POISE: POST-TRAINING INPUT-AWARE STATE ELIMINATION FOR DEEP STATE SPACE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

State space models (SSMs) trade long-context computation for a recurrent state, making state dimension a direct cost of autoregressive inference. Existing post-training reduction methods either prune modes in the trained coordinates or apply classical balanced truncation, whose controllability Gramian reflects generic white-noise excitation rather than the inputs encountered at deployment. We introduce POISE, a label-free, post-training method that instead balances each SSM against the state covariance induced by a small set of calibration inputs and allocates a global state budget across the network. For stable linear time-invariant (LTI) SSMs, POISE yields stable reduced systems and provides error guarantees for arbitrary inputs and finite horizons. We apply it to the LTI backbones S4D, SaShiMi and S5, and extend the method empirically to selective Mamba-2. On Mamba-2 1.3B, POISE removes 70% of the recurrent states for only 0.34 additional WikiText perplexity. At a 2048-token context, the reduced model comes within 0.6 perplexity of Llama-3.2-1B with a four-bit KV cache while using less memory.

1 INTRODUCTION

State space models are a competitive alternative to attention for long-sequence modelling (Gu et al., 2022b;a; Smith et al., 2023; Orvieto et al., 2023; Gu & Dao, 2024; Dao & Gu, 2024; De et al., 2024), and their cost is governed by the state dimension n: in autoregressive inference, the state plays the role of the KV cache, requiring $O(n)$ memory per system, while the per-step computation also grows with n. Compressing an SSM therefore amounts to reducing its state dimension. One could treat it as generic post-training compression and act on the weights, by quantization or low-rank methods (Nagel et al., 2020; Frantar et al., 2023; Li et al., 2021; Wang et al., 2025); but an SSM offers something a generic weight matrix does not, namely a *canonical* coordinate system for its own state. Balanced truncation (Moore, 1981; Pernebo & Silverman, 1982; Antoulas, 2005) puts the state in coordinates where the controllability and observability Gramians agree, $W_c = W_o = \mathrm{diag}(\sigma)$, and deletes the smallest σ_i — a principled choice of *which* states to remove, with a classical $\mathcal{H}_\infty$ guarantee (Enns, 1984; Glover, 1984). These Gramians measure, respectively, how strongly each state direction is excited by white-noise inputs over an infinite horizon and how much output energy that direction can produce, under the Euclidean norm that weights all output channels equally.

In a deep SSM, however, systems are stacked through nonlinearities, so the input of an intermediate system is not chosen by a designer: it is whatever the upstream layers produce on the deployment data, over a finite horizon, and the output error of that system matters only through its effect on the final loss. Existing extensions of balanced truncation adapt the Gramians to finite horizons, measured trajectories, or frequency weights, but can lose stability and/or retain worst-case-over-input guarantees. In this paper, we introduce an input-aware form of balanced truncation that weights the state according to the directions excited by deployment inputs over the relevant horizon, allows a static metric on the output, and preserves stability of the reduced system. Classical balanced truncation is recovered as a special case. Importantly, the reduction admits an exact identity for its finite-horizon output error, valid for any input process, and the resulting bound is minimized when the state weight is the covariance induced by the deployment inputs. We apply this reduction throughout a deep SSM to obtain POISE: each system is reduced under the inputs it actually receives, and its error at every retained order is used to allocate a single global state budget across the network. On S4D, SaShiMi, S5, and Mamba-2, this substantially improves over prior post-training state-reduction methods. Most existing methods either delete states in the trained coordinates (using scores derived

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.

from the parameters or, as in GHOST (Menezes & Kyrillidis, 2026), from calibration inputs) or change coordinates using classical balanced truncation, which does not adapt to the deployment inputs. POISE uses calibration inputs both to choose the state coordinates and to allocate states across the network, while requiring neither labels nor retraining.

Our contributions are as follows:

(a) Input-aware balanced truncation with finite-horizon guarantees. We introduce a form of balanced truncation that adapts the state coordinates to the input distribution encountered at deployment while preserving stability. For any input process and finite horizon, we derive an exact identity for the mean-square output error of the reduced system, together with a computable upper bound (Theorem 1). This bound is minimized when the reduction is weighted by the state covariance induced by the deployment inputs.

(b) From local reduction error to network-level loss. For a deep SSM, we show how the errors made by reducing individual systems combine in the final network loss. In particular, the loss from reducing one system at a time is determined by that system's own output error under a network-dependent metric (Proposition 2). We then show how to estimate this error from a few passes on the uncompressed network, either exactly by filtering the discarded coordinates or through the spectral bound derived for a single system.

(c) POISE: post-training reduction under a global state budget. Combining these results gives POISE (Algorithm 1), a label-free, post-training method that reduces every system under the inputs it actually receives and allocates a single state budget across the network. The allocation is obtained by one sort when using the spectral bound, or by dynamic programming when using the exact error curves, with no retraining. We evaluate POISE on S4D, SaShiMi, S5, and Mamba-2, including ablations on truncation, allocation, calibration, and computational cost.

2 Related work

Balanced truncation and its generalizations. Balanced truncation, introduced by Moore (1981), is a classical model reduction algorithm that preserves stability (Pernebo & Silverman, 1982) with error bounds in the $\mathcal{H}_\infty$ norm (Enns, 1984; Glover, 1984) and in the $\mathcal{H}_2$ norm (Antoulas, 2005; Chahlaoui, 2012). Weighting the Gramians adapts these guarantees to finite horizons (Gawronski & Juang, 1990; Redmann & Kürschner, 2018; Kürschner, 2018; Pontes Duff & Kürschner, 2021) or to frequency weighting (Enns, 1984; Wang et al., 1999). In particular, reweighting has had successful applications in subspace identification methods (Ljung, 1999) although guarantees are typically asymptotic in the sample size and hold for specific classes of inputs (Bauer, 1998). POISE builds on a similar weighted balanced truncation procedure. But it admits an exact reduced output-error identity with an a priori upper bound for any horizon and any input process, and it preserves the stability of a stable system, which these weighted variants can lose (Van Gestel et al., 2001).

Order reduction for deep SSMs. Several methods transfer model reduction to the post-training compression of deep SSMs. *Modal truncation*, as in LAST (Gwak et al., 2024), AIRE-Prune (Padhy et al., 2026) and GHOST (Menezes & Kyrillidis, 2026), considers diagonal state matrices only and prunes modes directly, with no change of basis. Their pruning score is designed either from the trained parameters alone or from calibration data, but the compressed model remains in the original coordinates. *Balanced truncation* (Bittner et al., 2025) has been applied as well but without the input law and with an expensive pruning criterion requiring labels. POISE applies both an input-aware balancing and an input-aware scoring from a small set of calibration inputs, and the compressed model can be re-diagonalized. CoBRAS (Otto et al., 2023), designed for nonlinear systems, also balances a state covariance estimated from data, and on an LTI system the two constructions coincide up to the trajectory horizon. For LTI systems, our analysis adds stability guarantees and gives an exact finite-horizon output-error identity; we further address how to allocate a global state budget across systems. Besides post-training compression, other regimes have been considered, such as in-training reduction, where CompreSSM (Chahine et al., 2026) applies a Hankel-energy threshold per system, with no global budget. Appendix C surveys the remaining SSM compression methods.

Allocating a budget across layers. Compressing a network under one budget means choosing the compression of every layer jointly. Post-training compression methods make this tractable in

two steps. The objective is a sum of per-layer errors (LeCun et al., 1990), each the loss the network incurs when that layer alone is compressed, measured at a few candidate levels (Cai et al., 2020) or approximated by a Hessian trace (Dong et al., 2019). Section 4 keeps this two-step structure for deep SSMs: balanced truncation gives every system its whole error curve from one balancing (Theorem 1), so the one-at-a-time loss does not have to be measured level by level across thousands of systems, and a single sort or dynamic programming allocates the budget.

3 Input-aware reduction of a single SSM

A deep SSM contains many linear state-space systems embedded between nonlinear layers. We begin by isolating one such system and asking how to reduce its state dimension. Section 4 will then apply the construction to all systems in the network and allocate their dimensions jointly.

Consider a stable discrete-time linear system $\mathcal{S}:\ x_{k+1} = Ax_k + Bu_k,\ y_k = Cx_k,\ x_0 = 0$ of order n, with m inputs and p outputs, with (A, B) controllable and (A, C) observable. We seek a system $\mathcal{S}_r$ of order $r < n$ that approximates $\mathcal{S}$ on the inputs that this block encounters inside the network. There are two reasons why ordinary balanced truncation is not quite the criterion we want. First, not all state directions are equally excited by the inputs occurring in a trained network. A reduction should preserve directions that are important under this *input distribution*, rather than under a generic excitation. Second, not all output errors need be equally important to the remainder of the network. Here we use two general weights: a matrix $P \succ 0$ on the state directions excited by the input, and an output filter Φ, represented by the metric $M := \Phi^\top \Phi \succ 0$. Thus the reduction is judged by the filtered error $\|\Phi(y_k - \hat{y}_k)\|_2^2 = (y_k - \hat{y}_k)^\top M(y_k - \hat{y}_k)$. For now P and M are arbitrary. The theorem below will identify the optimal input weight, while Section 4 will specify how these weights are chosen for a deep SSM.

3.1 Weighted Balanced Truncation

The output weight M induces the weighted observability Gramian $Q_M := \sum_{t\geq 0}(A^t)^\top C^\top M C A^t$, also defined as the unique solution to Stein's equation $A^\top Q_M A - Q_M + C^\top M C = 0$. The usual observability Gramian is Q_I. Since (A, C) is observable and $M \succ 0$, $Q_M \succeq \lambda_{\min}(M) Q_I \succ 0$. We balance the pair (P, Q_M) by choosing an invertible S such that

$$S^{-1} P S^{-\top} = S^\top Q_M S = \Lambda := \operatorname{diag}(\sigma_1, \ldots, \sigma_n), \quad \text{with } \sigma_1 \geq \cdots \geq \sigma_n > 0, \tag{1}$$

where $\sigma_i = \sigma_i(M, P) := \sqrt{\lambda_i(PQ_M)}$. For example, if $P = RR^\top$, $Q_M = LL^\top$, and $L^\top R = U\Lambda V^\top$, one may take $S = RV\Lambda^{-1/2}$.

We henceforth work in these balanced coordinates. With a slight abuse of notation, we replace $x \leftarrow S^{-1}x$, $A \leftarrow S^{-1}AS$, $B \leftarrow S^{-1}B$, and $C \leftarrow CS$. We partition the transformed matrices as

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \qquad B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \qquad C = \begin{bmatrix} C_1 & C_2 \end{bmatrix}, \qquad \Lambda = \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix},$$

where the leading blocks have dimension r. Truncating the trailing balanced coordinates then yields the order-r reduced system

$$\mathcal{S}_r: \qquad \hat{x}_{k+1} = A_{11}\hat{x}_k + B_1 u_k, \qquad \hat{y}_k = C_1 \hat{x}_k, \qquad \hat{x}_0 = 0. \tag{2}$$

Algorithm 2 summarizes the construction. Importantly, balancing is independent of r: once the transformation has been computed, every reduced order is obtained by taking a different leading block.

3.2 Exact reduction error

We now ask how the choice of P should depend on the inputs seen by the system. Let $\mathbf{u}_T = (u_k)_{k<T}$ denote any random input process and define its induced finite-horizon state covariance $P_{\mathbf{u}_T} := \frac{1}{T}\sum_{k<T}\mathbb{E}[x_k x_k^\top]$. We measure the output error by $\mathbb{E}\|\mathbf{y}_T - \hat{\mathbf{y}}_T\|_M^2$, where $\|\mathbf{y}_T - \hat{\mathbf{y}}_T\|_M^2 := \frac{1}{T}\sum_{k<T}(y_k - \hat{y}_k)^\top M(y_k - \hat{y}_k)$ is the time-averaged M-norm of one trajectory. The following theorem gives both an exact expression for this error and the choice of P that minimizes its resulting

upper bound. To state the results, we introduce $\Pi := S^{-1}P_{\mathbf{u}_T}S^{-\top}$, $\nu_k := x_k^{(2)}$ (the discarded part of the state), $\xi_k := x_k^{(1)} - \hat{x}_k$ (the state truncation error), and $w_k := A_{21}\xi_k + A_{22}\nu_k$.

Theorem 1 (Input-weighted balanced truncation)**.** *Let $P, M \succ 0$, and construct $\mathcal{S}_r$ using the WBT algorithm (Algorithm 2). For any horizon T and any input process $\mathbf{u}_T$:*

(i) *The reduced system is Schur stable: $\rho(A_{11}) < 1$.*

(ii) *The output error satisfies*

$$\mathbb{E}\|\mathbf{y}_T - \hat{\mathbf{y}}_T\|_M^2 = \operatorname{tr}(\Lambda_2\Pi_{22}) - \frac{1}{T}\sum_{k<T}\mathbb{E}[w_k^\top\Lambda_2 w_k] - \frac{1}{T}\mathbb{E}[\xi_T^\top\Lambda_1\xi_T]. \tag{3}$$

In particular, $\mathbb{E}\|\mathbf{y}_T - \hat{\mathbf{y}}_T\|_M^2 \le \varepsilon(r, M, P)^2 := \operatorname{tr}(\Lambda_2\Pi_{22})$.

(iii) *If $P_{\mathbf{u}_T} \succ 0$, the bound is minimized by balancing against the state covariance generated by the input itself:*

$$\min_{P\succ 0}\varepsilon(r, M, P)^2 = \varepsilon(r, M, P_{\mathbf{u}_T})^2 = \sum_{i>r}\sigma_i(M, P_{\mathbf{u}_T})^2. \tag{4}$$

Theorem 1 gives the motivation for input-aware balancing. For an arbitrary design matrix P, the discarded weighted singular values need not describe the error under the inputs of interest. When $P = P_{\mathbf{u}_T}$, however, the bound is smallest among all choices of P, and reduces to the spectral tail $\sum_{i>r}\sigma_i^2$. Thus the same trained system may admit different reductions under different input distributions.

Ordinary balanced truncation is recovered by taking $P = W_c$ (where the controllability Gramian W_c solves $AW_cA^\top - W_c + BB^\top = 0$) and $M = I$. Theorem 1 therefore also gives a distribution-dependent view of the classical method. In particular, part *(ii)* gives its exact finite-horizon Euclidean output error under the input process on which the system is actually run, whereas the classical $\mathcal{H}_\infty$ theory provides a worst-case bound over inputs of bounded energy. Part *(iii)* further shows that the classical choice $P = W_c$ generally does not minimize this finite-horizon bound. Instead, part *(iii)* singles out $P = P_{\mathbf{u}_T}$, the state covariance induced by the deployment input process. This gives a data-dependent analogue of weighted balanced truncation. Classical extensions adapt the reduction using input spectra, frequency bands, or time windows specified in advance by the designer (Enns, 1984; Gawronski & Juang, 1990; Wang et al., 1999); here the weighting is instead induced by the inputs that the system actually encounters. This is possible because Theorem 1 applies to any $P \succ 0$, allowing us to choose P from deployment data rather than fixing it a priori.

3.3 Estimating the input weight

The bound-minimizing matrix $P_{\mathbf{u}_T}$ does not require labels. Given N calibration sequences $\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(N)}$ drawn from the input distribution, run the original system and form $\widehat{P} := \frac{1}{NT}\sum_{m=1}^{N}\sum_{k<T} x_k^{(m)}(x_k^{(m)})^\top$, an unbiased estimate of $P_{\mathbf{u}_T}$. Estimating Gramians from trajectories is classical (Lall et al., 2002; Himpe, 2018; Kawano & Scherpen, 2021), and the next result quantifies the price of replacing the population covariance by this empirical one.

Proposition 1 (Finite-sample certificate)**.** *Suppose $P_{\mathbf{u}_T} \succ 0$ and assume that, for some $\eta < 1$, $(1-\eta)P_{\mathbf{u}_T} \preceq \widehat{P} \preceq (1+\eta)P_{\mathbf{u}_T}$. Then $\widehat{P} \succ 0$, and for every r,*

$$\varepsilon(r, M, \widehat{P})^2 \le \frac{1}{1-\eta}\sum_{i>r}\sigma_i(M, \widehat{P})^2, \qquad \varepsilon(r, M, \widehat{P})^2 \le \frac{1+\eta}{1-\eta}\,\varepsilon(r, M, P_{\mathbf{u}_T})^2. \tag{5}$$

The first inequality certifies the reduction built from $\widehat{P}$ using only its empirical weighted singular values and η, which Appendix A.2 measures. The second compares this certificate with the unavailable population-optimal weighting. Proposition 3 in Appendix D.3 gives finite sample bounds on η.

4 FROM A SINGLE SSM TO A DEEP SSM

Consider a network $F = \omega \circ \beta_L \circ \cdots \circ \beta_1$ of L blocks $\beta_\ell = \varphi_\ell \circ W_\ell \circ \Sigma_\ell$, where Σ_ℓ is a layer of SSMs acting in parallel, W_ℓ a linear mixing, and φ_ℓ a pointwise map inside a residual connection; ω is the final readout. We index the SSMs of all layers by $j = 1, \dots, J$, with orders n_j: a layer holds one SSM per channel on S4D and SaShiMi, and a single one on S5 and Mamba-2. For classification, F outputs logits and $\mathrm{softmax} \circ F$ gives the corresponding class probabilities. An *allocation* $\mathbf{r} = (r_1, \dots, r_J)$ specifies their reduced orders against the full order vector $\mathbf{n} = (n_1, \dots, n_J)$.

Let $\mathcal{D}$ be the deployment distribution of the network inputs $\mathbf{u}_T$. Given weights $\mathcal{P} = (P_j)_{j=1}^J$, $\mathcal{M} = (M_j)_{j=1}^J$, Algorithm 2 applied to each system produces the compressed network $\hat{F}(\,\cdot\,; \mathbf{r}, \mathcal{P}, \mathcal{M})$. A priori, any $P_j \succ 0$ and $M_j \succ 0$, which can both depend on $\mathcal{D}$ but also on $\mathbf{r}$, make it well defined. We measure fidelity to the original network, so no labels are required. For a convex C^2 function ψ, define the Bregman divergence $B_\psi(z', z) := \psi(z') - \psi(z) - \nabla\psi(z)^\top(z' - z)$ and the compression loss $\mathcal{L}_\psi(\mathbf{r}, \mathcal{P}, \mathcal{M}) := \mathbb{E}_{\mathbf{u}_T \sim \mathcal{D}} B_\psi\Big(\hat{F}(\mathbf{u}_T; \mathbf{r}, \mathcal{P}, \mathcal{M}), F(\mathbf{u}_T)\Big)$.

Ideally, under a budget of R states, we would solve $\min_{\mathbf{r}, \mathcal{P}, \mathcal{M}} \mathcal{L}_\psi(\mathbf{r}, \mathcal{P}, \mathcal{M})$ s.t. $\sum_{j=1}^J r_j \le R$, and $0 \le r_j \le n_j$. This problem is prohibitively expensive. Even with $\mathcal{P}$ and $\mathcal{M}$ fixed, evaluating every allocation can require up to $\prod_{j=1}^J (n_j + 1)$ compressed networks. Optimizing the weights as well only enlarges the search space. Our goal is therefore to construct a surrogate that (i) uses the single-SSM guarantees of Section 3, and (ii) can be evaluated with only a small number of network passes.

