

MaRK: Markov-adapted Recurrent Kernels for Dynamic Operator Conditioning in State Space Models

Syed Ibrahim Omer*

Department of Data Science
City University of Hong Kong

Ginny Y. Wong

NVIDIA AI Technology Center
NVIDIA, Santa Clara, USA

Xiangyu Zhao

Department of Data Science
City University of Hong Kong

Abstract

State Space Models (SSMs) offer an efficient alternative to Transformers for sequence modeling, yet conditioning pre-trained SSMs for iterative generation typically operates outside the recurrent operator, through input injection or activation modulation. While such mechanisms expose the model to conditioning information, they leave the underlying temporal dynamics fixed. We introduce MaRK (Markov-adapted Recurrent Kernels), a dynamic operator-conditioning framework that maps context vectors directly into bounded modulations of a frozen SSM’s recurrence (A), read-in (B), read-out (C), skip (D), and discretization (Δ) parameters. Viewed through the lens of Linear Parameter-Varying systems, MaRK induces a context-indexed family of Markov parameter sequences, allowing each diffusion timestep to reshape the model’s input-output memory kernel. We instantiate MaRK on a frozen 111M-parameter Hydra SSM backbone and study three adapter geometries: Hypernet, Chebyshev polynomial, and Discrete Cosine Transform kernels. Since these adapters modify the Markov parameter sequence through low-rank auxiliary maps on the frozen backbone, parameter-efficient fine-tuning arises as a structural consequence of the adaptation mechanism itself, requiring only 6.3–11M trainable auxiliary parameters to transition from a bidirectional objective to an iterative diffusion regime. The bounded recurrence parameterization further yields an analytic Affine Quadratic Stability certificate for the modulated recurrence. Through synthetic LPV recovery experiments and Markov-operator diagnostics, we show that MaRK recovers coordinate-invariant temporal operators under matched assumptions and produces distinct, stable timestep-conditioned memory profiles. Empirically, the Chebyshev variant yields the strongest performance, achieving an average validation loss of 2.55, followed by the DCT (2.59) and Hypernet (3.77) geometries. Together, these results provide initial evidence that dynamic operator modulation is a principled operator-level conditioning mechanism for adapting SSMs beyond input-stream injection and adaptive normalization.

1 Introduction

State Space Models (SSMs) have emerged as highly efficient, continuous-time alternatives to Transformers for large-scale sequence modeling [1–3]. However, adapting sequence models for conditional generation tasks (e.g., discrete diffusion) traditionally relies on post-hoc interventions [4–8]. This challenge is commonly addressed by two external conditioning paradigms: **Input-Stream Injection** [4] and **Adaptive Layer Normalization (AdaLN)** [9–11]. Input-stream injection concatenates

*Corresponding Author: ibrahsyed2-c@my.cityu.edu.hk

conditioning vectors directly into the sequence; while computationally cheap, it demands that the state-space dynamics route this context across the entire sequence length. For continuous-time SSMs, this relies on learned hidden states carrying context forward, which can dilute the prompt signal and induce temporal lag. Conversely, AdaLN dynamically modulates structural layer activations. While effective for Transformers, applied to SSMs, AdaLN operates externally to the core continuous-time sequence operator. It scales layer inputs and outputs but does not explicitly parameterize the underlying temporal dynamics (the A , B , C , and Δ matrices) as functions of the timestep context. Both paradigms condition the computation without directly parameterizing the Markov parameter sequence ($\mathcal{H}(c_t)$), the complete characterization of an SSM’s input-output temporal behavior.

To bridge this gap, we introduce **MaRK (Markov-adapted Recurrent Kernels)**, a structurally principled operator-level conditioning approach: **Dynamic Operator Modulation**. Rather than relying only on external activation normalization or input additions, MaRK moves the conditioning map into the temporal recurrence parameters themselves. By projecting the context vector into bounded multi-dimensional shifts of the fundamental continuous-time matrices, it dynamically modulates the internal recurrence (A), read-in (B), read-out (C), skip connections (D), and discretization steps (Δ) of a frozen base model [12]. We apply our method to the Hydra BERT SSM model, a bi-directional extension of the Mamba model parameterized as a quasi-separable matrix mixer [13]. Through MaRK, the continuous-time matrices of the pre-trained Hydra BERT SSM depend on external conditioning signals (diffusion timesteps), allowing the system’s temporal properties to vary dynamically at each step.

