and all $k \geq k_0$ with $k_0 \geq 0$ sufficiently large,

$$J(\theta^{k+1}) - J(\theta^\star) \leq r^2 \left(J(\theta^k) - J(\theta^\star) \right), \tag{37}$$

which directly implies that

$$J(\theta^k) - J(\theta^\star) \leq r^{2(k-k_0)} \left(J(\theta^{k_0}) - J(\theta^\star) \right). \tag{38}$$

Thus, an ϵ-optimal solution satisfying $J(\theta^k) - J(\theta^\star) \leq \epsilon$ for a given $\epsilon > 0$ is guaranteed for

$$k \geq k_0 + \frac{\log\left(\epsilon / (J(\theta^{k_0}) - J(\theta^\star)) \right)}{2 \log(r)}.$$

Consequently, poorer conditioning can require significantly more gradient iterations to achieve a prescribed level of suboptimality. These classical results provide an optimization-theoretic interpretation of the conditioning properties analyzed in Section 2.2.

A.4 Example 1: full version

Consider the scalar recurrent model

$$h_t = ah_{t-1} + bx_t, \qquad y_t = h_t + c, \qquad h_0 = 0, \tag{39}$$

where $a \in (0,1)$ is fixed and $\theta = [b, c]^\top$ contains the learnable parameters. Consider two training sequences satisfying $x_{1,K} = 1$, $x_{2,K} = -1$, and $x_{i,t} = 0$ for $t \neq K$, $i = 1, 2$ for some fixed $1 \leq K < N$. The terminal and auxiliary targets are generated by (39) for some parameter $\theta^\star$. We consider the squared-error loss $l^{\mathrm{p}}(y, \tilde{y}) = \frac{1}{2}(y - \tilde{y})^2$ and a single auxiliary observation at time K, i.e., $\mathcal{R} = \{K\}$, with $q_K = g$ and $l^{\mathrm{a}}_K = l^{\mathrm{p}}$. Consequently, $\theta^\star$ yields zero primary and auxiliary loss.

For the parameter direction $v^\star = [1, 0]^\top$, it holds that $B_k v^\star = 0$ for all $k > K$, while $\phi(t, K) = a^{t-K}$. Thus, the example satisfies precisely the forgetting and no-re-excitation mechanism of Theorem 1; see (21). The terminal sensitivities of the two training samples are $S^{\mathrm{p}}_1 = [a^{N-K}, 1]$, and $S^{\mathrm{p}}_2 = [-a^{N-K}, 1]$, respectively. Averaging the corresponding Gauss–Newton contributions over the two training samples in (11) yields

$$G^{\mathrm{p}} = \frac{1}{2}\left({S^{\mathrm{p}}_1}^\top S^{\mathrm{p}}_1 + {S^{\mathrm{p}}_2}^\top S^{\mathrm{p}}_2\right) = \begin{bmatrix} a^{2(N-K)} & 0 \\ 0 & 1 \end{bmatrix} \qquad \Rightarrow \qquad \kappa(G^{\mathrm{p}}) = a^{-2(N-K)}. \tag{40}$$

Consequently, $\kappa(G^{\mathrm{p}})$ grows exponentially with the distance between the excitation time K and the terminal output, illustrating Theorem 1 and Corollary 1.

In contrast, the sensitivities at the auxiliary supervision time K are $S^{\mathrm{a}}_{1,K} = [1, 1]$ and $S^{\mathrm{a}}_{2,K} = [-1, 1]$. Hence, $G^{\mathrm{a}} = \frac{1}{2}({S^{\mathrm{a}}_{1,K}}^\top S^{\mathrm{a}}_{1,K} + {S^{\mathrm{a}}_{2,K}}^\top S^{\mathrm{a}}_{2,K}) = I$, from which it follows that

$$G_\lambda = \begin{bmatrix} a^{2(N-K)} + \lambda & 0 \\ 0 & 1 + \lambda \end{bmatrix} \qquad \Rightarrow \qquad \kappa(G_\lambda) = \frac{1 + \lambda}{a^{2(N-K)} + \lambda} \leq \frac{1 + \lambda}{\lambda}. \tag{41}$$

In particular, Proposition 2 applies with $\mathcal{W} = \operatorname{span}\{v^\star\}$ and $m = \mu = 1$. Thus, while terminal supervision becomes exponentially ill-conditioned as $N - K$ increases, the conditioning under auxiliary supervision remains uniformly bounded; see (40) and (41) and compare (32). The auxiliary observation at time K preserves the parameter information in direction $v^\star$ before it is attenuated by the recurrent dynamics.