4.1 EVALUATING AND SIMPLIFYING THE NETWORK LOSS

We first express the network loss in terms of errors made by the individual SSMs. Let $\mathbf{u}_{j,T}$ denote the input received by system j in the *uncompressed* network (in particular, $\mathbf{u}_{1,T} = \mathbf{u}_T$), and define its teacher-forced error $\boldsymbol{\delta}_j := \mathbf{y}_{j,T}(\mathbf{u}_{j,T}) - \hat{\mathbf{y}}_{j,T}(\mathbf{u}_{j,T}; P_j, M_j)$, where $\hat{\mathbf{y}}_{j,T}(\mathbf{u}_{j,T}, P_j, M_j)$ is the output of system j, compressed with (P_j, M_j), on the input $\mathbf{u}_{j,T}$. Thus $\boldsymbol{\delta}_j$ isolates the local effect of reducing system j. These are the errors controlled by Theorem 1.

Define $\mathbf{n}_{-j}(r) := (n_1, \dots, n_{j-1}, r, n_{j+1}, \dots, n_J)$ the allocation in which only system j is reduced. And let $\langle a, b \rangle_\Gamma := \frac{1}{T}\langle a, \Gamma b \rangle$.

Proposition 2. *For every $(\mathbf{r}, \mathcal{P}, \mathcal{M})$ and every $\mathbf{u}_T$, there exist linear maps Γ_{jk}, forming a symmetric positive semidefinite block operator $(\Gamma_{jk})_{j,k} \succeq 0$, such that the following hold.*
(i) The network loss is a quadratic form in the teacher-forced errors:

$$\mathcal{L}_\psi(\mathbf{r}, \mathcal{P}, \mathcal{M}) = \frac{1}{2} \sum_{j=1}^J \sum_{k=1}^J \mathbb{E}_{\mathcal{D}} \big\langle \boldsymbol{\delta}_j, \boldsymbol{\delta}_k \big\rangle_{\Gamma_{jk}}. \tag{6}$$

The operators Γ_{jk} depend on ψ, $\mathbf{u}_T$, and the full compressed network $(\mathbf{r}, \mathcal{P}, \mathcal{M})$.
(ii) If only system j is reduced, then the corresponding one-at-a-time loss is

$$\mathcal{L}_\psi\big(\mathbf{n}_{-j}(r_j), \mathcal{P}, \mathcal{M}\big) = \frac{1}{2} \mathbb{E}_{\mathcal{D}} \|\boldsymbol{\delta}_j\|^2_{\Gamma_j}, \qquad \Gamma_j := \Gamma_{jj}, \tag{7}$$

where Γ_j is evaluated at $\mathbf{n}_{-j}(r_j)$.
In addition, if ψ is the log-sum-exp of the logits, $\psi(z) := \log \sum_c e^{z_c}$ for z the output of the network, the one-at-a-time loss is a Kullback–Leibler divergence,

$$\mathcal{L}_\psi\big(\mathbf{n}_{-j}(r_j), \mathcal{P}, \mathcal{M}\big) = \mathbb{E}_{\mathcal{D}}\, \mathrm{KL}\big(p(\mathbf{u}_T) \,\|\, \hat{p}_{-j}(\mathbf{u}_T)\big), \tag{8}$$

where $p := \mathrm{softmax} \circ F$ and $\hat{p}_{-j} := \mathrm{softmax} \circ \hat{F}\big(\,\cdot\,; \mathbf{n}_{-j}(r_j), \mathcal{P}, \mathcal{M}\big)$ is the output of the network with only system j compressed.

Proposition 2 separates local approximation from downstream sensitivity. The error $\boldsymbol{\delta}_j$ measures the perturbation created locally by reducing system j. The operator Γ_{jk} measures how the perturbations from systems j and k propagate through the remaining network and interact in the final loss. Thus the $\boldsymbol{\delta}_j$ are accessible from teacher-forced SSM reductions, whereas the Γ_{jk} encode the generally

complicated downstream network. Part *(i)* is an exact characterization, but it is not directly useful computationally: evaluating its right-hand side requires the Γ_{jk}. Part *(ii)* alleviates this problem for a one-at-a-time reduction. This motivates the additive surrogate

$$\min_{\mathbf{r}} \sum_{j=1}^{J} \mathcal{L}_\psi\big(\mathbf{n}_{-j}(r_j), \mathcal{P}, \mathcal{M}\big) \quad \text{s.t.} \quad \sum_{j=1}^{J} r_j \le R, \qquad 0 \le r_j \le n_j. \tag{9}$$

This one-at-a-time criterion has been used for post-training pruning and quantization of deep networks (Li et al., 2017; Cai et al., 2020; Yuan et al., 2023). For the log-sum-exp, each term can be measured directly by running the network with only system j compressed, without computing Γ_j. Moreover, if (P_j, M_j) depend only on r_j, the jth term depends only on r_j, so (9) is separable and can be minimized exactly by dynamic programming in $O(JR \max_j n_j)$ operations, as in Cai et al. (2020) for bit-width allocation (Appendix D.5).

For SSMs, however, evaluating this Γ curve, $\mathbb{E}_{\mathcal{D}}\|\boldsymbol{\delta}_j\|^2_{\Gamma_j}$ at every order, requires $\sum_j n_j$ compressed-network passes, which can be prohibitive when each system admits hundreds of possible state dimensions. We therefore seek a cheaper approximation to $\mathbb{E}_{\mathcal{D}}\|\boldsymbol{\delta}_j\|^2_{\Gamma_j}$ that exploits the single-SSM results of Section 3, provided we choose $P_j = P_{\mathbf{u}_{j,T}}$ as prescribed by Theorem 1.

4.2 Local errors at any order and network probes

Assume that $P_j = P_{\mathbf{u}_{j,T}}$ for every j and that M_j depends only on r_j. In the one-at-a-time setting of (9), $\mathbb{E}_{\mathcal{D}}\|\boldsymbol{\delta}_j\|^2_{\Gamma_j}$ is then a function of r_j alone. We replace this local error by $\mathbb{E}_{\mathcal{D}}\|\boldsymbol{\delta}_j\|^2_{M_j}$, which can be estimated efficiently using the single-SSM results of Section 3. We first explain how to construct this local error curve over all orders r_j, and then correct for the mismatch between the Γ_j- and M_j-weighted errors when forming the objective in (9).

Estimating $\mathbb{E}_{\mathcal{D}}\|\boldsymbol{\delta}_j\|^2_{M_j}$. There are two ways to estimate $\mathbb{E}_{\mathcal{D}}\|\boldsymbol{\delta}_j\|^2_{M_j}$ at any order: an exact one, at the cost of a filter, and the cheap spectral certificate of Theorem 1.

I. Exact. Using the notation of Section 3 for compressed SSM Σ_j, we can show (refer to the proof of Theorem 1) that the error at time k is $\boldsymbol{\delta}_{j,k} := y_{j,k} - \hat{y}_{j,k} = C_1\xi_k + C_2\nu_k$ where $\xi_0 = 0$, $\xi_{k+1} = A_{11}\xi_k + A_{12}\nu_k$. Hence $\boldsymbol{\delta}_j$ can be computed directly, at any order r_j, by applying a filter:

$$\boldsymbol{\delta}_j = H_{r_j} \star \nu, \qquad H_{r_j}(z) := C_2 + C_1(zI - A_{11})^{-1}A_{12}, \tag{10}$$

where $\star$ denotes convolution, and ν are the coordinates of the state that the truncation deletes (§3.2). One forward pass of the uncompressed network records the states, hence ν at every order, and filtering them gives $E_j(r_j) := \mathbb{E}\|\boldsymbol{\delta}_j\|^2_{M_j}$ for all r_j without running any compressed network.

II. Spectral certificate. Theorem 1 gives $\mathbb{E}\|\boldsymbol{\delta}_j\|^2_{M_j} \le E_j(r_j) := \sum_{i>r_j} \sigma_{j,i}(M_j, P_j)^2$. Thus one balancing produces the complete curve. With the empirical covariance $\widehat{P}_j$, Proposition 1 gives the corresponding finite-sample correction.

Handling the mismatch via network probes. There is a mismatch between $E_j(r_j)$ and $\mathbb{E}_{\mathcal{D}}\|\boldsymbol{\delta}_j\|^2_{\Gamma_j}$, which defines the objective in (9). To account for this mismatch, we replace the objective by a weighted sum of the $E_j(r_j)$: $\phi_c(\mathbf{r}) := \sum_{j=1}^{J} c_j E_j(r_j)$. Ideally, the weights would be chosen as $c_j(r_j) = \frac{\mathbb{E}_{\mathcal{D}}\|\boldsymbol{\delta}_j\|^2_{\Gamma_j}}{E_j(r_j)}$. Computing this ratio at every order, however, would require the same number of network passes as (9). We therefore approximate it by a quantity independent of the allocation. From here on, and in Algorithm 1, ψ is the log-sum-exp of the logits, so that each one-at-a-time term is the KL divergence of (8). The rate is obtained at a fixed *probe* allocation $\mathbf{r}^\pi = (r_j^\pi)_{j=1}^J$ that retains a fraction π of the states of every system: $r_j^\pi := \mathrm{round}(\pi n_j)$. We then set

$$c_j^\pi := c_j(r_j^\pi) = \frac{2\,\mathbb{E}_{\mathcal{D}}\,\mathrm{KL}\big(p(\mathbf{u}_T)\,\|\,\hat{p}_{-j}(\mathbf{u}_T)\big)}{E_j(r_j^\pi)}, \quad c^\pi := (c_j^\pi)_{j=1}^J$$

with $\hat{p}_{-j}$ taken at $\mathbf{n}_{-j}(r_j^\pi)$, the second form by Proposition 2. Computing each c_j^π requires only one forward pass. Thus, the network probes require J passes in total, rather than $\sum_j n_j$, and minimizing $\phi_{c^\pi}(\mathbf{r})$ over the allocation becomes computationally feasible.

Algorithm 1 POISE: input-weighted balanced truncation under one budget

Require: network F with J systems of orders $\mathbf{n}$; calibration inputs $u^{(1)}, \dots, u^{(N)}$, the first $N_p \le N$ of which serve the probes; budget R; probe allocation $\mathbf{r}^\pi$; variant $\in \{\text{SPECTRAL}, \text{EXACT}\}$.

1: one forward pass of F on $u^{(1)}, \dots, u^{(N)}$: record every system's state trajectory $x_{j,k}^{(m)}$ and the teacher output $p^{(m)} \leftarrow \operatorname{softmax}\big(F(u^{(m)})\big)$

2: $\widehat{P}_j \leftarrow \dfrac{1}{NT} \sum_{m \le N} \sum_{k<T} x_{j,k}^{(m)} \big(x_{j,k}^{(m)}\big)^\top$ for every j with optional ridge regularization

3: $(\bar{A}_j, \bar{B}_j, \bar{C}_j, \sigma_j) \leftarrow \textsc{Balance}\big(\Sigma_j, \widehat{P}_j, M = I\big)$ for every j

4: $\hat{F}(.; \mathbf{r}) \leftarrow$ the network F with $\textsc{Truncate}(\bar{A}_j, \bar{B}_j, \bar{C}_j; r_j)$ in place of Σ_j for every j

5: $E_j(r) \leftarrow \sum_{i>r} \sigma_{j,i}^2$ (SPECTRAL) or $\dfrac{1}{NT} \sum_{m \le N} \sum_{k<T} \big\|(H_r \star \nu_j^{(m)})_k\big\|^2$ (EXACT)

6: $c_j^\pi \leftarrow \dfrac{2}{N_p} \sum_{m \le N_p} \mathrm{KL}\big(p^{(m)} \,\|\, \operatorname{softmax} \hat{F}(u^{(m)}; \mathbf{n}_{-j}(r_j^\pi))\big) \big/ E_j(r_j^\pi)$ ▷ one pass per system

7: $\mathbf{r}^\star \leftarrow \arg\min \phi_{c^\pi}$ under the budget: a sort (SPECTRAL) or dynamic programming (EXACT)

8: deploy $\hat{F}(\,\cdot\,; \mathbf{r}^\star)$ with each system re-diagonalized

4.3 Algorithm

We propose POISE (Algorithm 1), which combines these ingredients with $M_j = I$; Appendix E discusses this choice. A single forward pass on the calibration inputs records the state trajectories, from which we obtain every $\widehat{P}_j$ and, in the exact variant, every error curve; balancing yields the spectral curves. The probes require J additional passes (see Table 4 for a FLOPs analysis). The exact variant also filters the recorded states once per system, to measure its error curve. The final allocation is then obtained by sorting in the spectral variant, or by dynamic programming in the exact variant (Appendix D.5).

5 Experiments

Setup. We compress all models post hoc, without retraining. We consider a four-layer S4D classifier on sequential MNIST (LeCun et al., 1998) (ten seeds, 256 single-input single-output systems), the published 24-layer SaShiMi checkpoint on SC09 audio (Warden, 2018), S5 on LRA-Text (Tay et al., 2021) (three seeds, one system per layer) and Mamba-2 130M and 1.3B on WikiText-2 (Merity et al., 2017) under GHOST's protocol (three calibration draws), see Appendix A for details. A reduction method makes two choices: *truncation*, which selects the state subspace retained within each system, and *allocation*, which selects the orders r_j subject to $\sum_j r_j \le R$. The published baselines differ from POISE in both choices. We compare the complete methods in §5.1 and isolate truncation and allocation separately in Appendix B. Tables give the mean and the standard deviation over seeds or draws. The probe keeps $\pi = 0.15$ of each system's states on every model; Appendix B.8 varies π and tests a one-step correction of the probe.

Baselines. We compare with the published modal rules LAST and AIRE-Prune, which score modes from the parameters alone, with Global $\mathcal{H}_\infty$, the pooled score LAST builds on, and with GHOST. We add classical balanced truncation ($P = W_c$ in Theorem 1). This baseline is our own implementation. We allocate its states using the same pooled sort as POISE with $c \equiv 1$, but on the classical unweighted spectral tail. Classical BT therefore differs from POISE with $c \equiv 1$ only in the input-side weight: $P = W_c$ for classical BT and $P = \widehat{P}_j$ for POISE. Appendix C gives the implementation details.

5.1 Complete-method comparison

Table 1 reports all four variants of POISE. With c^π, both curves beat every baseline on S4D, SaShiMi and Mamba-2, and the spectral curve, our default, is best on S5 and ahead of every baseline at every tabulated budget (Appendix B.1). On S4D, the default stays 0.05 points below the uncompressed model, whereas GHOST loses 0.43 points and the parameter-only rules lose 19–51; on SaShiMi it loses 2×10^{-4} bits, four orders of magnitude less than any modal rule; on S5, it is 17.8 points above the best modal rule and 12.2 above classical BT; and on Mamba-2 1.3B, it adds 1.95 perplexity,

Table 1: The complete methods, each with its own allocation, one budget per model. Accuracy (%) on S4D and S5, excess test NLL (bits) over the uncompressed model on SaShiMi, WikiText word perplexity on Mamba-2. Budget grids in Appendix B.1.

rule	S4D 0.15	SaShiMi 0.15	S5, LRA-Text 0.03	Mamba-2, 1.3B 0.10
Global $\mathcal{H}_\infty$	80.17 ±14.3	+3.2928	51.33 ±2.27	—
LAST	79.32 ±15.8	+2.3356	50.41 ±0.71	—
AIRE-Prune	47.58 ±20.8	+3.0889	50.02 ±0.03	—
GHOST	98.35 ±0.30	+1.8511	51.67 ±1.44	25.44 ±0.52
classical BT	98.11 ±0.31	+0.0150	57.19 ±1.52	—
POISE, spectral, $c \equiv 1$	98.69 ±0.31	+0.00298	64.03 ±13.5	20.99 ±0.03
POISE, spectral, c^π, $\pi = 0.15$	98.73 ±0.29	+0.00023	**69.42** ±5.55	**15.12** ±0.04
POISE, exact, $c \equiv 1$	**98.75** ±0.24	+0.00113	59.38 ±5.37	20.92 ±0.04
POISE, exact, c^π, $\pi = 0.15$	**98.75** ±0.27	**+0.00010**	56.23 ±9.15	15.15 ±0.04
uncompressed	98.78	0	88.67	13.17

compared with 12.3 for GHOST. Table 2 reports the deployed state size and the time required by POISE for each allocation. On S4D and SaShiMi, the probe adds roughly half the cost of the corresponding $c \equiv 1$ variant. Every configuration costs less than one training epoch on S4D and SaShiMi and at most 8.0 epochs on S5. With the channels of each layer divided into four groups, the state memory falls by about 80% on S4D and SaShiMi.

Table 2: Cost of POISE: state per stream in the deployed form, and time from the loaded checkpoint to the allocation, in training epochs in parentheses. Details in Appendix A.3.

	budget	S4D 0.15	SaShiMi 0.15	S5, LRA-Text 0.03	Mamba-2 1.3B 0.10
state	original	65.5 kB	1.84 MB	9.22 kB	50.3 MB
	POISE	13.9 kB (−79%)	0.310 MB (−83%)	280 B (−97%)	5.11 MB (−90%)
time	spectral, $c \equiv 1$	3.3 s (0.14)	254 s (0.34)	54.2 s (2.1)	17 min (< 0.0005)
	spectral, c^π, $\pi = 0.15$	5.2 s (0.22)	418 s (0.56)	81.8 s (3.1)	48 min (< 0.0013)
	exact, $c \equiv 1$	3.5 s (0.15)	419 s (0.56)	181 s (7.0)	30 min (< 0.0008)
	exact, c^π, $\pi = 0.15$	5.4 s (0.22)	583 s (0.78)	209 s (8.0)	61 min (< 0.0016)

5.2 Main empirical findings

We summarize here the main empirical findings; the supporting experiments and ablations are reported in Appendix B due to space constraints.

(i) The truncation is where the modal rules fail. Under a uniform allocation, so that only the truncation differs, ours is best on every model: 12.1 points above GHOST on S4D, 19.7 above the best modal rule on S5 at budget 0.10, and 4,000× fewer bits lost on SaShiMi (Table 10). The coordinate system used for truncation matters more than whether the score uses data: classical unweighted BT already beats GHOST. The gain also carries over to Mamba-2, which lies outside our theory (Appendix A). GHOST is the only modal rule that transfers there, and ours beats it at equal orders, 18.60 versus 25.44 at budget 0.1.

(ii) The probed scale drives the allocation, and can improve the baselines too. Across the four POISE variants, spectral or exact curves with $c \equiv 1$ or c^π, the probed scale gives the best allocation for each curve on SaShiMi and Mamba-2, and the best or a tie on S4D; with the spectral curve it improves over the uniform allocation by 1.59 points, 4.5× fewer bits and 3.48 perplexity on S4D, SaShiMi and Mamba-2, respectively (Appendix B.3). Changing the scale moves 18–64% of the states between layers, while changing the curve moves at most 11% (Appendix B.5). Applied to LAST and AIRE-Prune at budget 0.15, the same scale adds 26.6 points to AIRE-Prune but lowers LAST by 3.9 on S4D, and both remain far behind our truncation, by at least 23.3 points on S4D and three orders of magnitude on SaShiMi (Appendix B.4).

(iii) POISE benefits from matching the deployment data. On SaShiMi, trained on speech, calibrating on the evaluation distribution, speech or piano, reduces the excess loss by factors of 2–8. Even when calibrated on the other distribution, POISE still loses over 20× fewer bits than classical BT and 3,000× fewer than LAST, neither of which uses data (Appendix B.6).

(iv) The gains persist across budgets. POISE with its default spectral curve and c^{π} stays ahead of every baseline at every budget we tabulate: on S4D, it comes within about a point of the uncompressed accuracy with 5% of the states, whereas GHOST needs 10–15% (Figure 2, Appendix B.1).