Because MaRK relies on bounded continuous modulations via low-rank geometries, the same design yields a parameter-efficient adaptation route in our Hydra instantiation, providing a dynamic alternative to static low-rank adaptation methods [14]. We present three kernel geometries: Hypernet, Chebyshev Polynomial, and Discrete Cosine Transform (DCT) [15] kernels, that map conditioning contexts to scalar shifts and low-rank scales. By bounding parameter modulations via hyperbolic tangent saturation and softplus constraints, MaRK admits Affine Quadratic Stability (AQS) certificates for the modulated recurrence [16]. Applied to a 23-layer Hydra SSM, our adapters use 6.3–11M trainable parameters to condition the frozen 111M base structure, turning a computationally expensive pre-training problem into a targeted fine-tuning regime [13].

We evaluate this direct conditioning approach within an iterative mask-based diffusion framework using stability certificates, synthetic operator recovery, Markov-operator diagnostics, and internal adapter ablations. MaRK specifically adapts the frozen Hydra backbone to an iterative diffusion process, employing a Context-Adaptive Token-Level Noise Rescheduling (CART) objective to provide token-level, context-aware training signal within our iterative refinement framework [8].

Contributions.

- (i) **LPV-SSM Operator Conditioning:** We cast conditional generation as LPV control over SSM operators, directly modulating A , B , C , D , and Δ as an operator-level perspective beyond input injection and AdaLN.
- (ii) **MaRK Kernel Family:** We design three structural adapter geometries (Hypernet, Chebyshev, and DCT) that map conditioning vectors to low-rank modulations.
- (iii) **Stability Certificates and Operator Recovery:** We use bounded parameterization to obtain analytic AQS certificates and study Markov-operator recovery under matched synthetic assumptions.
- (iv) **Markov Operator Adaptation:** We provide evidence that MaRK conditions the Markov parameter sequence, with timestep-specific lag-norm profiles revealing distinct temporal-memory signatures.

2 Linear Parameter Varying (LPV) Foundation

We adapt the base time-invariant State Space Model using the theory of Linear Parameter Varying (LPV) models [16, 17]. Therefore, we view each modulated SSM layer as a continuous-time state-space system:

$$\dot{x}(\tau) = A(\mathbf{c}_t) x(\tau) + B(\mathbf{c}_t) u(\tau) \quad (1)$$

$$y(\tau) = C(\mathbf{c}_t) x(\tau) + D(\mathbf{c}_t) u(\tau) \quad (2)$$

where $x(\tau) \in \mathbb{R}^n$ is the state, $u(\tau) \in \mathbb{R}^m$ the input, $y(\tau) \in \mathbb{R}^m$ the output, n is the state dimension, m the embedding dimension, $\mathbf{c}_t$ is a dynamic conditioning vector tied to the diffusion timestep t obtained using standard DDPM/DiT sinusoidal (Fourier) encoding and two-layer MLP [4, 9], $\dot{x}(\tau)$ is the derivative of the state $x(\tau)$ with respect to continuous internal-time τ (i.e. $\frac{dx}{d\tau}$).

MaRK modulates the continuous-time parameters $Q \in \{A, B, C, D, \Delta\}$ using learned mapping functions $\psi_{\text{scale}}(\cdot)$ and $\psi_{\text{shift}}(\cdot)$. We use the primed notation to denote the modulated parameters $Q' \in \{A', B', C', D', \Delta'\}$.

For diagonalized scalar parameters $\Theta \in \{A_{\log}, \Delta_{\text{bias}}, D\}$ we apply bounded shifts to maintain stability, where β_{Θ} denotes the scalar magnitude of the bounded shifts:

$$\Theta' = \Theta + \beta_{\Theta} \tanh(\psi_{\text{shift}}(\mathbf{c}_t)) \quad (3)$$

For dense projection matrices $\Phi \in \{B, C\}$ we apply bounded scales and shifts, where $\alpha_{\Phi}, \beta_{\Phi}$ denote the scalar magnitude of the bounded scales and shifts respectively. To avoid the curse of dimensionality for large matrices $B, C \in \mathbb{R}^d$ (where $d = n_{\text{groups}} \times d_{\text{state}}$; n_{groups} is the number of SSM groups and d_{state} is the state dimension), modulation is applied via a rank- r factorized projection to these matrices:

$$\Phi' = \Phi \odot [1 + \alpha_{\Phi} \tanh(\psi_{\text{scale}}(\mathbf{c}_t))] + \beta_{\Phi} \tanh(\psi_{\text{shift}}(\mathbf{c}_t)) \quad (4)$$

Remark (Operator vs. Activation Control). While §1 contrasts these paradigms conceptually, the structural distinction is formalized in Eq. (3)–(4). By modulating the infinitesimal generator A and discretization Δ rather than applying scalar affine transforms (AdaLN), MaRK directly parameterizes the system’s eigenvalue distribution and frequency response. This makes the context $\mathbf{c}_t$ explicit in the temporal kernel’s decay properties, rather than treating it only as an external signal bias.