B Mathematical proofs

B.1 Proof of Theorem 1

Proof. Recall that condition (21) applied to the definition of Γ_t in (16) yields (19). Taking norms and using (20) then leads to

$$\|\Gamma_N v^\star\| \leq C\rho^{N-K}\|\Gamma_K v^\star\|. \tag{42}$$

The sensitivity representation in (15) with $D^{\mathrm{p}} v^\star = 0$ (22) and the first upper bound in (23) then implies that

$$\|S^{\mathrm{p}} v^\star\| \leq L_C C\rho^{N-K}\|\Gamma_K v^\star\|. \tag{43}$$

Together with the definition of G^{p} in (11) and the uniform upper bound of the Hessian in (23), this yields

$$v^{\star\top} G^{\mathrm{p}} v^\star = v^{\star\top} S^{\mathrm{p}\top} H^{\mathrm{p}} S^{\mathrm{p}} v^\star \leq L_H L_C^2 C^2 \|\Gamma_K v^\star\|^2 \rho^{2(N-K)}, \tag{44}$$

which proves (24) with $c := L_H L_C^2 C^2 \|\Gamma_K v^\star\|^2$, independent of N.

For $K \in \mathcal{R}$, positive semidefiniteness of the individual terms in (11) implies

$$G_\lambda \succeq \lambda G^{\mathrm{a}} = \frac{\lambda}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} S^{\mathrm{a}\top}_r H^{\mathrm{a}}_r S^{\mathrm{a}}_r \succeq \frac{\lambda}{|\mathcal{R}|} S^{\mathrm{a}\top}_K H^{\mathrm{a}}_K S^{\mathrm{a}}_K. \tag{45}$$

Pre- and post-multiplying (45) by $v^{\star\top}$ and $v^\star$, respectively, and using (25) yields (26), thus completing the proof. □

B.2 Proof of Corollary 1

Proof. By the Rayleigh quotient characterization and Theorem 1, we have that

$$\lambda_{\min}(G^{\mathrm{p}}) \le v^{\star\top} G^{\mathrm{p}} v^{\star} \le c\rho^{2(N-K)}. \tag{46}$$

Together with $\lambda_{\max}(G^{\mathrm{p}}) \ge \beta$, this directly yields (28), thus completing the proof. □

B.3 Proof of Proposition 2

Proof. Using the definition of G_λ in (11) and the conditions in (29), we obtain

$$G_\lambda = G^{\mathrm{p}} + \lambda G^{\mathrm{a}} \succeq m\Pi_{\mathcal{W}^\perp} + \lambda\mu\Pi_{\mathcal{W}} \succeq \min\{m, \lambda\mu\} I, \tag{47}$$

which establishes (30). Assuming additionally that $\|G_\lambda\| \le L_G$ uniformly for all N, we have $\lambda_{\max}(G_\lambda) \le L_G$, which implies that the condition number $\kappa(G_\lambda) = \lambda_{\max}(G_\lambda)/\lambda_{\min}(G_\lambda)$ can be bounded by (31), thus completing the proof. □

B.4 Proof of Proposition 3

Proof. Consider an arbitrary $\epsilon > 0$. By global optimality of $(\theta^\star_\lambda, \xi^\star_\lambda)$ and since $A \ge 0$, it follows that

$$P(\theta^\star_\lambda) \le J_\lambda(\theta^\star_\lambda, \xi^\star_\lambda) \le \inf_{\theta \in \Theta^\epsilon_{\mathrm{p}},\, \xi} J_\lambda(\theta, \xi) \le P^\star + \epsilon + \lambda \inf_{\theta \in \Theta^\epsilon_{\mathrm{p}}, \xi} A(\theta, \xi). \tag{48}$$

Consequently,

$$0 \le P(\theta^\star_\lambda) - P^\star \le \epsilon + \lambda \inf_{\theta \in \Theta^\epsilon_{\mathrm{p}}, \xi} A(\theta, \xi). \tag{49}$$

Letting $\epsilon \to 0^+$ yields

$$P(\theta^\star_\lambda) - P^\star \le \lambda \lim_{\epsilon \to 0^+} \inf_{\theta \in \Theta^\epsilon_{\mathrm{p}}, \xi} A(\theta, \xi). \tag{50}$$

Using the definition of $A^\star_{\mathrm{p}}$ proves (33).