5.3 Memory–perplexity tradeoff on Mamba-2 1.3B

We now ask whether reducing the recurrent state translates into a useful end-to-end memory–accuracy tradeoff at inference. A decoded batch of N_{batch} streams uses $W + N_{\text{batch}}s$ memory, where W is the memory of the shared weights and s the memory of the recurrent state of one stream. On Mamba-2 1.3B, POISE reduces s from 50.3 MB to 25.2, 15.3, and 5.11 MB at budgets 0.5, 0.3, and 0.1, with perplexities 13.30, 13.51, and 15.12 versus 13.17 uncompressed. The resulting savings grow with batch size: at budget 0.3, for example, the total footprint falls from 5.9 to 3.67 GB at $N_{\text{batch}} = 64$ and from 54 to 18.4 GB at $N_{\text{batch}} = 1024$ (Figure 1a).

At equal footprint, POISE improves substantially over GHOST and over using a smaller dense Mamba-2 model (Figures 1b,c). At $N_{\text{batch}} = 64$, the reduced 1.3B model reaches perplexity 13.51 in 3.67 GB, compared with 14.97 in 3.98 GB for dense Mamba-2 780M. We also compare against Transformers with quantized KV caches (Liu et al., 2024). At 2048 tokens and $N_{\text{batch}} = 64$, Llama-3.2-1B with a four-bit cache reaches 12.94 perplexity in 4.00 GB, versus 13.51 in 3.67 GB for POISE. Unlike a KV cache, however, the recurrent-state footprint is independent of context length, so its relative memory advantage grows with longer contexts.

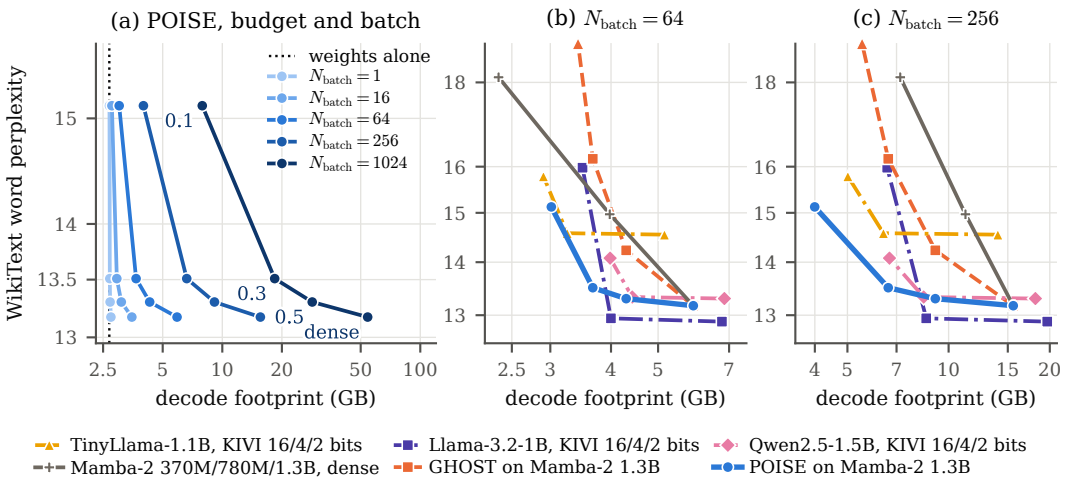


Figure 1: Decode footprint (weights plus N_{batch} streams of 2048 tokens, float16) against WikiText perplexity. (a) POISE on Mamba-2 1.3B at budgets 0.5, 0.3 and 0.1. (b), (c) The same at $N_{\text{batch}} = 64$ and 256, against GHOST, the dense Mamba-2 family and Transformers with a KIVI cache at 16, 4 and 2 bits; GHOST at 0.1 (25.4) is off the scale. Protocol in Appendix A.4.

6 Conclusion

In this work, we have proposed a novel compression framework that adapts to a small set of calibration inputs, requires no training/labels, and allocates reduced orders across systems by minimizing an output loss under a budget constraint. Theorem 1 certifies each reduced system of a stable LTI SSM. While the allocation across systems rests on a surrogate, as do layer-wise post-training methods such as GPTQ/OPTQ (Frantar et al., 2023) and SparseGPT (Frantar & Alistarh, 2023), our experimental results show that we can aggressively compress deep SSMs including Mamba-2 with the smallest incurred loss across baselines. Several future directions follow: extending theoretically to selective SSMs such as Mamba-2, and combining with quantization for hybrid-attention SSMs.

AI USE STATEMENT

In this work, we used generative AI tools for the following tasks: design or provide feedback on research methodology or experiments, implement methods. We have not used generative AI tools for the following tasks: generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, interpret results.

Additionally, we used generative AI tools for the following tasks: create or modify scientific figures, summarize or analyse existing literature, sourcing/searching for information, identify relevant literature. We have reviewed all AI-assisted work such as reading literature proposed to evaluate relevance, and we read/tested manually any generated code. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The supplementary material contains our code. Appendix A describes the models, data, calibration and budgets, and Appendix C how each baseline is implemented. Each theoretical result states its assumptions, and Appendix D proves every result of the main body.

REFERENCES

Steven Abreu, Jens E. Pedersen, Kade M. Heckel, and Alessandro Pierro. Q-S5: Towards quantized state space models. *arXiv preprint arXiv:2406.09477*, 2024.

Athanasios C. Antoulas. *Approximation of Large-Scale Dynamical Systems*. SIAM, 2005.

Dietmar Bauer. *Some Asymptotic Theory for the Estimation of Linear Systems Using Maximum Likelihood Methods or Subspace Algorithms*. PhD thesis, Technische Universität Wien, 1998.

Stella Biderman et al. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Matthias Bittner, Daniel Schnöll, Dominik Dallinger, Matthias Wess, and Axel Jantsch. Pruning state space models with model order reduction for efficient raw audio classification. In *2025 33rd European Signal Processing Conference (EUSIPCO)*, 2025.

Yaohui Cai, Zhewei Yao, Zhen Dong, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. ZeroQ: A novel zero shot quantization framework. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Makram Chahine, Philipp Nazari, Daniela Rus, and T. Konstantin Rusch. The curious case of in-training compression of state space models. In *International Conference on Learning Representations*, 2026.

Younes Chahlaoui. A posteriori error bounds for discrete balanced truncation. *Linear Algebra and its Applications*, 2012.

Hung-Yueh Chiang, Chi-Chih Chang, Natalia Frumkin, Kai-Chiang Wu, Mohamed S. Abdelfattah, and Diana Marculescu. Quamba2: A robust and scalable post-training quantization framework for selective state space models. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025a.

Hung-Yueh Chiang, Chi-Chih Chang, Natalia Frumkin, Kai-Chiang Wu, and Diana Marculescu. Quamba: A post-training quantization recipe for selective state space models. In *International Conference on Learning Representations*, 2025b.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Soham De et al. Griffin: Mixing gated linear recurrences with local attention for efficient language models. *arXiv preprint arXiv:2402.19427*, 2024.

Zhen Dong, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. HAWQ: Hessian aware quantization of neural networks with mixed-precision. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Dale F. Enns. Model reduction with balanced realizations: An error bound and a frequency weighted generalization. In *The 23rd IEEE Conference on Decision and Control*, 1984.

Haruka Ezoe and Kazuhiro Sato. Model compression method for S4 with diagonal state space layers using balanced truncation. *IEEE Access*, 2024.

Marco Forgione, Manas Mejari, and Dario Piga. Model order reduction of deep structured state-space models: A system-theoretic approach. In *2024 IEEE 63rd Conference on Decision and Control (CDC)*, 2024.

Elias Frantar and Dan Alistarh. SparseGPT: Massive language models can be accurately pruned in one-shot. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. OPTQ: Accurate quantization for generative pre-trained transformers. In *International Conference on Learning Representations*, 2023.

Wodek Gawronski and Jer-Nan Juang. Model reduction in limited time and frequency intervals. *International Journal of Systems Science*, 1990.

Keith Glover. All optimal Hankel-norm approximations of linear multivariable systems and their L^∞-error bounds. *International Journal of Control*, 1984.

Karan Goel, Albert Gu, Chris Donahue, and Christopher Ré. It's raw! Audio generation with state-space models. In *Proceedings of the 39th International Conference on Machine Learning*, 2022.

Aaron Grattafiori et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024.

Albert Gu, Karan Goel, Ankit Gupta, and Christopher Ré. On the parameterization and initialization of diagonal state space models. In *Advances in Neural Information Processing Systems*, 2022a.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022b.

Minseon Gwak, Seongrok Moon, Joohwan Ko, and PooGyeon Park. Layer-adaptive state pruning for deep state space models. In *Advances in Neural Information Processing Systems*, 2024.

Christian Himpe. emgr—the empirical Gramian framework. *Algorithms*, 2018.

Yu Kawano and Jacquelien M. A. Scherpen. Empirical differential Gramians for nonlinear model reduction. *Automatica*, 2021.

Patrick Kürschner. Balanced truncation model order reduction in limited time intervals for large systems. *Advances in Computational Mathematics*, 2018.

Sanjay Lall, Jerrold E. Marsden, and Sonja Glavaški. A subspace approach to balanced truncation for model reduction of nonlinear control systems. *International Journal of Robust and Nonlinear Control*, 2002.

Yann LeCun, John S. Denker, and Sara A. Solla. Optimal brain damage. In *Advances in Neural Information Processing Systems*, 1990.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient ConvNets. In *International Conference on Learning Representations*, 2017.

Yuhang Li et al. BRECQ: Pushing the limit of post-training quantization by block reconstruction. In *International Conference on Learning Representations*, 2021.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. KIVI: A tuning-free asymmetric 2bit quantization for KV cache. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Lennart Ljung. *System Identification: Theory for the User*. Prentice Hall, 2nd edition, 1999.

Michael Menezes and Anastasios Kyrillidis. GHOST: Unmasking phantom states in Mamba2 via grouped hidden-state output-aware selection & truncation. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017.

Bruce C. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, 1981.

Juan Pablo Muñoz, Jinjie Yuan, and Nilesh Jain. Mamba-Shedder: Post-transformer compression for efficient selective structured state space models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2025.

Markus Nagel, Rana Ali Amjad, Mart van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? Adaptive rounding for post-training quantization. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Antonio Orvieto, Samuel L. Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting recurrent neural networks for long sequences. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Samuel E. Otto, Alberto Padovan, and Clarence W. Rowley. Model reduction for nonlinear systems by balanced truncation of state and gradient covariance. *SIAM Journal on Scientific Computing*, 2023.

Apurba Prasad Padhy, Fernando Camacho, and Saibal Mukhopadhyay. AIRE-Prune: Asymptotic impulse-response energy for state pruning in state space models. In *International Conference on Learning Representations*, 2026.

Lars Pernebo and Leonard M. Silverman. Model reduction via balanced state space representations. *IEEE Transactions on Automatic Control*, 1982.

Igor Pontes Duff and Patrick Kürschner. Numerical computation and new output bounds for time-limited balanced truncation of discrete-time systems. *Linear Algebra and its Applications*, 2021.

Martin Redmann and Patrick Kürschner. An output error bound for time-limited balanced truncation. *Systems & Control Letters*, 2018.

Hiroki Sakamoto and Kazuhiro Sato. A deep state-space model compression method using upper bound on output error. *arXiv preprint arXiv:2510.14542*, 2025a.

Hiroki Sakamoto and Kazuhiro Sato. Compression method for deep diagonal state space model based on H^2 optimal reduction. *IEEE Control Systems Letters*, 2025b.

Paul Schwerdtner, Jules Berman, and Benjamin Peherstorfer. Hankel singular value regularization for highly compressible state space models. In *Advances in Neural Information Processing Systems*, 2025.

Ibne Farabi Shihab, Sanjeda Akter, and Anuj Sharma. Efficient unstructured pruning of Mamba state-space models for resource-constrained environments. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025.

Jimmy T. H. Smith, Andrew Warrington, and Scott W. Linderman. Simplified state space layers for sequence modeling. In *International Conference on Learning Representations*, 2023.

Yi Tay et al. Long Range Arena: A benchmark for efficient transformers. In *International Conference on Learning Representations*, 2021.

Joel A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 2012.

Tony Van Gestel, Bart De Moor, Brian D. O. Anderson, and Peter Van Overschee. On frequency weighted balanced truncation: A constructive counterexample to Enns' conjecture. In *2001 European Control Conference (ECC)*, 2001.

G. Wang, V. Sreeram, and W. Q. Liu. A new frequency-weighted balanced truncation method and an error bound. *IEEE Transactions on Automatic Control*, 1999.

Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. SVD-LLM: Truncation-aware singular value decomposition for large language model compression. In *International Conference on Learning Representations*, 2025.

Pete Warden. Speech Commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

An Yang et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Zhihang Yuan, Yuzhang Shang, Yue Song, Dawei Yang, Qiang Wu, Yan Yan, and Guangyu Sun. ASVD: Activation-aware singular value decomposition for compressing large language models. *arXiv preprint arXiv:2312.05821*, 2023.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. TinyLlama: An open-source small language model. *arXiv preprint arXiv:2401.02385*, 2024.

CONTENTS OF THE APPENDIX

A Experimental protocol **15**

A.1 The balanced truncation algorithm 17

A.2 Estimation error of the covariance 17

A.3 Cost 18

A.4 Protocol for quantized methods 19

B Ablations **21**

B.1 The complete methods at every budget 21

B.2 The truncation, at a fixed allocation 23

B.3 The allocation, at a fixed truncation 24

B.4 Applying the scaling rule to related work 24

B.5 The rank allocations 25

B.6 Calibrating on a different law 28

B.7 The probed criterion against the Γ curve 28

B.8 A correction for the probe 28

B.9 Calibration size 29

C Baselines **31**

D Proofs of the results in the main body **33**

D.1 Proof of Theorem 1 33

D.2 Proof of Proposition 1 35

D.3 Concentration of the plug-in covariance 35

D.4 Proof of Proposition 2 36

D.5 Dynamic programming for the one-at-a-time criterion 37

D.6 A single sort minimizes the weighted tail 37

E The choice of the output metric M_j **38**

E.1 A scalar metric leaves the truncation unchanged 38

E.2 Metric shapes on S5 38

A EXPERIMENTAL PROTOCOL

Models and data. *Four-layer S4D.* The model has four layers of 64 single-input single-output channels, with 64 states per channel ($J = 256$ systems and $16{,}384$ states in total). We train it on sequential MNIST using ten seeds and evaluate it on $10{,}000$ held-out sequences. The uncompressed accuracy is 98.78%.

SaShiMi. We use the published 24-layer SC09 checkpoint (Goel et al., 2022) with the authors' pipeline and evaluate it on all $4{,}107$ test clips. The uncompressed NLL is 1.9270 bits. There is one checkpoint and one calibration draw, so each SaShiMi result represents a single run.

S5. We train three checkpoints per LRA task with the authors' pipeline. These reproduce the published accuracies, with 88.7 against 89.31 on Text and 61.4 against 62.15 on ListOps. The pipeline keeps the epoch of best validation accuracy; LRA-Text has no validation split, so there the epoch is chosen on the test split, for every rule alike. Each system corresponds to a whole layer.

Mamba-2. We use the released 130M and 1.3B checkpoints evaluated by GHOST (Menezes & Kyrillidis, 2026), with 24 and 48 layers respectively. Each layer has $d_{\text{state}} = 128$ states shared across its heads and is treated as one system. Following GHOST, we evaluate WikiText word perplexity with lm-eval at $2{,}048$ tokens. The uncompressed values are 25.85 and 13.17, matching those reported by GHOST.

Calibration. Calibration uses only training inputs. We use the first 256 sequences on S4D and S5 and 320 clips on SaShiMi. For Mamba-2, we follow GHOST and use 128 WikiText-2 training sequences of $1{,}024$ tokens in each of three draws. The probes use 256, 64, 8 and 32 calibration inputs respectively, with $\pi = 0.15$. These choices are discussed in Appendices B.8 and B.9.

Budgets. A budget is a fraction of the usable states, defined below. Every rule of a table runs at the same state count (a retained complex mode counts two states), up to rounding, and on SaShiMi the other rules keep more (see Allocation), in the same run and on the same test split. Tables report mean $\pm$ standard deviation over seeds, or over calibration draws on Mamba-2. On Mamba-2, at budget b, GHOST keeps $k = d_{\text{state}} - \lfloor d_{\text{state}}(1-b) \rfloor$ states in every layer, and `POISE` is given the same total, Lk over the L layers, which it allocates across layers.

Allocation. The spectral variant allocates by the sort of Proposition 4 and the exact variant by dynamic programming (Appendix D.5), on every model, with no state reserved. On SaShiMi, 14 of the $3{,}584$ channels could not be stably balanced at any tolerance; they keep their original kernel, count no states and take no part in any allocation. Only classical BT and `POISE` decline channels, and Tables 2 and 5 count these channels in the memory. The other rules take their budget as a fraction of all $229{,}376$ states rather than of the usable ones, so at the tabulated budgets they keep more states than `POISE`, even with the declined channels counted.

Baselines. Appendix C defines every baseline and how we run it.

How each system is realized. The following implementation steps are required to apply Algorithm 1 successfully to the trained checkpoints.

1. *Computation of the observability factor.* On the four-layer model and on SaShiMi, Q_M solves the Stein equation of §3.1. On S5 we obtain its factor directly from a QR decomposition of the Krylov stack $K = [C; CA; \ldots; CA^{T-1}]$, which is the Stein series truncated at a horizon T. The factor avoids forming $Q_M = K^\top K$, which would square the condition number, and the ListOps checkpoints have modes on or outside the unit circle, where the infinite-horizon Gramian does not exist. On LRA-Text the truncated series matches the Stein solution to 1.9×10^{-5}; on ListOps we use $T = 4000$, about twice its longest sequence. On Mamba-2 the observability side is the measured moment $\widehat{Q}$ described below, since C_t depends on the input.
2. *Usable states.* Each system is first reduced to its numerically minimal realization by treating Hankel values below 10^{-10} of the largest value as zero. This leaves $16{,}261$–$16{,}355$ of $16{,}384$ states on the four-layer model and $214{,}772$ of $229{,}376$ on SaShiMi. Budgets are

fractions of these usable states. A returned value of $\rho(A_{11}) \geq 1$ can only result from a failed factorization. When this occurs, we increase the rank tolerance until $\rho(A_{11}) < 1$. This was necessary for 7 of the 256 channels in one SaShiMi layer.

3. *Singular covariances.* The calibration inputs do not excite every state direction, so $\widehat{P}_j$ can be singular, while Theorem 1 requires $P \succ 0$; a small ridge restores it. Its median condition number reaches 2×10^{10} on SaShiMi. On S4D and SaShiMi $\widehat{P}_j$ is factored by Cholesky after adding the smallest ridge $\epsilon \operatorname{tr}(\widehat{P}_j)/n_j\, I$, $\epsilon \in \{0, 10^{-12}, \ldots, 10^{-6}\}$, that returns positive Hankel values and $\rho(A_{11}) < 1$; a ridge was needed on 18% of the S4D systems and on no SaShiMi channel. On S5, $\widehat{P}_j$ is factored by its eigendecomposition, and the directions it does not excite, whose Hankel values are zero, are removed.

Mamba-2 layers. Within a head, a Mamba-2 layer updates its state as $h_t = a_t h_{t-1} + B_t x_t$ and reads it out as $y_t = C_t^\top h_t$, where the scalar a_t and the vectors $B_t, C_t \in \mathbb{R}^N$ depend on the input. Theorem 1 therefore does not apply. We instead perform balancing using measured moments in place of the Gramians, $\widehat{P} = \sum_t h_t h_t^\top$ for the state and $\widehat{Q} = \sum_t C_t C_t^\top$ for the output map, summed over calibration tokens, heads and head dimensions. Balancing $(\widehat{P}, \widehat{Q})$ as in Algorithm 2 gives S. Let $U \in \mathbb{R}^{N \times r}$ contain its first r columns, and let $V \in \mathbb{R}^{r \times N}$ contain the first r rows of S^{-1}. Then $VU = I_r$, and the product UV projects onto the r leading balanced coordinates.

The reduced state $z_t = V h_t$ satisfies $z_t = a_t z_{t-1} + (V B_t) x_t$, because a_t is a scalar, and the reduced output is $\hat{y}_t = (U^\top C_t)^\top z_t = C_t^\top U V h_t$. The reduced layer is therefore a Mamba-2 layer with r states. We implement it by replacing C_t with $(UV)^\top C_t$ in the original layer. *Choice of projection.* Because the projected recursion is exact, the error is only in the output, $y_t - \hat{y}_t = C_t^\top (I - UV) h_t$. We therefore seek a rank-r projection UV that minimizes $\sum_t \|C_t^\top (I - UV) h_t\|^2$. This sum depends on the joint moments of C_t and h_t. Replacing them by the product of the separate moments $\widehat{Q}$ and $\widehat{P}$ turns it into $\operatorname{tr}\big((I - UV)^\top \widehat{Q} (I - UV) \widehat{P}\big) = \|L^\top (I - UV) R\|_F^2$, with $\widehat{P} = RR^\top$ and $\widehat{Q} = LL^\top$, whose minimizer over rank-r projections is the balanced truncation of $(\widehat{P}, \widehat{Q})$. The replacement is exact when C_t and h_t are independent. In a Mamba-2 layer both depend on the input, so this projection minimizes an approximation of the output error rather than the error itself. Section 5.2 compares it with GHOST's diagonal mask projections at the same orders, where it gives the lower perplexity.

The exact curve. Since $A_t = a_t I$, the balanced partition has $A_{12,t} = 0$. The truncation error is therefore given directly by the readout error, without an additional error-state recurrence. At order r, the exact curve for layer j is

$$E_j(r) = \sum_t \big(C_t^\top (I - U_r V_r) h_t\big)^2,$$

where the sum includes calibration tokens, heads and head dimensions, and $U_r V_r$ projects onto the first r balanced coordinates.

Let S be the full balancing transformation and define $q_t = (S^\top C_t) \odot (S^{-1} h_t)$. The discarded coordinates contribute $\sum_{i>r} q_{t,i}$ to the readout error, so

$$E_j(r) = \sum_{i>r} \sum_{k>r} K_{ik}, \qquad K = \sum_t q_t q_t^\top.$$

Accumulating the single $N \times N$ matrix K therefore provides the errors at all orders, without a separate pass for each order. On Mamba-2, the exact variant thus requires one additional matrix accumulation rather than the filtering pass used for the other models (Table 2).

Checks. *The realization reproduces the model.* At full order, our realization preserves S4D accuracy and differs from the uncompressed network by at most 4×10^{-7} bits on SaShiMi and 3.1×10^{-4} in the S5 logits. For each reduced SaShiMi channel, we compare with the convolution kernel computed by the model itself, rather than our reconstruction. The median relative error is 5×10^{-5}. The reduced S5 layers are evaluated by an FFT convolution of their kernel in float64, since its relative error grows with the ℓ_1 norm of the layer's impulse response and float32 is not accurate enough for the slowest layers; the modal baselines use the model's own float32 scan. The S5 state of Table 2 is computed from the retained orders.

The Mamba-2 projection is an r-state layer. Every Mamba-2 perplexity is measured the cheap way, as GHOST's are: the released model runs at full size and only its readout changes, C_t replaced by $(UV)^\top C_t$ after the activation, a one-line hook on the HuggingFace implementation. Nonetheless, the deployed form is the genuine r-state layer, HuggingFace's own forward with $B_t \to VB_t$, $C_t \to U^\top C_t$ and the scan run on the r-dimensional state $z_t = Vh_t$. This is what the memory of §5.3 counts. We verified that the two agree on the 1.3B model over four 2,048-token test windows: in float32 the SSM outputs agree to 5.5×10^{-7} and the perplexities to 2×10^{-5}, and in bfloat16 or float16, the precision Figure 1 counts, the perplexity changes by at most 0.0011. The r-state layer keeps the input projection at full width and adds V and $U^\top$, $2rN$ weights per layer, whereas GHOST's pruned layer also drops the rows of the input projection and the convolution channels that produce the removed entries of B_t and C_t; at budget 0.10 on 1.3B it has 1.7% more parameters and FLOPs per token than GHOST.

Stability. Every truncated system has $\rho(A_{11}) < 1$ wherever Theorem 1*(i)* applies, that is wherever the trained system is itself stable: on the four-layer model, on SaShiMi and on every order and seed of LRA-Text, whose maximum is 0.9993. We checked every S5 cell. Those with $\rho(A_{11}) \geq 1$ are all on ListOps, whose checkpoints carry modes outside the unit disc, so the theorem's hypothesis fails before any truncation. The largest is $\rho(A_{11}) = 1.000675$.

Re-diagonalization. Every table evaluates the truncation in its balanced coordinates. `POISE` deploys it re-diagonalized, a change of coordinates that keeps the transfer function. We re-evaluated every S4D and SaShiMi classical-BT and `POISE` model in that form, exact-curve rows included, in float32: accuracy is identical on every S4D model and NLL is unchanged at the printed precision on SaShiMi, with eigenvector condition numbers at most 2.3×10^3. On S5, at budget 0.10, the re-diagonalized layers change no prediction.

A.1 The balanced truncation algorithm

Algorithm 2 details the standard steps for balancing once the input Gramian is provided.

Algorithm 2 Weighted Balanced Truncation (WBT) of one SSM

Part A — Balance. Require: (A, B, C); weights $P \succ 0$, $M \succ 0$
1: Solve $A^\top Q_M A - Q_M + C^\top M C = 0$
2: $P = RR^\top$, $Q_M = LL^\top$
3: $U\Lambda V^\top \leftarrow \mathrm{svd}(L^\top R)$, $S \leftarrow RV\Lambda^{-1/2}$
4: $(\bar{A}, \bar{B}, \bar{C}) \leftarrow (S^{-1}AS, S^{-1}B, CS)$
5: **return** $(\bar{A}, \bar{B}, \bar{C})$ and $(\sigma_1, \ldots, \sigma_n)$

Part B — Truncate. Require: Output of Part A and order r
6: **return** $(\bar{A}_{11}, \bar{B}_1, \bar{C}_1)$

A.2 Estimation error of the covariance

Proposition 1 requires $\eta < 1$, where η is the smallest number with $(1-\eta)P_{\mathbf{u}_T} \preceq \widehat{P} \preceq (1+\eta)P_{\mathbf{u}_T}$. The true covariance is not available, so η is measured against a reference estimate from 1,024 further training sequences, disjoint from the calibration set. Calibration sets of different sizes are prefixes of one draw. For each calibration size N, we first compute the median η across systems within each model. Table 3 reports the mean and standard deviation of these medians across models. It also reports the largest η across all systems and models, and the fraction of systems with $\eta < 1$.

Table 3: Estimation error η of the covariance against calibration size N.

four-layer S4D, 10 seeds, 256 systems each				S5, LRA-Text, 3 seeds, 6 layers each			
N	median	largest	$\eta < 1$	N	median	largest	$\eta < 1$
32	0.326 ± 0.010	1.758	98.75%	4	0.746 ± 0.039	0.937	100%
64	0.225 ± 0.006	1.004	99.96%	16	0.794 ± 0.016	1.119	89%
128	0.201 ± 0.005	0.977	100%	64	0.483 ± 0.055	0.757	100%
256	0.184 ± 0.004	0.825	100%	256	0.256 ± 0.135	0.535	100%

At $N = 256$, the size we use, every system satisfies $\eta < 1$, and most already do with far fewer sequences.

A.3 COST

Table 2 reports compression time from a checkpoint loaded in memory to the selected allocation. This includes constructing the state-space realization of each system but excludes evaluation. For S4D, SaShiMi and S5, the training reference is the median epoch time for the same checkpoints on the same GPU, namely 23.9 s, 744 s and 26 s respectively. For Mamba-2, the reference is its 300B-token pretraining (Dao & Gu, 2024), estimated at the GPU's peak bf16 rate. The reported fractions of pretraining time are therefore upper bounds. The exact variants additionally require the curve pass and, on SaShiMi and Mamba-2, a second calibration pass, since storing their recorded states is too costly.

Parallelism. After the calibration pass, every step of `POISE` is independent across systems. One pass through the uncompressed network accumulates the covariances of all layers; each system is then balanced on its own, each probe pass reduces a single system of the same uncompressed network, and each exact curve is computed from that system's own calibration inputs. These steps can therefore be distributed over processes or GPUs. The times in Table 2 use one GPU, with the per-channel balancing of S4D and SaShiMi run in parallel on the CPU cores. On Mamba-2 1.3B, the $J = 48$ probe passes take 1,825 of the 2,850 s of the spectral c^π variant and split directly across GPUs. GHOST calibrates each layer on the outputs of the already pruned layers before it, so its layers are processed one after another.

Protocols behind Table 2. The deployed models use diagonal recurrences with channels batched in four groups per layer. Channels in each group are padded to the largest order in that group. State memory is the storage of one stream's recurrent state, using float32, or float16 on Mamba-2. On SaShiMi, the recurrence of S4 stores each state as one complex entry, both members of a conjugate pair included, so a state takes 8 bytes in the original model and in the reduced one. S5 on LRA-Text is bidirectional, so its state is stored once per direction. A real mode counts as one state but takes a complex entry, so on S4D `POISE` stores about 11% more than its state count. State memory is reported for c^π, with the other variants within 4%. Timings use one GH200. State memory and time on S4D and S5 are averaged over seeds. The S4D times were measured at budget 0.10; only the final allocation step depends on the budget, and the sort takes 5 ms. Peak memory in Table 5 is measured by `torch.cuda.max_memory_allocated` during a full-length recurrent pass, excluding weights and inputs.

Table 4 counts floating-point operations, where one FLOP is one real addition or multiplication. Part (a) covers Algorithm 1, run once per network on N calibration sequences of length T. Of these, N_p sequences are used by the probes. C_F is the cost of one recurrent forward pass of F on one sequence, and $n_{\text{tot}} = \sum_j n_j$ is the total number of states. Part (b) covers one recurrent step of system j, with m_j inputs and p_j outputs, reduced to order r_j. The counts of part (a) are those of S4D and SaShiMi. On S5, Q_M comes from a QR decomposition of the Krylov stack, $O(T \sum_j p_j n_j^2)$, and the exact curve is evaluated by a scan at every order of each layer, $O(NT \sum_j n_j(n_j + p_j))$ per order; on Mamba-2 it costs one more calibration pass and $O(n_j^2)$ per layer.

Grouping the channels of a layer. The deployed recurrence batches the channels of each layer into one tensor and pads them to the largest order in that layer. Storage therefore depends on the largest channel order rather than only on the total number of retained states. Grouping channels and padding within each group changes the storage layout without changing the model. The channel recurrences are independent, and all measured layouts produce identical outputs.

We sort channels by order and use dynamic programming to find the partition into at most G contiguous groups that minimizes padded storage. Each group requires a separate kernel launch per step. Table 5 reports measurements for `POISE` with c^π on one GH200 in float32. These include recurrent state memory for one stream, peak memory at batch size 64, and time per step at batch sizes 64 and 1024, reported as the median of seven runs. Four groups per layer retain most of the memory savings of one group per order at substantially lower runtime cost. We therefore use four groups per layer in Table 2.

Table 4: FLOP counts, in order of magnitude. (a) Algorithm 1 with the spectral variant, once per network. (b) One inference step of one system.

(a) Algorithm 1, spectral variant

step	FLOPs
calibration pass, recording every state	$NC_F + O(NT\, n_{\mathrm{tot}})$
state covariances $\widehat{P}_j$	$O(NT \sum_j n_j^2)$
balancing: two Cholesky factors, one SVD, the congruence	$O\big(\sum_j n_j^3 + \sum_j (m_j + p_j) n_j^2\big)$
probe passes, one per system	JN_pC_F
allocation, one sort of n_{tot} tail decrements	$O(n_{\mathrm{tot}} \log n_{\mathrm{tot}})$
exact variant, error curves in place of the tails	$O\big(NT \log T\, n_{\mathrm{tot}} + T \sum_j n_j^3\big)$
exact variant, dynamic programming in place of the sort	$O(JR \max_j n_j)$
Γ *curve* (9)*, for reference*	$n_{\mathrm{tot}} N_p C_F + O(JR \max_j n_j)$

(b) inference, per time step

realization of system j	FLOPs
original, diagonal	$O\big((m_j + p_j)\, n_j\big)$
reduced, dense $\bar{A}_{11}$	$O\big(r_j^2 + r_j(m_j + p_j)\big)$
reduced, diagonalized	$O\big((m_j + p_j)\, r_j\big)$

Table 5: Memory and time of the deployed recurrence by channel grouping, `POISE` with c^π, $\pi = 0.15$, at budget 0.15; groups counted over the whole network. On SaShiMi, the channels that keep their original kernel (Appendix A) form one extra group in each of the six layers that have them.

				ms per step	
layout	groups	state	peak memory	batch 64	batch 1024
four-layer S4D, budget 0.15*, ten seeds*					
one group per layer	4	42.9 kB	5.13 MB	1.49	1.54
four groups per layer	16	13.9 kB	1.32 MB	3.60	3.66
one group per order	66	10.9 kB	0.87 MB	11.77	11.94
24*-layer SaShiMi, budget* 0.15					
one group per layer	30	0.615 MB	91.0 MB	11.01	11.31
four groups per layer	102	0.310 MB	40.1 MB	16.31	16.54
one group per order	406	0.268 MB	34.7 MB	33.97	34.83

A.4 Protocol for quantized methods

We count weights and caches from the released configurations in float16. A Mamba-2 recurrent state contains $d_{\mathrm{inner}} \times d_{\mathrm{state}}$ values per layer. A KV cache contains $2\times$ layers $\times$ KV heads $\times$ head dimension values per token, and we report its size at 2048 tokens. The resulting parameter counts agree with the published counts. Figure 1 plots the float32 perplexities of the tables; the deployed r-state model with its state stored in float16, the precision the footprint counts, reproduces them to 0.0011 (Appendix A). We report WikiText word perplexities under GHOST's evaluation harness, using windows of 2048 tokens and float32.

The state memory counts the SSM state. The convolution window, 1.25 MB per stream in float16 on Mamba-2 1.3B, is not reduced and is left out of Figure 1 and Table 2; at $N_{\mathrm{batch}} = 64$ it adds 80 MB, and the reduced model stays below Llama-3.2-1B with a four-bit cache. We evaluate the three Transformers (Grattafiori et al., 2024; Yang et al., 2024; Zhang et al., 2024) here. Our evaluation reproduces the published Pythia-1.4B (Biderman et al., 2023) value of 14.7181 to four decimal places.

The harness scores each window in one pass without reading back a cache. We therefore apply KIVI inside attention at the point where cached values would be used. Keys are quantized per channel in groups of 32 tokens, and values per token in groups of 32 channels. Each query uses its most recent 128 tokens in full precision. At 16 bits, this implementation reproduces the dense perplexities exactly. The cache memory count includes a 16-bit scale and zero point per group, together with

the full-precision window. We retain this window as in KIVI because it substantially improves the accuracy of two-bit quantization. Without it, the perplexity increase for Llama-3.2-1B is 18.6, compared with 3.09 when the window is retained.

B ABLATIONS

We first compare the complete methods across budgets in Appendix B.1, extending Section 5.1. We then isolate truncation at a fixed allocation in Appendix B.2 and allocation at a fixed truncation in Appendix B.3. The remaining tables vary one factor of the configuration in Section 5 at a time while holding the others fixed. Appendix A gives the experimental protocol.

B.1 THE COMPLETE METHODS AT EVERY BUDGET

Section 5.1 compares the complete methods at one budget per model. Here we report the budget grids. We tabulate the most aggressive budgets, where compression is hardest. At each budget, every method retains the same total number of states and distributes them across layers using its own allocation rule. Appendix C defines the rules and the scope of the baseline comparison.

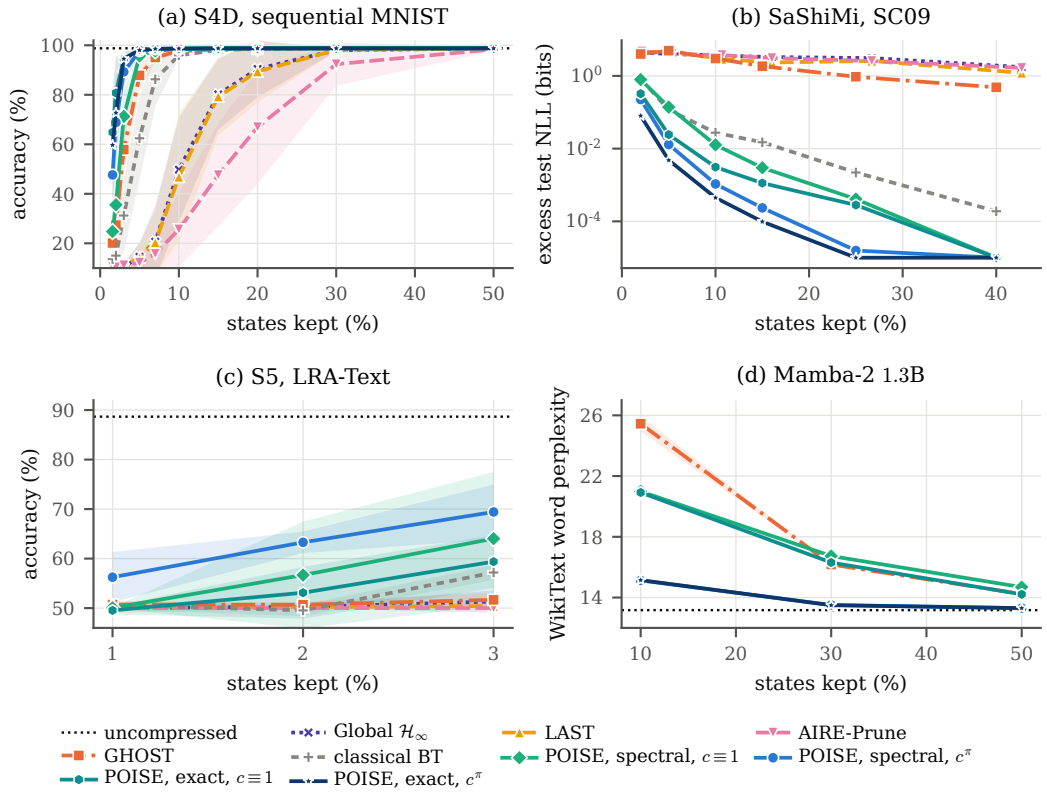


Figure 2: Predictive performance as a function of the retained state fraction. Each method distributes the global budget using its own allocation rule. Dotted lines indicate the uncompressed models. We report (a) accuracy on S4D over ten seeds, (b) excess test NLL on SaShiMi, (c) accuracy on S5 over three seeds and (d) WikiText word perplexity on Mamba-2 1.3B over three calibration draws. The corresponding results are given in Tables 6, 7, 8 and 9. Shaded bands show one standard deviation over seeds or draws.