For the second claim, optimality of $(\theta^\star_{\lambda_1}, \xi^\star_{\lambda_1})$ and $(\theta^\star_{\lambda_2}, \xi^\star_{\lambda_2})$ implies

$$P(\theta^\star_{\lambda_1}) + \lambda_1 A(\theta^\star_{\lambda_1}, \xi^\star_{\lambda_1}) \le P(\theta^\star_{\lambda_2}) + \lambda_1 A(\theta^\star_{\lambda_2}, \xi^\star_{\lambda_2}), \tag{51}$$

$$P(\theta^\star_{\lambda_2}) + \lambda_2 A(\theta^\star_{\lambda_2}, \xi^\star_{\lambda_2}) \le P(\theta^\star_{\lambda_1}) + \lambda_2 A(\theta^\star_{\lambda_1}, \xi^\star_{\lambda_1}). \tag{52}$$

Adding both inequalities and rearranging yields

$$(\lambda_2 - \lambda_1) A(\theta^\star_{\lambda_2}, \xi^\star_{\lambda_2}) \le (\lambda_2 - \lambda_1) A(\theta^\star_{\lambda_1}, \xi^\star_{\lambda_1}).$$

Dividing by $(\lambda_2 - \lambda_1) > 0$ establishes the first property in (34). Using this in (51) then leads to

$$P(\theta^\star_{\lambda_1}) - P(\theta^\star_{\lambda_2}) \le \lambda_1 \left(A(\theta^\star_{\lambda_2}, \xi^\star_{\lambda_2}) - A(\theta^\star_{\lambda_1}, \xi^\star_{\lambda_1}) \right) \le 0,$$

which establishes the second property and thus completes the proof. □

C Experiments on ListOps

Setup. Following the ListOps task definition (Nangia & Bowman, 2018), we generate nested expressions over the operators $\{\mathrm{MIN}, \mathrm{MAX}, \mathrm{MED}, \mathrm{SM}\}$ and digit operands in $\{0, \ldots, 9\}$. Each operator has between two and six uniformly sampled arguments, and the root is always an operator. At non-root nodes below the maximum tree depth of five, a digit is generated with probability 0.5; at the maximum depth, nodes are forced to be digits. We require every expression to contain at least two proper operator subexpressions and use rejection sampling to obtain sequences of the prescribed exact length. For each training length, we generate $400{,}000$ training, $10{,}000$ validation, and $10{,}000$ held-out test sequences.

All experiments use a causal six-layer Mamba-2 model (Dao & Gu, 2024) with model dimension $d = 128$, state dimension $d_{\mathrm{state}} = 64$, convolution kernel size 4, expansion factor 2, and head dimension 32. Models are trained for 30 epochs, corresponding to $93{,}750$ optimizer updates, with minibatch size $B = 128$ using AdamW with learning rate 10^{-4}; weight decay and dropout are

Table 1: Relative numerical rank $\mathrm{rank}_\mu(S)$ of the sensitivity matrices.

Sensitivity matrix	$\mu = 0.1$	$\mu = 0.01$
Terminal only (S^{p}, 320 rows)	87	306
Terminal only, matched rows ($S^{\mathrm{p}}_{\mathrm{matched}}$, 1,280 rows)	102	938
Terminal + auxiliary ($\mathcal{S}$, 1,280 rows)	139	1,074

disabled, gradients are clipped at 0.5, and we use a cosine learning-rate schedule with three warm-up epochs. For paired comparisons, runs use the same parameter initialization, generated data, and minibatch shuffling.

Auxiliary observations are placed at the closing positions of completed proper subexpressions, excluding the root. In the reference configuration, all such positions are supervised using the corresponding subexpression value, the terminal readout is shared with the auxiliary predictions, and the auxiliary-loss weight is $\lambda = 0.3$.

Training objective. We use multiclass cross-entropy for both the primary and auxiliary losses. Since the number of selected auxiliary positions generally differs between samples, we use the sample-dependent extension described in Appendix A.1. Specifically, letting $\mathcal{R}_i$ denote the set of auxiliary positions selected for sample i, we average the auxiliary loss first over the selected positions of each sample and subsequently over the minibatch:

$$A(\theta, \xi) = \frac{1}{B} \sum_{i=1}^{B} A_i(\theta, \xi), \qquad A_i(\theta, \xi) = \begin{cases} \dfrac{1}{|\mathcal{R}_i|} \displaystyle\sum_{r \in \mathcal{R}_i} \ell^a(z_{i,r}(\theta, \xi), \tilde{z}_{i,r}), & |\mathcal{R}_i| > 0, \\ 0, & |\mathcal{R}_i| = 0. \end{cases}$$

Here, $\ell^a = \ell^p$ denotes the multiclass cross-entropy loss. In the reference configuration, $\mathcal{R}_i \neq \emptyset$ for every sample because each generated expression contains at least two proper subexpressions. The shared-readout reference configuration has no auxiliary-only parameters ξ, whereas the separate-readout ablation (E) below introduces such parameters through the additional auxiliary head.

Dynamical forgetting: relative propagated sensitivity. To complement the absolute sensitivities reported in Figure 2, we consider the relative propagated sensitivity and define the sensitivity retention ratio

$$R(r) := \frac{I_{\mathrm{prop}}(r)}{I_{\mathrm{loc}}(r) + 10^{-12}}.$$

Figure 3 reports $R(r)$ for all considered intermediate positions together with the median over the same eight equal-width temporal-distance intervals used in Figure 2. The relative propagated sensitivity decreases with increasing distance from the terminal output, corroborating that the reduction in I_{prop} is not merely due to variation in the locally available sensitivity.

Parameter observability: relative numerical rank. To quantify the differences between the singular-value spectra of the terminal sensitivity matrix S^{p}, the row-count-matched terminal matrix $S^{\mathrm{p}}_{\mathrm{matched}}$, and the augmented sensitivity matrix $\mathcal{S}$, we consider the relative numerical rank

$$\mathrm{rank}_\mu(S) := \left| \left\{ j : \frac{\sigma_j(S)}{\sigma_1(S)} > \mu \right\} \right|,$$

corresponding to the number of singular values of S that satisfy the condition $\sigma_j(S)/\sigma_1(S) > \mu$ for a given threshold $\mu > 0$. Table 1 reports this quantity for the sensitivity matrices S^{p}, $S^{\mathrm{p}}_{\mathrm{matched}}$, and $\mathcal{S}$. At $\mu = 0.1$, the matched terminal sensitivity $S^{\mathrm{p}}_{\mathrm{matched}}$ has $\mathrm{rank}_\mu = 102$, whereas the augmented sensitivity $\mathcal{S}$ reaches 139, corresponding to a 36.3% increase. Using the less restrictive threshold $\mu = 0.01$ yields the same qualitative conclusion, with ranks 938 and 1,074, respectively, corresponding to a 14.5% increase. Hence, the higher numerical rank of the auxiliary-augmented sensitivity matrix cannot be explained solely by its larger number of rows.

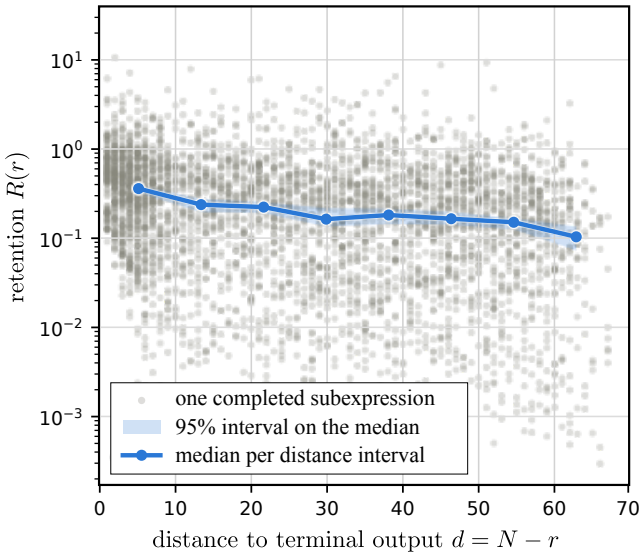


Figure 3: Relative propagation of parameter sensitivity in the dynamical-forgetting experiment at individual auxiliary-observation positions. The retention ratio $R(r)$ is shown as a function of the temporal distance from the terminal output. Markers indicate median values over the same eight equal-width distance intervals used in Figure 2, with shaded regions indicating 95% bootstrap intervals. The decrease with temporal distance shows that a progressively smaller amount of the locally available sensitivity remains visible at the terminal output.