B.1.1 FOUR-LAYER S4D

POISE with c^π is the best rule at almost every budget (Table 6). At the tightest budget, 0.10, it is ahead of GHOST and classical balanced truncation on every seed, and far ahead of the rules that read the parameters alone. Among the modal rules, scoring with calibration inputs is what lifts GHOST above Global $\mathcal{H}_\infty$. Balancing against the input covariance provides a further point. Figure 2 (top left) sweeps the budget from 1 to 50% of the states. POISE comes within a point of the uncompressed accuracy with about 5% of the states, GHOST and classical balanced truncation need 10 to 15%, and

Table 6: Accuracy (%) on S4D, sequential MNIST, ten seeds; uncompressed 98.78.

	budget			
rule	0.10	0.15	0.20	0.30
Global $\mathcal{H}_\infty$	49.74 ±21.1	80.17 ±14.3	90.47 ±11.5	98.02 ±0.60
LAST	46.84 ±25.7	79.32 ±15.8	89.39 ±12.5	98.09 ±0.53
AIRE-Prune	25.62 ±15.0	47.58 ±20.8	66.92 ±23.0	92.36 ±8.8
GHOST	97.68 ±0.43	98.35 ±0.30	98.59 ±0.23	98.72 ±0.22
classical BT	95.95 ±1.01	98.11 ±0.31	98.46 ±0.24	98.70 ±0.17
POISE, spectral, $c \equiv 1$	98.51 ±0.39	98.69 ±0.31	98.73 ±0.30	98.77 ±0.30
POISE, spectral, c^π, $\pi = 0.15$	**98.67** ±0.30	98.73 ±0.29	**98.77** ±0.28	98.77 ±0.29
POISE, exact, $c \equiv 1$	98.65 ±0.23	**98.75** ±0.24	98.75 ±0.27	**98.78** ±0.27
POISE, exact, c^π, $\pi = 0.15$	**98.67** ±0.32	**98.75** ±0.27	98.76 ±0.28	98.77 ±0.28

the parameter-only rules 30 to 50%. At 5% and 3% POISE leads GHOST by 9.9 and 31.4 points on every seed.

B.1.2 SaShiMi

Table 7: Excess test NLL (bits) on SaShiMi, SC09; uncompressed 1.9270.

	budget	
rule	0.15	0.25
Global $\mathcal{H}_\infty$	+3.2928	+3.0972
LAST	+2.3356	+2.5795
AIRE-Prune	+3.0889	+2.5758
GHOST	+1.8511	+0.9480
classical BT	+0.0150	+0.0022
POISE, spectral, $c \equiv 1$	+0.00298	+0.00041
POISE, spectral, c^π, $\pi = 0.15$	+0.00023	+0.00002
POISE, exact, $c \equiv 1$	+0.00113	+0.00028
POISE, exact, c^π, $\pi = 0.15$	**+0.00010**	**+0.000003**

On SaShiMi, we report the increase in test NLL, the model's training objective, relative to the uncompressed model (Table 7). With 15% of the states, POISE increases NLL by 0.01% of the uncompressed value. The ranking matches S4D, but the differences span several orders of magnitude: classical balanced truncation loses two orders of magnitude fewer bits than the modal rules, GHOST included, and POISE a further factor of 65 to 110, of which the input law alone gives about 5 and the probed scale the rest.

B.1.3 S5 on Long Range Arena

We run S5 on two Long Range Arena tasks, Text and ListOps, with three trained checkpoints each (Table 8). A system is a whole layer, and we use $M = I$ (Appendix E). The budgets differ by task: on ListOps every rule is within ten points of chance below budget 0.30, so its columns start there. Classical BT has no entry on ListOps, whose checkpoints carry modes outside the unit disc, where W_c is undefined.

The default, the spectral curve with c^π, gives the best result at every budget on both tasks, ahead of every baseline, classical BT included.

B.1.4 Mamba-2

On both models, c^π substantially reduces perplexity with the spectral curve, the more so the tighter the budget. The benefit of the exact curve decreases at tighter budgets: with $c \equiv 1$ it is slightly better than the spectral curve on 1.3B, and on 130M it is worse at budget 0.1.

Table 8: Accuracy (%) on S5, three seeds per task; uncompressed 88.67 on LRA-Text and 61.42 on LRA-ListOps. Aggressive budgets are shown on LRA-Text.

	LRA-Text, budget			LRA-ListOps, budget	
rule	0.01	0.02	0.03	0.30	0.50
Global $\mathcal{H}_\infty$	50.05 ±0.08	50.22 ±0.26	51.33 ±2.27	8.50 ±0.85	12.65 ±1.54
LAST	50.15 ±1.15	50.20 ±0.16	50.41 ±0.71	16.85 ±4.03	15.63 ±4.73
AIRE-Prune	50.00 ±0.01	50.00 ±0.00	50.02 ±0.03	17.10 ±8.80	25.62 ±16.1
GHOST	50.72 ±0.29	50.69 ±0.61	51.67 ±1.44	18.28 ±3.71	30.23 ±6.18
classical BT	50.63 ±0.89	49.53 ±1.16	57.19 ±1.52	—	—
POISE, spectral, $c \equiv 1$	50.05 ±0.09	56.67 ±10.8	64.03 ±13.5	20.25 ±2.73	30.90 ±6.63
POISE, spectral, c^π, $\pi = 0.15$	**56.24** ±5.09	**63.28** ±2.16	**69.42** ±5.55	**23.52** ±5.20	**42.02** ±11.7

Table 9: WikiText word perplexity on Mamba-2, mean over three calibration draws, standard deviation in small type; uncompressed 25.85 at 130M and 13.17 at 1.3B. GHOST is the only baseline of Appendix C that applies to a selective SSM. Rows marked † are reported by Menezes & Kyrillidis (2026) under the same protocol, from one run, and prune state channels at the same sparsity as GHOST; at 130M only budget 0.5 is reported.

	budget		
rule	0.5	0.3	0.1
Mamba-2 130*M*			
Magnitude†	910.0	—	—
Random†	45.60	—	—
Taylor†	10^{16}	—	—
GHOST	29.03 ±0.03	37.82 ±0.12	65.12 ±0.71
POISE, spectral, $c \equiv 1$	29.16 ±0.02	32.64 ±0.04	41.34 ±0.19
POISE, spectral, c^π, $\pi = 0.15$	**26.36** ±0.00	27.23 ±0.01	**34.34** ±0.26
POISE, exact, $c \equiv 1$	28.88 ±0.01	32.52 ±0.07	44.69 ±0.19
POISE, exact, c^π, $\pi = 0.15$	26.37 ±0.01	**27.21** ±0.00	34.52 ±0.19
Mamba-2 1.3*B*			
Magnitude†	29.34	39.96	69.49
Random†	17.77	30.14	64.21
Taylor†	13.94	4255	4690
GHOST	14.23 ±0.00	16.18 ±0.02	25.44 ±0.52
POISE, spectral, $c \equiv 1$	14.69 ±0.02	16.73 ±0.03	20.99 ±0.03
POISE, spectral, c^π, $\pi = 0.15$	**13.30** ±0.01	13.51 ±0.01	**15.12** ±0.04
POISE, exact, $c \equiv 1$	14.21 ±0.04	16.30 ±0.04	20.92 ±0.04
POISE, exact, c^π, $\pi = 0.15$	**13.30** ±0.01	**13.49** ±0.00	15.15 ±0.04

Menezes & Kyrillidis (2026) also report SparseGPT (Frantar & Alistarh, 2023) on the weights that produce B_t and C_t. It zeroes individual weights and keeps the full state, so it is not in Table 9. On 1.3B with 50, 70 and 90% of those weights removed it reaches 13.25, 13.52 and 15.51. POISE with c^π nearly matches it at the first two sparsities while removing the state instead, and is 0.39 lower at the third.

B.2 The truncation, at a fixed allocation

We fix each system's order to $r_j = \varepsilon n_j$ before any score is computed, so no rule moves states between the channels of a layer; on the LTI models, modal rules keep whole conjugate pairs, so the orders are even. At these fixed orders, LAST's normalization and AIRE-Prune's score keep the order of the modes within a system, so LAST retains the same states as Global $\mathcal{H}_\infty$, while AIRE-Prune retains those selected by its raw score (Appendix C). Table 10 therefore compares the remaining distinct truncation rules.

Table 10: The truncation at a fixed allocation, $r_j = \varepsilon n_j$ for every rule. Accuracy (%) on S4D and S5, higher is better; excess test NLL (bits) on SaShiMi and WikiText word perplexity on Mamba-2 1.3B, lower is better. Uncompressed: 98.78, 1.9270, 88.67, 13.17. On Mamba-2 a system is a layer and the data-free rules are not defined.

coordinates	score	S4D $\varepsilon = 6/64$	SaShiMi $\varepsilon = 10/64$	S5, LRA-Text $\varepsilon = 0.10$	Mamba-2, 1.3B $\varepsilon = 0.10$
modal	$\mathcal{H}_\infty$: Global $\mathcal{H}_\infty$, LAST	35.59 ±14.8	+2.8217	53.57 ±3.33	—
modal	$\mathcal{H}_2$: AIRE-Prune	34.98 ±13.4	+3.2903	56.30 ±10.7	—
modal	GHOST, as published	84.93 ±6.87	+3.3824	52.15 ±2.06	25.44 ±0.52
balanced	classical BT	86.53 ±5.50	+0.00325	74.82 ±4.36	—
balanced	ours, Algorithm 2	**97.02** ±1.06	**+0.00068**	**75.99** ±6.14	**18.60** ±0.07

B.3 The allocation, at a fixed truncation

Section B.2 compares truncations at a fixed allocation. Here we fix the truncation to that of POISE and vary the allocation. Using the state budget of the uniform allocation in Table 10 and budget 0.03 on S5, Table 11 compares GHOST's equal per-layer share, LAST's per-layer score normalization, and our pooled sort with either $c \equiv 1$ or c^π.

Table 11: The allocation with our truncation fixed. On S5 and Mamba-2 a layer is one system, so the per-layer share is the uniform allocation. On S5 the uniform allocation is GHOST's even split over the six layers.

allocation	S4D $\varepsilon = 6/64$	SaShiMi $\varepsilon = 10/64$	S5, LRA-Text $\varepsilon = 0.03$	Mamba-2, 1.3B $\varepsilon = 0.10$
uniform, $r_j = \varepsilon n_j$	97.02 ±1.06	+0.00068	50.13 ±0.22	18.60 ±0.07
same share per layer (GHOST's)	98.60 ±0.23	+0.00060	= uniform	= uniform
LAST's normalization	98.40 ±0.35	+0.00245	52.90 ±4.96	—
POISE, spectral, $c \equiv 1$	98.42 ±0.37	+0.00228	64.03 ±13.5	20.99 ±0.03
POISE, spectral, c^π, $\pi = 0.15$	98.61 ±0.34	+0.00015	**69.42** ±5.55	**15.12** ±0.04
POISE, exact, $c \equiv 1$	**98.65** ±0.23	+0.00090	59.38 ±5.37	20.92 ±0.04
POISE, exact, c^π, $\pi = 0.15$	**98.65** ±0.32	**+0.00007**	56.23 ±9.15	15.15 ±0.04

The probed scale gives the best allocation for each curve on SaShiMi and Mamba-2, and the best or a tie on S4D. On SaShiMi and Mamba-2 the probe is what makes pooling pay; on S4D and S5 pooling pays even without it.

B.4 Applying the scaling rule to related work

Table 12 compares combinations of score and scale on the four-layer model over ten seeds. A *score* assigns a cost to deleting a state in the coordinates used by the rule. A *scale* weights these costs across systems. Comparisons within a score block vary the scale, while comparisons across blocks at the same scale vary the score.

The probed scale in §4.2 does not depend on the POISE score and can therefore be applied heuristically to any rule that sums per-state scores. We compute c_j using the rule's own score and construct the probe by deleting its lowest-scoring modes. It improves AIRE-Prune at every budget and LAST at the tightest, on most seeds, whereas the rules' original normalizations mostly do worse than pooling their raw scores. The POISE score gives the highest accuracy at every scale and budget on all ten seeds, far ahead of both modal scores.

On SaShiMi, the probed scale also lowers the modal rules' excess NLL, more at the larger budget, while their original normalizations help little or not at all. At the same scale, modal truncation still yields excess NLL three to four orders of magnitude larger than balanced truncation.

Table 12: Score against scale. S4D: accuracy (%), ten seeds; SaShiMi: excess test NLL (bits), uncompressed 1.9270. The modal rules' own-normalization rows are those of Tables 6 and 7.

		S4D, budget				SaShiMi, budget	
score	scale	0.10	0.15	0.20	0.30	0.15	0.25
POISE	$c_j \equiv 1$	98.51 ±0.4	98.69 ±0.3	98.73 ±0.3	**98.77** ±0.3	+0.00298	+0.00041
	$c_j = c_j^\pi$, $\pi = 0.15$	**98.67** ±0.3	**98.73** ±0.3	**98.77** ±0.3	**98.77** ±0.3	**+0.00023**	**+0.00002**
LAST	$c_j \equiv 1$	49.74 ±21.1	**80.17** ±14.3	90.47 ±11.5	98.02 ±0.6	+3.293	+3.097
	own normalization	46.84 ±25.7	79.32 ±15.8	89.39 ±12.5	**98.09** ±0.5	+2.336	+2.580
	$c_j = c_j^\pi$, $\pi = 0.15$	**55.05** ±24.5	75.44 ±22.2	**91.56** ±7.0	97.44 ±1.7	**+1.268**	**+0.433**
AIRE-Prune	$c_j \equiv 1$	30.57 ±14.5	51.11 ±17.2	79.09 ±12.1	95.42 ±4.2	+3.530	+2.449
	own normalization	25.62 ±15.0	47.58 ±20.8	66.92 ±23.0	92.36 ±8.8	+3.089	+2.576
	$c_j = c_j^\pi$, $\pi = 0.15$	**52.40** ±27.7	**74.17** ±22.9	**93.00** ±5.7	**97.31** ±1.8	**+1.630**	**+0.854**

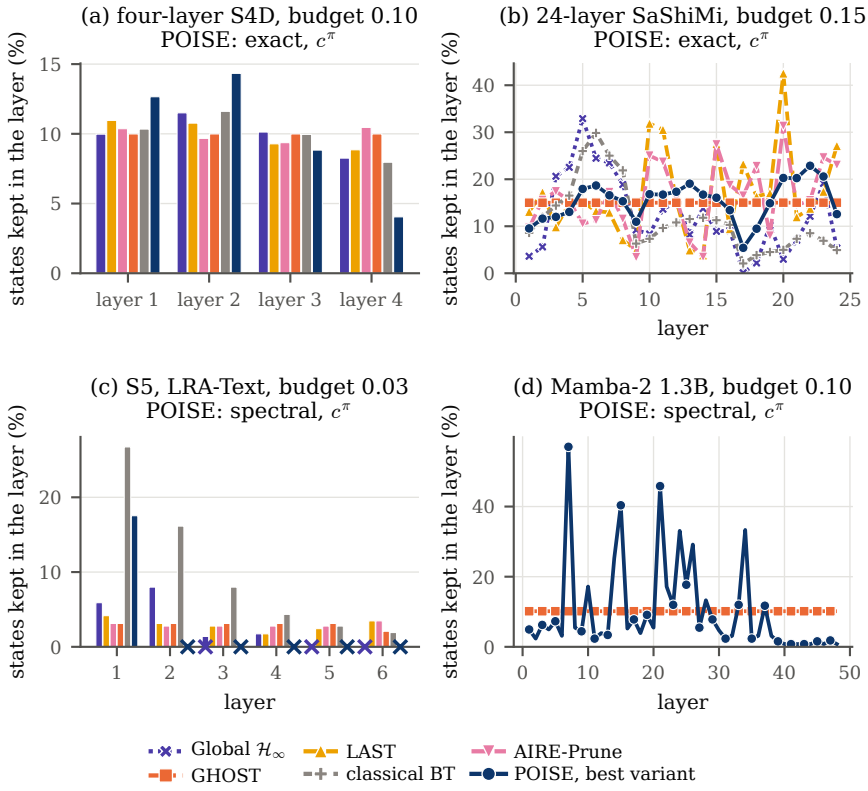


Figure 3: Per-layer allocation of retained states: (a) S4D at budget 0.10, (b) SaShiMi at 0.15, (c) S5 at 0.03 and (d) Mamba-2 1.3B at 0.10, with POISE in its best variant at that budget (named in the panel). The rules are drawn in the order of Table 6. Classical BT is shown on S4D, SaShiMi and S5; there is no Mamba-2 run of it. A cross below a bar or line marks a layer with no retained states on at least one seed.

B.5 The rank allocations

Figure 3 shows where each rule places its states. No rule reserves a state per system, and all of them leave some empty. At budget 0.10, POISE leaves 36 ($c \equiv 1$) to 66 (c^π) of the 256 S4D channels without a state, the modal rules 43 to 73, and GHOST 61. On S5, where each system is a whole layer, Global $\mathcal{H}_\infty$ leaves one or two layers without states on each of the three seeds at budget 0.03. At a fixed allocation every system keeps εn_j states, so no system is empty, and the modal rules still fall far below our truncation (Table 10). Conversely, GHOST empties 61 channels in its own allocation

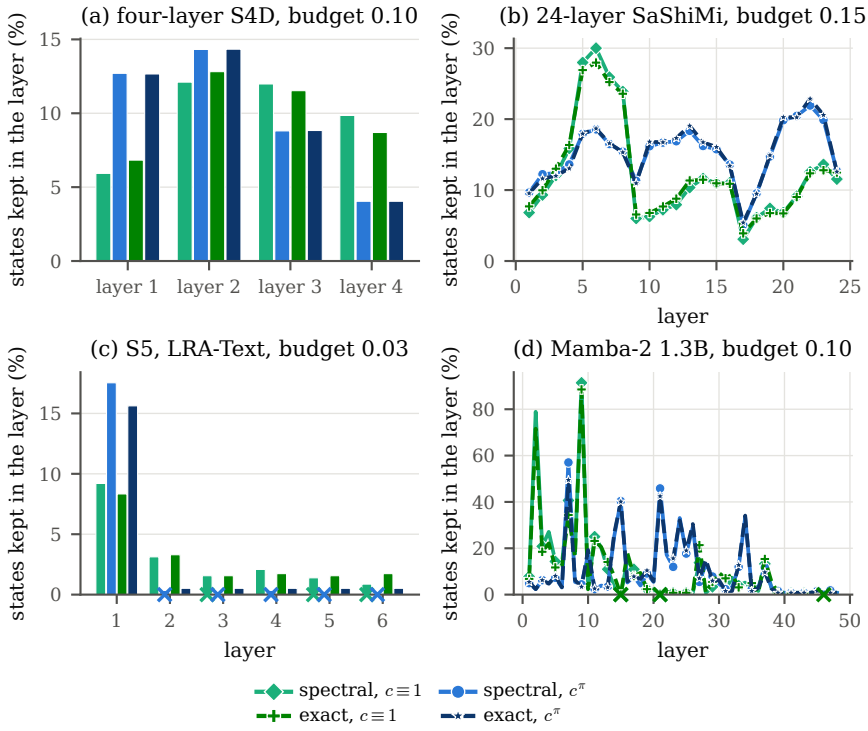


Figure 4: POISE's four allocations, spectral or exact curve and $c \equiv 1$ or c^π ($\pi = 0.15$), on the same models and budgets: (a) S4D, (b) SaShiMi, (c) S5 and (d) Mamba-2 1.3B. Crosses have the same meaning as in Figure 3; with no state reserved, the probed scale puts almost all of the S5 states in the first layer.

and stays close to the uncompressed model. These results show that the quality of the truncation is a major source of the gap, and that empty systems do not explain it. In its best variant on each model, POISE gives the earlier layers more states on S4D, S5 and Mamba-2. On SaShiMi its allocation is close to uniform across layers.