Ablation study. We complement the main experiments with a systematic ablation study at sequence length $N = 72$, using three paired random seeds per configuration. The reference training configuration supervises all completed proper subexpressions using their true subexpression values, shares the terminal classification readout with the auxiliary predictions, and uses $\lambda = 0.3$. Unless stated otherwise, each ablation varies one design choice relative to this reference while keeping the remaining training setup fixed.

Specifically, we vary

(A) the auxiliary weight $\lambda \in \{0.03, 0.1, 0.3, 0.9, 3\}$;

(B) the fraction of available auxiliary positions $\{0.1, 0.25, 0.5, 1\}$[2];

(C) the temporal placement of a matched number of auxiliary observations, comparing early, uniformly distributed, and late positions[3];

(D) the information contained in the auxiliary target, comparing the true subexpression value with the repeated terminal label; and

(E) the auxiliary readout, comparing the shared terminal readout with a separate auxiliary head.

For each configuration, we report four quantities. The final primary accuracy is the validation accuracy at the final training epoch. Length generalization is assessed by evaluating the accuracy on a separate, disjoint held-out test set of $10{,}000$ sequences of length $N = 288$. Computational overhead is reported as the percentage change in total wall-clock training time relative to the corresponding terminal-only run. Finally, optimization speed is quantified by $T_P(\eta)$, the first optimizer step at which the smoothed primary loss reaches the baseline-derived reference level $\eta = 1.0267$, corresponding to the mean terminal-only primary loss after 40% of the training budget.

[2]The retained auxiliary positions form deterministic nested subsets, such that increasing the supervision fraction adds positions without removing those selected at lower fractions.

[3]For the placement ablation, early and late observations are selected from the first and last thirds of the sequence, respectively, while the uniform variant distributes the same number of observations approximately evenly over all eligible positions. Since the placement variants are therefore sparser than the reference configuration, they are compared primarily with one another.

Table 2: Ablations of temporal placement (C), auxiliary target (D), and readout sharing (E). Values are mean ± sample standard deviation over three paired seeds. The early, uniform, and late placement variants use a matched number of auxiliary observations and are compared primarily with one another. Values corresponding to the reference configuration are highlighted in color.

Ablation	Configuration	Final acc.	$N = 288$ acc.	Overhead [%]	$T_P(\eta)/10^4$
baseline	terminal only	0.557 ±0.002	0.380 ±0.030	0.0 ±0.0	3.55 ±0.47
C: placement	early	0.587 ±0.016	0.387 ±0.030	0.2 ±0.2	3.25 ±0.17
	uniform	0.598 ±0.004	0.395 ±0.012	0.6 ±0.6	3.02 ±0.18
	late	0.620 ±0.011	0.340 ±0.111	0.1 ±0.1	2.54 ±0.06
D: target	subexpr. value	0.612 ±0.001	0.410 ±0.022	1.0 ±0.1	2.75 ±0.21
	terminal label	0.572 ±0.013	0.396 ±0.023	1.1 ±0.03	3.71 ±0.58
E: readout	shared	0.612 ±0.001	0.410 ±0.022	1.0 ±0.1	2.75 ±0.21
	separate	0.594 ±0.011	0.400 ±0.007	0.9 ±0.4	3.33 ±0.15

Ablations A and B are visualized in Figure 4, whereas Ablations C–E are summarized in Table 2. In both cases, we report the mean across the three paired seeds, with error bars and tabulated uncertainties indicating the sample standard deviation, respectively.

The ablations reveal several consistent design effects. Varying the auxiliary-loss weight λ reveals a non-monotonic trade-off, qualitatively consistent with Propositions 2 and 3: moderate weights yield the strongest optimization and validation performance, whereas excessively large weights reduce the benefit of auxiliary supervision. The supervision-density ablation (B) shows pronounced diminishing returns: even relatively sparse supervision retains most of the benefit of supervising all available subexpressions. In contrast, the information content of the auxiliary target (D) has a substantial effect. Supervising the true subexpression value leads to faster optimization and higher primary and length-generalization accuracy than repeating the terminal label at the same intermediate positions. Sharing the terminal readout with the auxiliary predictions (E) likewise outperforms using a separate auxiliary head, suggesting that both the information content and the architectural integration of auxiliary supervision matter.