Figure 4 compares the four variants of POISE. Under the same scale rule the spectral and exact curves give nearly the same allocation, moving at most 11% of the states between layers. This is why the spectral variant is close to the exact one on S4D and Mamba-2 in Table 1 without the curve pass, at less cost (Table 2). Changing the scale rule moves 18 to 64% of the states: the probed scale is what changes the allocation, and it gives the best results on S4D, SaShiMi and Mamba-2 (Table 11).

Figures 5 and 6 show the order of every channel at budget 0.15. The deployed recurrence pads each channel to the largest order in its layer, shown in grey. Concentrating states in fewer channels therefore increases padded storage. On S4D, the ratios of stored to retained states are 3.6 and 4.3 for the spectral curve with $c \equiv 1$ and c^π respectively, and 3.3 and 4.2 for the exact curve, averaged over ten seeds (Figure 5 draws seed 0). Grouping the channels of a layer by order removes most of this padding (Appendix A.3), which is why Table 2 uses four groups per layer.

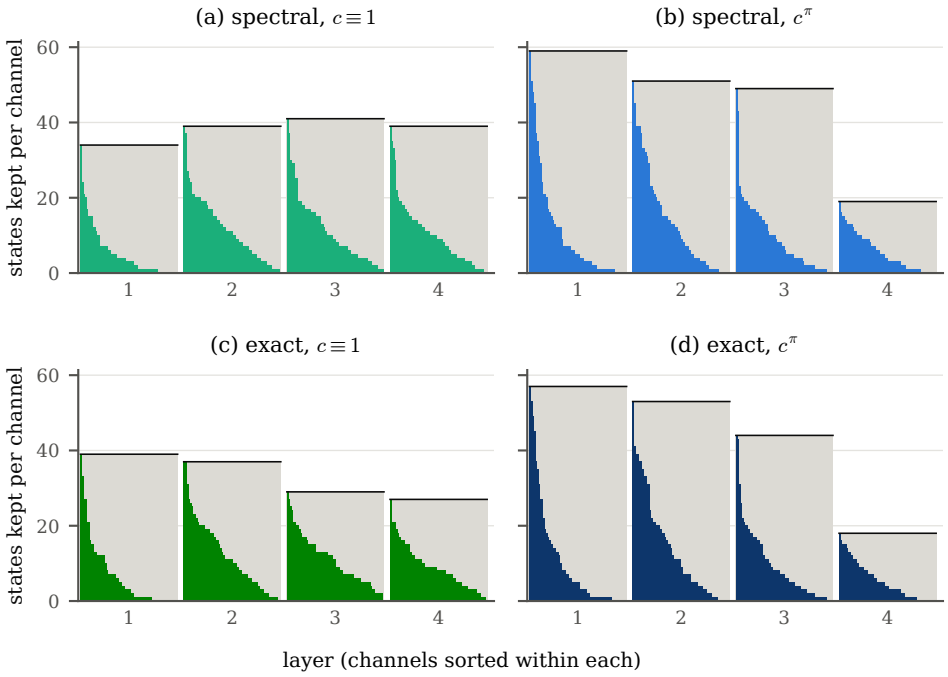


Figure 5: States kept per channel (colour) and stored after padding each channel to its layer's widest (grey), under POISE's four allocations on the four-layer S4D model with seed 0 and budget 0.15. Channels are sorted within each layer. Every panel retains 2,450 states. Padding stores 9,792 (4.00×) with (a) spectral $c \equiv 1$, 11,392 (4.65×) with (b) spectral c^{π}, 8,448 (3.45×) with (c) exact $c \equiv 1$ and 11,008 (4.49×) with (d) exact c^{π}.

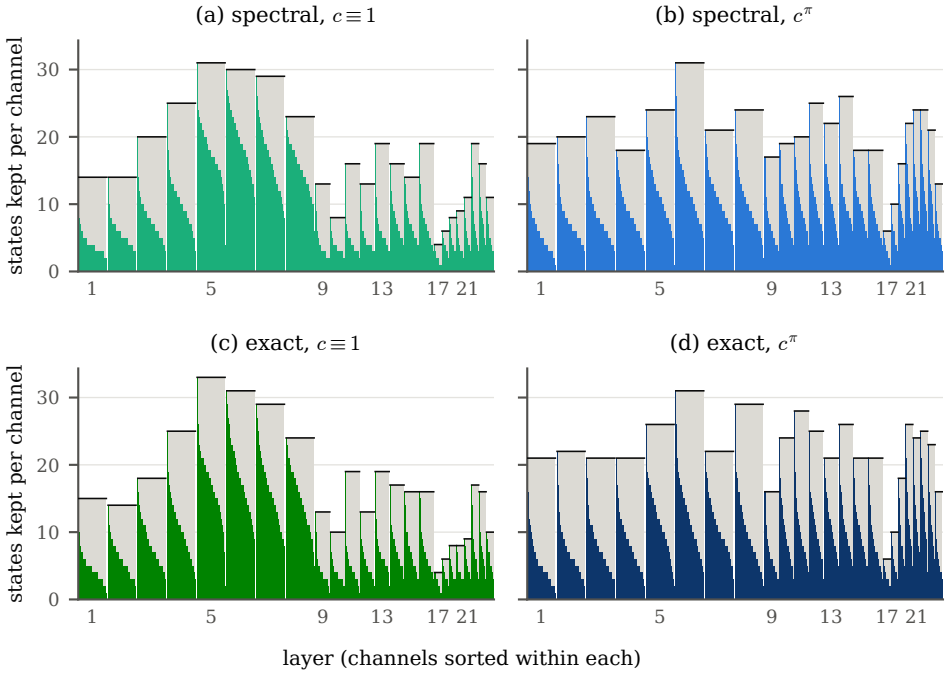


Figure 6: Retained and stored states for the 24-layer SaShiMi model at budget 0.15, using the format of Figure 5. Every panel retains 32,216 states. Padding stores 68,096 (2.11×), 75,904 (2.36×), 69,120 (2.15×) and 82,176 (2.55×), in the order of the panels (a)–(d).

B.6 CALIBRATING ON A DIFFERENT LAW

POISE uses calibration inputs from the deployment distribution. To examine distribution mismatch, Table 13 compares two reductions of the SaShiMi model trained on speech (SC09). We calibrate on 320 clips of either speech or piano (YouTubeMix, uncompressed NLL 2.1897 bits), then evaluate both reductions on both datasets. Matching the calibration and evaluation distributions lowers the excess NLL at both budgets, and even a mismatched calibration keeps it more than an order of magnitude below classical balanced truncation and three below LAST, neither of which uses calibration data.

Table 13: Excess test NLL (bits) on SaShiMi trained on speech; POISE with the spectral curve and c^π.

		tested on speech		tested on piano	
rule	calibrated on	0.15	0.25	0.15	0.25
LAST	none	+2.3356	+2.5795	+2.3530	+2.2365
classical BT	none	+0.0150	+0.0022	+0.0627	+0.0060
POISE	speech	**+0.00023**	**+0.00002**	+0.00076	+0.00008
POISE	piano	+0.00050	+0.00009	**+0.00029**	**+0.00001**

B.7 THE PROBED CRITERION AGAINST THE Γ CURVE

Table 14 solves (9) exactly by dynamic programming on the Γ curve, the one-at-a-time loss $\mathbb{E}\|\boldsymbol{\delta}_j\|^2_{\Gamma_j}$ measured at every order of every system, and compares it with the probed criterion ϕ_{c^π} of §4.2 and with $c \equiv 1$. This reference is computationally feasible here because S4D is small. Evaluating the Γ curve requires one network pass per candidate order of each system, giving 16,384 passes per seed compared with 256 for the probes. The count grows with the total state dimension, to about 2×10^5 passes on SaShiMi and 6,144 on Mamba-2 1.3B. For the latter, one pass over the calibration set takes about 40 s, giving approximately 70 GPU-hours for the Γ curve compared with 30 minutes for the probes.

The table reports accuracy and the ratio of $\sum_j \mathbb{E}\|\boldsymbol{\delta}_j\|^2_{\Gamma_j}$ to its optimum on the probe family. Both are reported as mean $\pm$ standard deviation over ten seeds.

Table 14: The Γ curve against one probe: accuracy (%) on S4D, ten seeds, and the time per seed on one GH200: for the Γ curve, the whole run over all four budgets, its 12 evaluations included, so an upper bound; for the sorts, one allocation, as in Table 2.

	budget				
	0.10	0.15	0.20	0.30	time
Γ curve (9), dynamic programming	98.75 ±0.23	98.77 ±0.27	98.76 ±0.29	98.77 ±0.27	$<$ 161 s
POISE, c^π, $\pi = 0.15$, sort	98.67 ±0.30	98.73 ±0.29	98.77 ±0.28	98.77 ±0.29	5.2 s
POISE, $c \equiv 1$, sort	98.51 ±0.39	98.69 ±0.31	98.73 ±0.30	98.77 ±0.30	3.3 s
$\sum_j \mathbb{E}\|\boldsymbol{\delta}_j\|^2_{\Gamma_j}$ / optimum, c^π, $\pi = 0.15$	4.3 ±0.9	4.2 ±0.7	5.0 ±0.8	4.4 ±0.7	
$\sum_j \mathbb{E}\|\boldsymbol{\delta}_j\|^2_{\Gamma_j}$ / optimum, $c \equiv 1$	8.4 ±2.0	11.2 ±2.5	14.5 ±4.4	19.8 ±7.5	

The Γ-curve optimum improves accuracy slightly at budget 0.10 and 0.15, on eight of ten seeds, and not at larger budgets, for 64 times as many forward passes. The probed criterion comes much closer to the optimum of that objective than $c \equiv 1$ does.

B.8 A CORRECTION FOR THE PROBE

The probe estimates each rate at r_j^π, which may differ from the allocated order r_j. We test one correction that re-probes every system at its allocated order, replaces c_j^π with $\sqrt{c_j^\pi \, c_j(r_j)}$, and repeats the sort. Here $c_j(r_j)$ is the ratio of §4.2 measured at the allocated order. The correction requires J additional passes per budget.

The correction helps where the probe order differs substantially from the allocated order (Table 15): at the tightest budgets on S4D, on SaShiMi and on S5. On Mamba-2 it does not help: on 130M the re-probed rate differs from c_j^π by more than a factor of two in only 2 of the 24 layers.

Table 15: One correction against c^π ($\pi = 0.15$, spectral curve). Accuracy (%) on S4D (ten seeds) and S5 LRA-Text (three seeds), excess test NLL (bits) on SaShiMi, WikiText perplexity on Mamba-2 (130M, three draws; 1.3B, one draw).

model	budget	c^π	one correction
S4D	0.03	89.26 ±6.10	**93.91** ±1.64
	0.05	97.73 ±0.77	**98.21** ±0.27
	0.10	98.67 ±0.30	**98.72** ±0.25
SaShiMi	0.05	+0.01296	**+0.01038**
	0.10	+0.00107	**+0.00087**
	0.15	+0.00023	**+0.00017**
S5, LRA-Text	0.07	78.56 ±7.86	**84.79** ±4.29
Mamba-2, 130M	0.10	**34.34** ±0.26	35.38 ±0.26
Mamba-2, 1.3B	0.10	**15.16**	15.29

Table 16 compares individual probe orders with two ways of combining the rates measured at nine probe orders, $\pi \in \{0.01, 0.03, 0.06, 0.10, 0.15, 0.22, 0.32, 0.45, 0.65\}$. The table lists five of these as single probes. The *geomean* variant uses the geometric mean of the nine rates, while *logmid* uses the midpoint of their range on a logarithmic scale. Accuracy is reported as mean ± standard deviation over ten seeds.

Table 16: Probe order: accuracy (%) on S4D, ten seeds.

	budget			
scale	0.10	0.15	0.20	0.30
$c \equiv 1$	98.51 ±0.39	98.69 ±0.31	98.73 ±0.30	98.77 ±0.30
c^π, $\pi = 0.01$	98.69 ±0.24	98.76 ±0.22	98.76 ±0.26	98.77 ±0.28
c^π, $\pi = 0.06$	98.71 ±0.24	98.77 ±0.26	98.77 ±0.28	98.77 ±0.28
c^π, $\pi = 0.15$	98.67 ±0.30	98.73 ±0.29	98.77 ±0.28	98.77 ±0.29
c^π, $\pi = 0.32$	98.68 ±0.22	98.76 ±0.27	98.77 ±0.30	98.77 ±0.29
c^π, $\pi = 0.65$	98.33 ±0.39	98.70 ±0.28	98.72 ±0.31	98.78 ±0.27
geomean	98.72 ±0.22	98.77 ±0.25	98.78 ±0.27	98.77 ±0.28
logmid	98.70 ±0.26	98.77 ±0.24	98.77 ±0.28	98.77 ±0.29

Across a sample of 192 channels of the 24-layer model, the rates vary by a median factor of 5 over the probe orders. On two of the three S5-Text checkpoints, probing the first layer at $\pi = 0.15$ reverses confident predictions, and the correction described above suits that layer (Table 15).

B.9 Calibration size

Calibration inputs are used both to estimate the covariances $\widehat{P}_j$, using N inputs, and to evaluate the probe in §4.2, using N_p inputs. Both sets are prefixes of the same draw, so the total requirement is $\max(N, N_p)$ inputs rather than $N + N_p$. The main experiments take $N = N_p = 256$ on the four-layer model and $N = 320$, $N_p = 8$ clips on SaShiMi. We vary the two sizes separately, then together. Four-layer numbers are mean ± standard deviation over ten seeds on 10,000 test sequences; *moved* is the percentage of the budget allocated differently from the largest size, 2048. The uncompressed model scores 98.78.

The covariance alone. The rule $c \equiv 1$ has no probe, so its rows in Table 17 vary N alone. Four sequences already match the accuracy of 2048 at every budget and move little of the budget. SaShiMi behaves alike: with $c \equiv 1$, 10 clips give nearly the excess NLL of 320.

Table 17: Calibration size N on S4D, ten seeds: accuracy (%), and *moved*, the percentage of the budget that the allocation places in different systems than it does at $N = 2048$. For c^π ($\pi = 0.15$) the probes use the same N sequences ($N_p = N$).

	budget 0.10		budget 0.15		budget 0.20		budget 0.30	
N	acc.	moved	acc.	moved	acc.	moved	acc.	moved
POISE, c^π, $\pi = 0.15$								
4	98.31 ±0.49	39 ±8	98.62 ±0.42	37 ±7	98.73 ±0.30	35 ±7	98.78 ±0.27	29 ±6
8	98.44 ±0.46	33 ±7	98.70 ±0.26	31 ±7	98.76 ±0.26	29 ±6	98.78 ±0.26	24 ±5
16	98.60 ±0.23	30 ±7	98.70 ±0.29	28 ±6	98.77 ±0.25	25 ±5	98.76 ±0.28	21 ±5
32	98.66 ±0.20	24 ±6	98.73 ±0.29	22 ±5	98.76 ±0.28	21 ±5	98.77 ±0.29	16 ±4
128	98.69 ±0.19	17 ±3	98.74 ±0.28	15 ±2	98.78 ±0.29	14 ±2	98.77 ±0.28	11 ±2
256	98.67 ±0.30	11 ±1	98.73 ±0.29	10 ±2	98.77 ±0.28	10 ±1	98.77 ±0.29	8 ±1
512	98.69 ±0.21	8 ±1	98.76 ±0.25	7 ±1	98.77 ±0.28	7 ±1	98.77 ±0.29	6 ±1
2048	98.70 ±0.20	—	98.75 ±0.28	—	98.78 ±0.27	—	98.77 ±0.29	—
POISE, $c \equiv 1$								
4	98.45 ±0.36	7 ±1	98.70 ±0.30	6 ±1	98.71 ±0.31	5 ±1	98.77 ±0.30	4 ±0
32	98.48 ±0.40	2 ±0	98.69 ±0.31	2 ±0	98.72 ±0.32	2 ±0	98.77 ±0.29	1 ±0
256	98.51 ±0.39	1 ±0	98.69 ±0.31	1 ±0	98.73 ±0.30	1 ±0	98.77 ±0.30	0 ±0
2048	98.45 ±0.37	—	98.69 ±0.33	—	98.73 ±0.30	—	98.77 ±0.30	—

The probe alone. Table 18 fixes the covariance ($N = 256$ sequences on the four-layer model, 320 clips on SaShiMi) and varies N_p only. On the four-layer model it reports accuracy at budget 0.10; on SaShiMi, the excess bits at budgets 0.25 and 0.15.

Table 18: Probe sample size N_p for c^π ($\pi = 0.15$), with the covariance fixed ($N = 256$ on S4D, 320 clips on SaShiMi). S4D: accuracy (%) at budget 0.10, ten seeds, and *moved*, the percentage of the budget placed in different systems than with $N_p = 2048$ (for $c \equiv 1$, than c^π with $N_p = 2048$). SaShiMi: excess test NLL (bits) at budgets 0.15 and 0.25.

four-layer S4D, budget 0.10			SaShiMi, one checkpoint		
N_p	accuracy	moved	N_p	0.15	0.25
4	98.37 ±0.49	39 ±8	1	+0.00042	+0.00005
8	98.39 ±0.47	33 ±8	2	+0.00032	+0.00004
32	98.67 ±0.19	24 ±7	4	+0.00026	+0.00003
256	98.67 ±0.30	11 ±1	8	+0.00023	+0.00002
2048	98.70 ±0.20	—	16	+0.00026	+0.00001
$c \equiv 1$	98.51 ±0.39	60 ±5	32	+0.00021	+0.00001

On the four-layer model 32 probe sequences suffice. SaShiMi predicts every sample, 16,000 predictions per clip, so four clips already suffice.

Both together. The c^π rows of Table 17 set $N_p = N$, the setting of the four-layer model. A few dozen sequences put c^π close to $N = 2048$ at every budget. Since performance under covariance-only calibration stabilizes from $N = 4$, these differences reflect the probe sample size and agree with Table 18.

C BASELINES

Table 19 summarizes the five baseline rules. The following paragraphs define each rule and its implementation, then explain the scope of the comparison relative to other compression methods.

Table 19: The five baselines. *Coordinates*: whether the rule changes the state coordinates before truncation. *Data* indicates whether scoring uses calibration inputs. *Allocation* describes how orders are assigned across systems. Classical BT has no allocation rule of its own. In our tables it is allocated with the pooled sort of `POISE` at $c \equiv 1$.

rule	coordinates	data	allocation
Global $\mathcal{H}_\infty$	none, modal	no	one threshold on the pooled score
LAST	none, modal	no	one threshold on a per-layer normalized score
AIRE-Prune	none, modal	no	as LAST, with an $\mathcal{H}_2$ score
GHOST	none, modal	yes	the same sparsity in every layer
classical BT	balanced at $P = W_c$	no	a pooled sort (spectral) with $c \equiv 1$ on unweighted singular values
`POISE`	balanced at $P = \widehat{P}_j$	yes	a pooled sort (spectral) or dynamic programming (exact), with c_j^π or $c \equiv 1$

Every baseline uses the same real state count as `POISE`, up to rounding, with each retained complex mode counted as two states. Four of the five delete states in the model's own coordinates, without balancing. Three of those, Global $\mathcal{H}_\infty$, LAST and AIRE-Prune, score the modes of a diagonal state matrix using only the parameters. GHOST uses calibration data to score states, deleting modes in diagonal systems and state channels in Mamba-2. All four apply directly to diagonal systems in S4D and S5. They apply to SaShiMi's DPLR layers after diagonalization, which reproduces the model's kernel with a median relative error of 10^{-5} and an NLL change of $+0.0000$ bits without pruning. The fifth, classical BT, is a balanced truncation and applies to any realization. For a diagonal system with modes λ_i, discretized input column $\bar{B}_i$ and output column C_i, write

$$q_i := \frac{\|C_i\|^2 \, \|\bar{B}_i\|^2}{\big(1 - |\lambda_i|\big)^2}, \qquad q_i' := \frac{\|C_i\|^2 \, \|\bar{B}_i\|^2}{1 - |\lambda_i|^2}, \tag{11}$$

the squared $\mathcal{H}_\infty$ and $\mathcal{H}_2$ norms of the first-order subsystem of mode i.

Global $\mathcal{H}_\infty$. This rule pools the scores q_i across the model and applies a single threshold.

LAST (Gwak et al., 2024). A layer is one block-diagonal system whose modes are sorted by descending q_i. The score of mode i is the rank-dependent ratio

$$\mathrm{LAST}(i) = \frac{q_i}{\sum_{j \le i} q_j}, \tag{12}$$

pooled over the model and thresholded once, with no state reserved per channel. The score is input-agnostic and invariant under $C \mapsto \alpha C$ within a layer. At its authors' operating points, our implementation loses 1.12 points on LRA-Text with 60% of the states removed, over three checkpoints, against the 0.36 that Gwak et al. (2024) report.

AIRE-Prune (Padhy et al., 2026). This rule uses the same construction as LAST, replacing q_i with q_i' in (12). The denominator $1 - |\lambda_i|^2$ vanishes on the unit circle and is negative outside it, where the score no longer represents the energy of a stable mode. None of the $81{,}920$ modes of the four-layer models or the $114{,}688$ modes of SaShiMi lies on or outside the unit circle. The S5 checkpoints on ListOps contain such modes. We give these modes an infinite score in every modal rule, since an unstable mode has infinite energy, so they are always retained; there are 3 to 5 per checkpoint. Balancing treats them like every other state (Appendix A).

GHOST (Menezes & Kyrillidis, 2026). GHOST scores each state channel of a Mamba-2 layer by its output energy on calibration inputs. The score sums the squared state multiplied by the squared

output projection over tokens. Channels below a threshold are deleted without changing coordinates. For a diagonal SISO system the same score is

$$s_i = |C_i \bar{B}_i|^2 \, \frac{1}{NT} \sum_{m=1}^{N} \sum_{k=1}^{T} \Big| \sum_{\tau=0}^{k-1} \lambda_i^{\tau} \, u_{k-1-\tau}^{(m)} \Big|^2 ,$$

the exact output energy of mode i under the calibration inputs. We follow the published procedure by pruning every layer to the same sparsity. Modes are pooled only across channels within a layer, with no state reserved per channel. Layers are processed in execution order, using activations from the network with the preceding layers already pruned. The layers therefore cannot be scored in a single pass. On SaShiMi this takes 24 calibration passes, one per layer.

Classical BT. This baseline is our own construction. We apply balanced truncation to each system using the controllability Gramian, $P = W_c$, and $M = I$, the data-free case of Theorem 1. It has no allocation rule of its own, so we allocate it with the pooled sort of `POISE` on its unweighted tail $\sum_{i>r_j} \sigma_i^2$ ($c \equiv 1$), with no state reserved. Compared with `POISE` using the spectral curve and $c \equiv 1$, this baseline changes only the input law. Bittner et al. (2025) apply the same reduction after training to S-Edge layers. Their allocation procedure is discussed below.

Methods not among the baselines. We compare against rules that reduce the state dimension of a trained model's SSM systems without retraining, to match the compression objective of `POISE` and allow budgets to be matched system by system. The methods below modify other structures, use different budget units or include training.

- *Quantization* (Abreu et al., 2024; Chiang et al., 2025b;a) lowers the precision of weights, activations or the stored state but keeps the state dimension, so it cannot be run at a given state count. It is complementary to `POISE`.
- *Weight or block pruning.* Mamba-Shedder (Muñoz et al., 2025) removes whole blocks or SSM modules, so the state shrinks only by whole layers. SparseGPT (Frantar & Alistarh, 2023) and Shihab et al. (2025) zero individual weights and keep the state dimension; the latter also fine-tunes.
- *Reduction within training.* CompreSSM (Chahine et al., 2026), Ezoe & Sato (2024), Sakamoto & Sato (2025b), Forgione et al. (2024) and Schwerdtner et al. (2025) reduce during training, retrain after reducing, or regularize training, so they do not apply to a given checkpoint. The truncations they apply after training are balanced or modal, which our classical BT and modal baselines cover.

CoBRAS (Otto et al., 2023) is the closest construction to ours. It also balances two covariances estimated from data: on the input side the covariance of the state, as we do, and on the output side the covariance of the sensitivities of later outputs to the current state, sampled along the trajectories by the adjoint dynamics. On a linear time-invariant system those sensitivities are CA^k, so their covariance is the Stein series of Q_M truncated at the trajectory length, and the two constructions coincide up to the horizon. What Theorem 1 adds is the exact mean-square error of the truncated system and its stability, which follow from the Stein equation and are not stated there. CoBRAS was built for nonlinear systems, where it linearizes along trajectories. On Mamba-2, the readout covariance $\widehat{Q}$ we balance (Appendix A) is that sensitivity covariance at horizon one. CoBRAS reduces one system at a time and has no allocation across systems.

Bittner et al. (2025) applies balanced truncation to each layer after training without retraining. We do not run their allocation procedure, which lowers each layer's order by one state at a time and re-evaluates labelled accuracy after each step, requiring one pass over a labelled evaluation set per state removed. On SaShiMi we estimate that one run at one threshold takes more than six GPU-days, against 418 s for `POISE`. It also needs labels, which `POISE` does not.

Sakamoto & Sato (2025a) reduces each layer of a deep SSM with linear quadratic-output layers after training, without retraining. It minimizes a finite-horizon $\mathcal{H}_2$ error by gradient descent, with per-layer orders selected manually. No code is released, so we discuss the method but do not evaluate it.

D PROOFS OF THE RESULTS IN THE MAIN BODY

This appendix proves Theorem 1 in §D.1, Proposition 1 in §D.2, and Proposition 2 in §D.4. Section D.3 gives the concentration bound for η used in Proposition 1. Section D.6 proves the exactness of the single sort in the spectral variant of Algorithm 1. The invariance of single-output truncations to the output metric is proved in §E.1. Unless stated otherwise, systems are expressed in the balanced coordinates of §3.1 and partitioned as there.

D.1 PROOF OF THEOREM 1

Proof. Part (i): stability. A congruence preserves the weighted Stein equation: with $A \leftarrow S^{-1}AS$, $C \leftarrow CS$ and $Q_M \leftarrow S^\top Q_M S$, the left-hand side of the Stein equation defining Q_M in §3.1 becomes $S^\top(A^\top Q_M A - Q_M + C^\top MC)S = 0$. By §3.1 the transformed Q_M is Λ, so

$$A^\top \Lambda A - \Lambda + C^\top MC = 0,$$

whose $(1,1)$ block reads

$$\Lambda_1 = A_{11}^\top \Lambda_1 A_{11} + A_{21}^\top \Lambda_2 A_{21} + C_1^\top MC_1. \tag{13}$$

Write $X := A_{21}^\top \Lambda_2 A_{21} + C_1^\top MC_1$, so that (13) reads $\Lambda_1 = A_{11}^\top \Lambda_1 A_{11} + X$. Here $\Lambda_1 \succ 0$, since its entries are the $\sigma_i = \sqrt{\lambda_i(PQ_M)} > 0$ with $P, Q_M \succ 0$, and $X \succeq 0$, since $\Lambda_2 \succeq 0$ and $M \succ 0$. Normalize by Λ_1: put

$$Y := \Lambda_1^{-1/2} X \Lambda_1^{-1/2}, \qquad \hat{A} := \Lambda_1^{1/2} A_{11} \Lambda_1^{-1/2},$$

and multiply (13) by $\Lambda_1^{-1/2}$ on both sides:

$$I = \hat{A}^\top \hat{A} + Y. \tag{14}$$

Define $\gamma_r := \lambda_{\min}(Y)$. Two facts follow from (14). *First*, $0 \le \gamma_r \le 1$: $Y \succeq 0$ gives $\gamma_r \ge 0$, and $Y = I - \hat{A}^\top \hat{A} \preceq I$, because $\hat{A}^\top \hat{A} \succeq 0$, gives $\lambda_{\min}(Y) \le \lambda_{\max}(Y) \le 1$. *Second*, $Y \succeq \gamma_r I$, since every eigenvalue of the symmetric Y is at least its smallest one, so (14) gives

$$\hat{A}^\top \hat{A} = I - Y \preceq (1-\gamma_r)\, I, \qquad \text{that is} \qquad \|\hat{A}\|_2 \le \sqrt{1-\gamma_r}.$$

Undoing the normalization, this is $A_{11}^\top \Lambda_1 A_{11} \preceq (1-\gamma_r)\Lambda_1$. Since $\hat{A}$ is similar to A_{11} and the spectral radius is at most any induced norm,

$$\|\hat{A}^k\|_2 \le \|\hat{A}\|_2^k \le (1-\gamma_r)^{k/2}, \qquad \rho(A_{11}) = \rho(\hat{A}) \le \|\hat{A}\|_2 \le \sqrt{1-\gamma_r}.$$

For strict stability without assuming $\gamma_r > 0$, let $A_{11}v = \lambda v$ with $v \neq 0$, and suppose $|\lambda| \ge 1$. Multiplying (13) by v^* on the left and by v on the right gives

$$(|\lambda|^2 - 1)\, v^* \Lambda_1 v + \left\|\Lambda_2^{1/2} A_{21} v\right\|^2 + \left\|M^{1/2} C_1 v\right\|^2 = 0,$$

a sum of nonnegative terms, so all three vanish. As $\Lambda_1 \succ 0$, this forces $|\lambda| = 1$ and $A_{21}v = 0$, so that

$$A \begin{bmatrix} v \\ 0 \end{bmatrix} = \begin{bmatrix} A_{11}v \\ A_{21}v \end{bmatrix} = \lambda \begin{bmatrix} v \\ 0 \end{bmatrix},$$

and λ is an eigenvalue of A of modulus one, contradicting the stability of A. Only $A_{21}v = 0$ is used, not $M^{1/2}C_1 v = 0$, which is why the argument does not depend on M.

Part (ii): the error identity. Both systems are driven by the same input, which cancels from the error of the retained coordinates, so with $\xi_0 = 0$,

$$\xi_{k+1} = A_{11}\xi_k + A_{12}\nu_k, \qquad \boldsymbol{\delta}_k := y_k - \hat{y}_k = C_1\xi_k + C_2\nu_k. \tag{15}$$

In passing, these are the state and output equations of a system driven by the discarded coordinates, so the error is the response $\boldsymbol{\delta} = H_r \star \nu$ of the filter $H_r(z) := C_2 + C_1(zI - A_{11})^{-1}A_{12}$; the proof below does not use this form, and §4.2 does. Write $V(\xi) := \xi^\top \Lambda_1 \xi \ge 0$ for the storage of the retained block, and stack the error state as $\zeta_k := [\,\xi_k^\top \;\nu_k^\top\,]^\top$. Then w_k is nothing but the discarded block of $A\zeta_k$:

$$A\zeta_k = \begin{bmatrix} \xi_{k+1} \\ w_k \end{bmatrix}, \qquad C\zeta_k = \boldsymbol{\delta}_k.$$

Evaluate the quadratic form of $A^\top \Lambda A - \Lambda + C^\top M C = 0$ at ζ_k, term by term. Since $\Lambda = \mathrm{diag}(\Lambda_1, \Lambda_2)$ is block diagonal and $A\zeta_k = [\,\xi_{k+1}^\top \; w_k^\top\,]^\top$,

$$\zeta_k^\top A^\top \Lambda A \zeta_k = (A\zeta_k)^\top \Lambda (A\zeta_k) = V(\xi_{k+1}) + w_k^\top \Lambda_2 w_k, \qquad \zeta_k^\top \Lambda \zeta_k = V(\xi_k) + \nu_k^\top \Lambda_2 \nu_k,$$

and $\zeta_k^\top C^\top M C \zeta_k = (C\zeta_k)^\top M (C\zeta_k) = \boldsymbol{\delta}_k^\top M \boldsymbol{\delta}_k$. Adding the three with the signs of the Stein equation gives 0, that is, pathwise and for every k,

$$\boldsymbol{\delta}_k^\top M \boldsymbol{\delta}_k + V(\xi_{k+1}) - V(\xi_k) = \nu_k^\top \Lambda_2 \nu_k - w_k^\top \Lambda_2 w_k. \tag{16}$$

Summing over $k = 0, \dots, T-1$ and taking expectations telescopes the storage,

$$\mathbb{E}\|\mathbf{y}_T - \hat{\mathbf{y}}_T\|_M^2 = \frac{1}{T}\sum_{k<T} \mathbb{E}\big[\nu_k^\top \Lambda_2 \nu_k\big] - \frac{1}{T}\sum_{k<T} \mathbb{E}\big[w_k^\top \Lambda_2 w_k\big] - \frac{1}{T}\mathbb{E}\big[\xi_T^\top \Lambda_1 \xi_T\big],$$

since $\xi_0 = 0$, hence $V(\xi_0) = 0$, and $V(\xi_T) = \xi_T^\top \Lambda_1 \xi_T$. This is the only step at which the deployed input enters. The discarded coordinates ν_k are the trailing block of $S^{-1}x_k$, with x_k in the original coordinates, so

$$\frac{1}{T}\sum_{k<T} \mathbb{E}\big[\nu_k \nu_k^\top\big] = \big(S^{-1} P_{\mathbf{u}_T} S^{-\top}\big)_{22} = \Pi_{22},$$

and the first term equals $\mathrm{tr}(\Lambda_2 \Pi_{22})$. Note that S is built from the design matrix P while Π is built from the deployment covariance $P_{\mathbf{u}_T}$; the two coincide only at the minimiser of part *(iii)*, where $\Pi = \Lambda$. The inequality in (3) follows from $\Lambda_1, \Lambda_2 \succeq 0$. Nothing so far relates P to $P_{\mathbf{u}_T}$, so parts *(i)* and *(ii)* hold for every $P \succ 0$.

Part (iii): the minimising P. Write $W := Q_M^{1/2}$, $H := W P_{\mathbf{u}_T}^{1/2}$, and let $\mathcal{V} := \mathrm{range}(S_1)$ be the subspace the truncation keeps. We show that the bound sees P only through $\mathcal{V}$, and then choose $\mathcal{V}$.

Balancing gives $S^\top Q_M S = \Lambda$, hence $S^{-1} = \Lambda^{-1} S^\top Q_M$ and the lower block of S^{-1} is $\Lambda_2^{-1} S_2^\top Q_M$. Since $\Pi = S^{-1} P_{\mathbf{u}_T} S^{-\top}$, its $(2,2)$ block is therefore

$$\Pi_{22} = \Lambda_2^{-1} S_2^\top Q_M \, P_{\mathbf{u}_T} \, Q_M S_2 \Lambda_2^{-1}, \qquad \text{so} \qquad \mathrm{tr}\big(\Lambda_2 \Pi_{22}\big) = \mathrm{tr}\big(\Lambda_2^{-1} S_2^\top Q_M P_{\mathbf{u}_T} Q_M S_2\big)$$

by cyclicity of the trace. Now split $Q_M = WW$ on both sides and $P_{\mathbf{u}_T} = P_{\mathbf{u}_T}^{1/2} P_{\mathbf{u}_T}^{1/2}$ in the middle, and read the factors off: with $N := W S_2 \Lambda_2^{-1} S_2^\top W$,

$$\mathrm{tr}\big(\Lambda_2^{-1} S_2^\top Q_M P_{\mathbf{u}_T} Q_M S_2\big) = \mathrm{tr}\Big(\big(W P_{\mathbf{u}_T}^{1/2}\big)^\top \big[W S_2 \Lambda_2^{-1} S_2^\top W\big] \big(W P_{\mathbf{u}_T}^{1/2}\big)\Big) = \mathrm{tr}\big(H^\top N H\big),$$

again by cyclicity. Since N is symmetric and idempotent, $\mathrm{tr}(H^\top N H) = \mathrm{tr}(H^\top N^\top N H)$, so

$$\varepsilon(r, M, P)^2 = \mathrm{tr}\big(\Lambda_2 \Pi_{22}\big) = \|NH\|_F^2. \tag{17}$$

Here N is the orthogonal projector onto $(W\mathcal{V})^\perp$: the columns of $U_2 := W S_2 \Lambda_2^{-1/2}$ are orthonormal, since $S_2^\top Q_M S_2 = \Lambda_2$, and orthogonal to those of WS_1, since $S_1^\top Q_M S_2 = 0$, and $N = U_2 U_2^\top$, one $\Lambda_2^{-1/2}$ from each factor. In (17) the design matrix appears in N alone: H is built from M and the deployment covariance only. So the bound is the energy of H outside $W\mathcal{V}$, and all P does is choose that r-dimensional subspace.

Whatever P is, N is an orthogonal projector of rank $n - r$, so Eckart–Young bounds the whole family at once:

$$\varepsilon(r, M, P)^2 = \|NH\|_F^2 \;\geq\; \sum_{i>r} \sigma_i(H)^2.$$

The singular values of H are the $\sigma_i(M, P_{\mathbf{u}_T})$: $HH^\top = W P_{\mathbf{u}_T} W$ and $P_{\mathbf{u}_T} Q_M = (P_{\mathbf{u}_T} W) W$ are the two products AB and BA of $A = W$ and $B = P_{\mathbf{u}_T} W$, which share their eigenvalues. The bound is attained at $P = P_{\mathbf{u}_T}$, which is admissible and makes $\Pi = \Lambda$, so that $\varepsilon(r, M, P_{\mathbf{u}_T})^2 = \mathrm{tr}(\Lambda_2^2) = \sum_{i>r} \sigma_i(M, P_{\mathbf{u}_T})^2$. This is (4). □

D.2 Proof of Proposition 1

Proof. Write $\widehat{\sigma}_i := \sigma_i(M, \widehat{P})$, and let $\widehat{S}$, $\widehat{\Lambda}$ and $\Pi = \widehat{S}^{-1} P_{\mathbf{u}_T} \widehat{S}^{-\top}$ be those of the balancing at $\widehat{P}$. Let $1 + \varrho := \lambda_{\max}\big(\widehat{P}^{-1/2} P_{\mathbf{u}_T} \widehat{P}^{-1/2}\big)$, that is, the smallest constant with

$$P_{\mathbf{u}_T} \preceq (1 + \varrho)\, \widehat{P}.$$

A congruence preserves the Loewner order, so applying $X \mapsto \widehat{S}^{-1} X \widehat{S}^{-\top}$ to both sides, and using $\widehat{S}^{-1} \widehat{P} \widehat{S}^{-\top} = \widehat{\Lambda}$, gives $\Pi \preceq (1 + \varrho)\widehat{\Lambda}$. A principal submatrix inherits the ordering, so $\Pi_{22} \preceq (1 + \varrho)\widehat{\Lambda}_2$, and the trace against $\widehat{\Lambda}_2 \succeq 0$ is monotone:

$$\varepsilon(r, M, \widehat{P})^2 = \operatorname{tr}\big(\widehat{\Lambda}_2 \Pi_{22}\big) \;\le\; (1 + \varrho)\, \operatorname{tr}\big(\widehat{\Lambda}_2^2\big) = (1 + \varrho) \sum_{i>r} \widehat{\sigma}_i^2.$$

The constant is sharp: at $\widehat{P} = P_{\mathbf{u}_T}/(1 + \varrho)$ one has $\Pi = (1 + \varrho)\widehat{\Lambda}$ and the inequality is an equality.