The temporal-placement ablation (C) exhibits a distinct trade-off. Among the matched-density variants, late auxiliary observations yield the fastest optimization and the highest in-distribution accuracy at $N = 72$, whereas uniformly distributed observations attain better mean accuracy on the longer $N = 288$ sequences. This behavior appears to reflect the hierarchical structure of ListOps rather than temporal distance alone. Applying the same placement rules to an independent held-out set of 10,000 $N = 72$ expressions yields mean subtree spans of 7.88, 14.73, and 36.17 tokens and mean tree depths of 3.37, 3.18, and 1.94 for early, uniform, and late placement, respectively. Thus, early observations tend to correspond to smaller, more deeply nested subexpressions, whereas late observations tend to correspond to larger, more root-adjacent subexpressions. This structural difference may contribute to the observed training-length/generalization trade-off and shows that the placement results should not be interpreted as a pure temporal-distance effect. This also highlights a practical implication of our framework: if structural knowledge identifies intermediate computations that are particularly informative for the primary task, auxiliary observations can be placed selectively at those positions rather than according to temporal proximity alone.

Finally, the measured training-time overhead remains close to zero across all configurations, indicating no substantial additional runtime in the considered implementation.

Overall, these results show that sparse but informative auxiliary supervision can improve both primary-task optimization and predictive performance with negligible training-time overhead, while its effectiveness depends critically on the weighting, target information, and placement of the auxiliary observations.

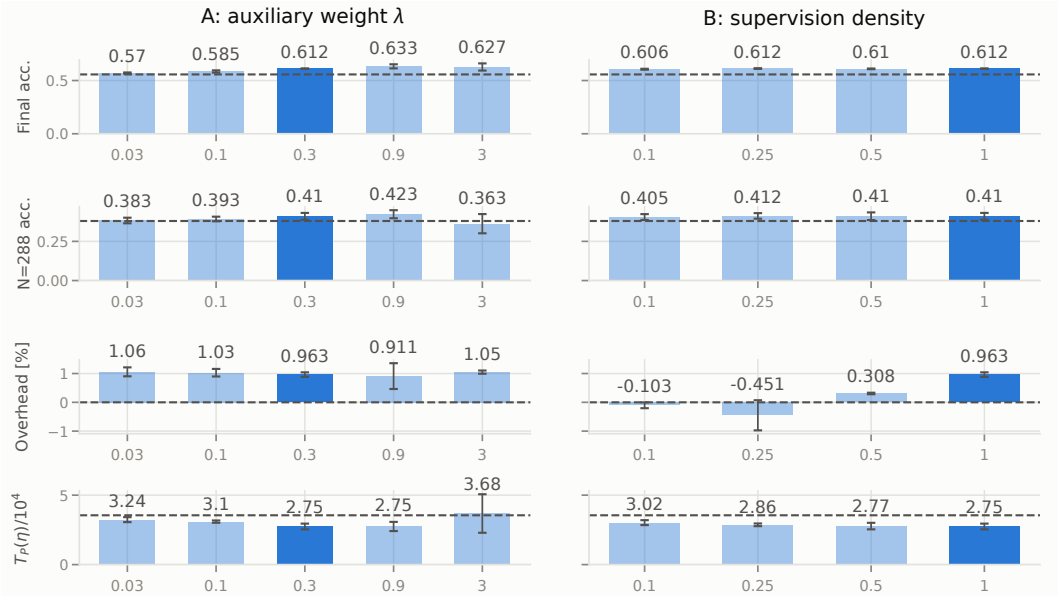


Figure 4: Ablations of the auxiliary-loss weight λ (A) and supervision density (B) at $N = 72$. Bars show the mean over three paired seeds, with error bars indicating the sample standard deviation; numerical values denote the corresponding means. Dashed lines indicate the terminal-only baseline. The reference configuration ($\lambda = 0.3$ and full supervision density) is highlighted. We report final validation accuracy, length-generalization accuracy at $N = 288$, training-time overhead relative to terminal-only training, and the optimization statistic $T_P(\eta)$ with $\eta = 1.0267$, expressed in units of 10^4 optimizer steps.