It remains to read ϱ off the assumption. The left half of the sandwich $(1 - \eta)P_{\mathbf{u}_T} \preceq \widehat{P} \preceq (1 + \eta)P_{\mathbf{u}_T}$ gives $P_{\mathbf{u}_T} \preceq \widehat{P}/(1 - \eta)$, hence $1 + \varrho \le 1/(1 - \eta)$, which is the first inequality of (5); it also gives $\widehat{P} \succeq (1 - \eta)P_{\mathbf{u}_T} \succ 0$. For the second, $\sigma_i(M, P)^2 = \lambda_i(Q_M^{1/2} P Q_M^{1/2})$ is monotone in P, so the sandwich gives $\widehat{\sigma}_i^2 \le (1 + \eta)\, \sigma_i(M, P_{\mathbf{u}_T})^2$ for every i, and with (4),

$$\varepsilon(r, M, \widehat{P})^2 \;\le\; \frac{1}{1 - \eta} \sum_{i>r} \widehat{\sigma}_i^2 \;\le\; \frac{1 + \eta}{1 - \eta} \sum_{i>r} \sigma_i(M, P_{\mathbf{u}_T})^2 = \frac{1 + \eta}{1 - \eta}\, \varepsilon(r, M, P_{\mathbf{u}_T})^2.$$

□

D.3 Concentration of the plug-in covariance

Every certificate in the paper carries the factor $1/(1 - \eta)$ of Proposition 1, and η is the only statistical quantity in them: it compares $\widehat{P}$ with the covariance $P_{\mathbf{u}_T}$ it estimates and involves neither the order nor the allocation. Let the calibration set be N independent input sequences $\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(N)}$ drawn from $\mathcal{D}$, and write $Z^{(m)} := \frac{1}{T} \sum_{k<T} x_k^{(m)} x_k^{(m)\top}$ for the time-averaged state second moment of the m-th, so that $\widehat{P} = \frac{1}{N} \sum_m Z^{(m)}$ and $\mathbb{E} Z^{(m)} = P_{\mathbf{u}_T}$. The time steps within a sequence are dependent; the sequences are not, and that is all the argument uses. Throughout this subsection η is the smallest number admissible in Proposition 1, that is $\eta := \big\| P_{\mathbf{u}_T}^{-1/2} \widehat{P} P_{\mathbf{u}_T}^{-1/2} - I \big\|$, since $(1 - \eta)P_{\mathbf{u}_T} \preceq \widehat{P} \preceq (1 + \eta)P_{\mathbf{u}_T}$ is that inequality written out.

Proposition 3 (Concentration of η)**.** *Let $P_{\mathbf{u}_T} \succ 0$ and let $L < \infty$ be the smallest constant with $\big\| P_{\mathbf{u}_T}^{-1/2} Z^{(m)} P_{\mathbf{u}_T}^{-1/2} \big\| \le L$ almost surely, which is finite whenever the inputs are bounded. Then $L \ge 1$, and for every $\delta \in (0, 1)$, with probability at least $1 - \delta$,*

$$\eta \;\le\; \sqrt{\frac{2L \log(2n/\delta)}{N}} \;+\; \frac{2L \log(2n/\delta)}{3N}. \tag{18}$$

In particular, if $N \ge 4L \log(2n/\delta)$, then $\eta \le 7/8$ with probability at least $1 - \delta$: $\widehat{P}$ is admissible and (5) *holds at every order and for every allocation at once, including the allocation selected from the same calibration set.*

Proof. Put $Y^{(m)} := P_{\mathbf{u}_T}^{-1/2} Z^{(m)} P_{\mathbf{u}_T}^{-1/2} \succeq 0$: independent, with $\mathbb{E} Y^{(m)} = I$ and $\|Y^{(m)}\| \le L$ almost surely. Then $L \ge 1$: $\lambda_{\max}$ is convex, being a supremum of linear maps $Y \mapsto v^\top Y v$, so Jensen's inequality and the almost-sure bound give $1 = \lambda_{\max}(\mathbb{E} Y^{(m)}) \le \mathbb{E} \lambda_{\max}(Y^{(m)}) \le L$. Since $\widehat{P} = \frac{1}{N} \sum_m Z^{(m)}$, whitening by $P_{\mathbf{u}_T}$ turns the definition of η into a deviation of an average of independent matrices from its mean,

$$\eta = \Big\| \sum_{m=1}^{N} X_m \Big\|, \qquad X_m := \frac{Y^{(m)} - I}{N},$$

with the X_m independent, centred, symmetric and identically distributed. Matrix Bernstein needs two quantities. *The almost-sure bound:* the eigenvalues of $Y^{(m)}$ lie in $[0, L]$, so those of $Y^{(m)} - I$ lie in $[-1, L-1]$ and

$$\|X_m\| \;\le\; \frac{\max(L-1,1)}{N} \;\le\; \frac{L}{N},$$

using $L \ge 1$. *The variance:* the X_m being identically distributed, the sum of their second moments is N times one of them, and $\mathbb{E}Y^{(m)} = I$ leaves

$$\sum_{m=1}^{N} \mathbb{E}X_m^2 = \frac{1}{N^2}\sum_{m=1}^{N} \mathbb{E}\big(Y^{(m)} - I\big)^2 = \frac{\mathbb{E}\big(Y^{(1)}\big)^2 - I}{N} \;\preceq\; \frac{L-1}{N}\, I \;\preceq\; \frac{L}{N}\, I,$$

because $0 \preceq Y \preceq LI$ gives $Y^2 = Y^{1/2} Y Y^{1/2} \preceq LY$, hence $\mathbb{E}Y^2 \preceq L\,\mathbb{E}Y = LI$. The matrix Bernstein inequality (Tropp, 2012), with these two quantities, then gives

$$\Pr\Big(\Big\|\sum_m X_m\Big\| \ge t\Big) \;\le\; 2n \exp\Big(-\frac{t^2/2}{L/N + Lt/(3N)}\Big).$$

Setting the right-hand side to δ, the positive root of the resulting quadratic in t is at most the right-hand side of (18), by $\sqrt{a+b} \le \sqrt{a} + \sqrt{b}$. If $N \ge 4L\log(2n/\delta)$ the two terms are at most $\sqrt{1/2}$ and $1/6$, whose sum is below $7/8$. The bound holds for every allocation at once because η does not depend on r. □

D.4 Proof of Proposition 2

The proof assumes that the network is continuously differentiable in the outputs of its SSMs, as every architecture here is. The operators Γ_{jk} come from the mean value theorem, so they depend on the errors themselves: part (i) is an exact identity, not a fixed quadratic form.

Proof. Part (i). Fix $(\mathbf{r}, \mathcal{P}, \mathcal{M})$ and $\mathbf{u}_T$. For $j = 1, \dots, J+1$, let H_j be the output of the network in which systems $j, \dots, J$ are reduced and systems $1, \dots, j-1$ are kept at full order, so that $H_1 = \hat{F}(\mathbf{u}_T; \mathbf{r}, \mathcal{P}, \mathcal{M})$ and $H_{J+1} = F(\mathbf{u}_T)$, and put $\Delta_j := H_{j+1} - H_j$. The networks H_j and H_{j+1} differ only in system j, whose input is the teacher-forced $\mathbf{u}_{j,T}$ since everything upstream is at full order. Write $a_j := \hat{\mathbf{y}}_{j,T}(\mathbf{u}_{j,T}, P_j, M_j)$, so that $\mathbf{y}_{j,T}(\mathbf{u}_{j,T}) = a_j + \boldsymbol{\delta}_j$, and let $\mathcal{T}_j$ map the output trajectory of system j to the output of the network whose systems $j+1, \dots, J$ are reduced; $\mathcal{T}_j$ is C^1 on the architectures considered. The mean value theorem in integral form gives

$$\Delta_j = \mathcal{T}_j\big(a_j + \boldsymbol{\delta}_j\big) - \mathcal{T}_j(a_j) = D_j \boldsymbol{\delta}_j, \qquad D_j := \int_0^1 D\mathcal{T}_j\big(a_j + s\boldsymbol{\delta}_j\big)\, ds,$$

and the differences telescope:

$$F(\mathbf{u}_T) - \hat{F}(\mathbf{u}_T; \mathbf{r}, \mathcal{P}, \mathcal{M}) = \sum_{j=1}^{J} \Delta_j = \sum_{j=1}^{J} D_j \boldsymbol{\delta}_j.$$

D_j depends on the reduced systems downstream through $\mathcal{T}_j$, and on (r_j, P_j, M_j) through a_j.

Write $z := F(\mathbf{u}_T)$ and $\hat{z} := \hat{F}(\mathbf{u}_T; \mathbf{r}, \mathcal{P}, \mathcal{M})$. Taylor's theorem with integral remainder, applied to $t \mapsto \psi\big(z + t(\hat{z} - z)\big)$, gives

$$B_\psi(\hat{z}, z) = \tfrac{1}{2}(\hat{z} - z)^\top W (\hat{z} - z), \qquad W := 2\int_0^1 (1-t)\, \nabla^2 \psi\big((1-t)z + t\hat{z}\big)\, dt \;\succeq\; 0,$$

W being positive semidefinite because ψ is convex, and depending on ψ and on the whole of $(\mathbf{r}, \mathcal{P}, \mathcal{M})$ through $\hat{z}$. Substituting $\hat{z} - z = -\sum_j D_j \boldsymbol{\delta}_j$ and expanding the square,

$$B_\psi(\hat{z}, z) = \frac{1}{2}\sum_{j,k} \big\langle \boldsymbol{\delta}_j, \boldsymbol{\delta}_k \big\rangle_{\Gamma_{jk}}, \qquad \Gamma_{jk} := T\, D_j^\top W D_k,$$

the factor T being the time average carried by $\langle \cdot, \cdot \rangle_\Gamma$. Hence $\Gamma_{jk}^\top = \Gamma_{kj}$ and $(\Gamma_{jk})_{j,k} = T\, \mathbf{D}^\top W \mathbf{D} \succeq 0$ with $\mathbf{D} := (D_1, \dots, D_J)$; the maps Γ_{jk} depend on ψ through W. Taking the expectation over $\mathbf{u}_T \sim \mathcal{D}$ gives the identity of part *(i)*.

Part (ii). At $\mathbf{n}_{-j}(r_j)$ every system $k \neq j$ is the original one, so $\boldsymbol{\delta}_k = 0$ and only the term $j = k$ survives in the double sum of part *(i)*, with $\Gamma_{jj} = T\, D_j^\top W D_j$ evaluated at that allocation: there $\mathcal{T}_j$ is the map through the uncompressed downstream systems and $\hat{z}$ is the output of the network with system j alone reduced. This is (7).

The log-sum-exp case. For ψ the log-sum-exp, $B_\psi(\hat{z}, z) = \mathrm{KL}(p \,\|\, \hat{p})$ with $p = \mathrm{softmax}(z)$ and $\hat{p} = \mathrm{softmax}(\hat{z})$, since the Bregman divergence of the log-partition function is the Kullback–Leibler divergence between the distributions it generates. At $\mathbf{n}_{-j}(r_j)$, $\hat{p} = \hat{p}_{-j}$, and taking the expectation over $\mathbf{u}_T \sim \mathcal{D}$ in the definition of $\mathcal{L}_\psi$ gives (8). □

D.5 Dynamic programming for the one-at-a-time criterion

At $\mathbf{n}_{-j}(r_j)$ the network is the original one with system j alone replaced by its order-r_j reduction, so the j-th term of (9) is a function of (r_j, P_j, M_j), and a function $f_j(r_j)$ of r_j alone when P_j and M_j do not depend on the other orders. The objective ϕ of §4.2 is separable in the same way, with $f_j = c_j^\pi E_j$, and the exact variant of Algorithm 1 solves it by the recursion below.

Let $V_j(b)$ be the minimum of $\sum_{k \le j} f_k(r_k)$ subject to $\sum_{k \le j} r_k \le b$ and $0 \le r_k \le n_k$. Since the terms are separable, $V_j(b) = \min_{0 \le r \le \min(n_j, b)} \{V_{j-1}(b - r) + f_j(r)\}$, with $V_0(b) = 0$; the recursion is exact, and each of its J stages costs $O(R \max_j n_j)$. This is the allocation ZeroQ solves for bit-widths, its §III-B and equation (4), also by dynamic programming (Cai et al., 2020).

D.6 A single sort minimizes the weighted tail

Proposition 4 (Exactness of the sort)**.** *Let $c \in \mathbb{R}^J$ with $c_j \ge 0$, let $\sigma_{j,1} \ge \sigma_{j,2} \ge \cdots \ge \sigma_{j,n_j} \ge 0$ for every j, and let $0 \le R \le \sum_j n_j$. Consider*

$$\min_{\mathbf{r}} \ \phi_c(\mathbf{r}) = \sum_{j=1}^{J} c_j \sum_{i > r_j} \sigma_{j,i}^2 \qquad \text{s.t.} \qquad \sum_{j=1}^{J} r_j \le R, \quad 0 \le r_j \le n_j. \tag{19}$$

Order the pairs (j, i) with $1 \le i \le n_j$ by decreasing $g_{j,i} := c_j \sigma_{j,i}^2$, breaking ties by the smallest (j, i) in lexicographic order; let $\mathcal{S}$ be the first R pairs and $r_j^\star := \big|\{i : (j, i) \in \mathcal{S}\}\big|$. Then $\mathbf{r}^\star$ minimizes (19)*, it spends the whole budget, $\sum_j r_j^\star = R$, and the pairs of system j in $\mathcal{S}$ are exactly $(j, 1), \ldots, (j, r_j^\star)$.*

Proof. For every feasible $\mathbf{r}$,

$$\phi_c(\mathbf{r}) = \sum_{j=1}^{J} \sum_{i=1}^{n_j} g_{j,i} \ - \ G(\mathbf{r}), \qquad G(\mathbf{r}) := \sum_{j=1}^{J} \sum_{i=1}^{r_j} g_{j,i},$$

and the first term does not depend on $\mathbf{r}$, so minimising ϕ_c is maximising G. $G(\mathbf{r})$ is the sum of g over the set $S(\mathbf{r}) := \{(j, i) : 1 \le i \le r_j\}$, which has $\sum_j r_j \le R$ elements. Since every $g_{j,i} \ge 0$, the sum over any set of at most R pairs is at most the sum of the R largest values, which is the sum over $\mathcal{S}$:

$$G(\mathbf{r}) \ \le \ \sum_{(j,i) \in \mathcal{S}} g_{j,i} \qquad \text{for every feasible } \mathbf{r}.$$

It remains to show that $\mathcal{S} = S(\mathbf{r}^\star)$, that is, that $\mathcal{S}$ is a prefix in every system. Let $(j, i) \in \mathcal{S}$ with $i \ge 2$. Then $g_{j,i-1} \ge g_{j,i}$, because $c_j \ge 0$ and $\sigma_{j,\cdot}$ is nonincreasing, and $(j, i-1)$ precedes (j, i) lexicographically, so $(j, i-1)$ comes before (j, i) in the order that defines $\mathcal{S}$ and belongs to $\mathcal{S}$ too. Hence the pairs of system j in $\mathcal{S}$ are $(j, 1), \ldots, (j, r_j^\star)$, so $\mathcal{S} = S(\mathbf{r}^\star)$. Then $0 \le r_j^\star \le n_j$, $\sum_j r_j^\star = R$, and $G(\mathbf{r}^\star)$ attains the bound above, so $\mathbf{r}^\star$ is a minimizer. □

Only the order of the $g_{j,i}$ matters, so multiplying every c_j by one positive constant leaves $\mathbf{r}^\star$ unchanged. The argument needs c_j to be a constant per system: with an order-dependent rate $c_j(r_j)$ the terms $g_{j,i}$ are no longer fixed, the selected set need not be a prefix, and (9) needs dynamic programming instead.

E THE CHOICE OF THE OUTPUT METRIC M_j

Theorem 1 holds for every $M \succ 0$, allowing the output error to be measured in a norm determined by the rest of the network. Proposition 2 identifies Γ_j as the corresponding loss metric. Since Γ_j is not static, it cannot be used directly as M_j. We use $M_j = I$ and examine this choice below. For systems with one output, the metric does not change the retained subspace at a fixed order. For systems with several outputs, it can change the subspace, so we compare three metric shapes on S5.

E.1 A SCALAR METRIC LEAVES THE TRUNCATION UNCHANGED

Proposition 5 (A multiple of the identity does not change the truncation)**.** *Let $M = \mu I$ with $\mu > 0$ and $P \succ 0$. For every order r, the truncations of Algorithm 2 at $M = \mu I$ and $M = I$ have the same transfer function. Consequently, the reduced system's input-output behaviour, its error $\boldsymbol{\delta}_j$ and the loss $\mathcal{L}_\psi$ are independent of μ.*

Proof. Let Q_μ denote Q_M at $M = \mu I$. The series representation in §3.1 gives

$$Q_\mu = \sum_{t \geq 0} (A^t)^\top C^\top (\mu I) C A^t = \mu Q_1.$$

Its Cholesky factor is therefore $L_\mu = \sqrt{\mu}\, L_1$. Given the singular value decomposition $L_1^\top R = U \Lambda_1 V^\top$ at $\mu = 1$, we have

$$L_\mu^\top R = U \big(\sqrt{\mu}\, \Lambda_1\big) V^\top.$$

Thus, $\Lambda_\mu = \sqrt{\mu}\, \Lambda_1$ and $\sigma_i(\mu I, P) = \sqrt{\mu}\, \sigma_i(I, P)$ for every i. The balancing transformation satisfies

$$S_\mu = R V \Lambda_\mu^{-1/2} = \mu^{-1/4} S_1,$$

so the balanced realizations obey

$$S_\mu^{-1} A S_\mu = S_1^{-1} A S_1, \qquad S_\mu^{-1} B = \mu^{1/4} S_1^{-1} B, \qquad C S_\mu = \mu^{-1/4} C S_1.$$

The leading blocks satisfy the same relations. The factors $\mu^{1/4}$ and $\mu^{-1/4}$ cancel in the reduced transfer function $\bar{C}_1 (zI - \bar{A}_{11})^{-1} \bar{B}_1$, which is therefore identical at $M = \mu I$ and $M = I$. The two truncations produce the same output for every input and hence the same error $\boldsymbol{\delta}_j$. With the other systems unchanged, they also give the same loss $\mathcal{L}_\psi$. For repeated singular values, the argument holds for any fixed choice of singular vectors at $\mu = 1$, used consistently in both constructions. □

When the system has one output ($p = 1$), every positive-definite metric is a positive scalar multiple of the identity. Proposition 5 therefore covers all metric choices at a fixed order. This applies to the four-layer S4D classifier and SaShiMi, where each system is a single channel.

E.2 METRIC SHAPES ON S5

In S5, each system is a whole layer with several outputs, so the metric can change the retained subspace. We compared three metric shapes on S5, LRA-Text. These are the identity, a diagonal matrix formed from the layer's empirical output covariance, and a diagonal matrix whose entries are the squared column norms of the next layer's input map.

To compare shapes at a fixed scale, we normalize each metric to have a mean diagonal entry of one. This matters because replacing I with μI multiplies the layer's tail by μ, as follows from Proposition 5. Without normalization, this scalar would also change the layer's weight in the pooled sort. The normalization leaves $M = I$ unchanged, making it the control for the $c \equiv 1$ allocation used here.

On one checkpoint, the three metrics give accuracies of 86.21, 86.14 and 86.22 at budget 0.05, and 88.15, 88.08 and 88.09 at budget 0.10, respectively. The differences are below 0.1 percentage points at either budget. The third metric gives exactly the same allocation as the identity, while the second reallocates a few of the 58 and 115 retained states. These observations are limited to this checkpoint and these budgets. We have not tested other metric shapes or other tasks.