Finally, we obtain the LPV-SSM form from equations (1) and (2):

$$\dot{x}(\tau) = \bar{A}'(\mathbf{c}_t) x(\tau) + \bar{B}'(\mathbf{c}_t) u(\tau) \quad (5)$$

$$y(\tau) = C'(\mathbf{c}_t) x(\tau) + D'(\mathbf{c}_t) u(\tau) \quad (6)$$

where $\bar{A}'$ and $\bar{B}'$ are the discretized versions of the continuous-time parameters A' and B' , obtained via the Zero-Order Hold (ZOH) [12].

The complete input-output behavior of this LPV-SSM is fully characterized by the Markov parameter, where $\mathcal{H}(\mathbf{c}_t)$ is the Markov Parameter Sequence, $h_k(\tau)$ is the k^{th} Markov Parameter, and k is the shift index [18]:

$$\mathcal{H}(\mathbf{c}_t) = \{h_k(\tau) = C'(\mathbf{c}_t) \bar{A}'^k(\mathbf{c}_t) \bar{B}'(\mathbf{c}_t)\}_{k=0}^{\infty} \quad (7)$$

This object is the natural invariant of the system: two SSMs with different internal coordinates but identical Markov sequences are behaviorally indistinguishable. Crucially, by making each matrix in this product condition-dependent, MaRK adapts the full temporal operator at each generation step rather than acting externally on activations or inputs. We term this approach Markov-adapted Recurrent Kernels (MaRK).

2.1 Constructive Stability via Log-Space Parameterization.

The bounded modulations above support a constructive stability argument through the ZOH (Zero-Order Hold) discretization chain. In Hydra’s diagonal SSM, each recurrence eigenvalue $\bar{A}_i$ is produced by the composition:

$$A = -\exp(A'_{\log}) < 0 \quad (8)$$

$$\Delta = \text{softplus}(\Delta_{\text{input}} + \Delta'_{\text{bias}}) > 0 \quad (9)$$

$$\bar{A}_i = \exp(A_i \cdot \Delta_i) \in (0, 1) \quad (10)$$

Equation (8) follows from $\exp(x) > 0$ for all $x \in \mathbb{R}$, hence $-\exp(A'_{\log}) < 0$. Equation (9) holds because $\text{softplus}(x) = \log(1 + e^x) > 0$ for all $x \in \mathbb{R}$. Finally, Equation (10) follows since $A_i < 0$ and $\Delta_i > 0$ imply $A_i \cdot \Delta_i < 0$, and therefore $\bar{A}_i = \exp(A_i \cdot \Delta_i) \in (0, 1)$.

Proposition 2.1 (Constructive AQS stability). *For the MaRK-modulated diagonal SSM, the identity matrix $P = I$ is a valid common Lyapunov function satisfying the AQS condition $\bar{A}(\mathbf{c}_t)^{\top} P \bar{A}(\mathbf{c}_t) - P \preceq -\varepsilon I$ for all admissible conditioning contexts $\mathbf{c}_t$, with stability margin $\varepsilon = 1 - \max_i \bar{A}_i^2 > 0$.*

Proof sketch. Since A is diagonal, the Lyapunov condition reduces element-wise to $\bar{A}_i^2 - 1 < 0$ for each head i . By Equations (8)–(10), every element $\bar{A}_i \in (0, 1)$ for admissible conditioning contexts. Because $\bar{A}_i(\delta)$ is monotonically decreasing in the shift δ (larger shift $\rightarrow$ more negative $A \rightarrow$ more decay), the worst-case spectral radius occurs at $\delta = -\beta_A$, bounded away from 1 by the finite hyper-rectangle of modulations. $\square$

3 Method

3.1 Pretrained Hydra BERT SSM backbone

We instantiate MaRK on top of a pre-trained Hydra BERT backbone, a bidirectional extension of Mamba implemented through a quasiseparable matrix mixer [13]. Hydra adopts the “matrix mixer” view of sequence mixing, in which a sequence mixer is a structured linear map $Y = MX$ on the token sequence. In contrast to ad-hoc bidirectional SSM heuristics that combine separate forward/backward models by addition or concatenation [19, 20], Hydra parameterizes a single mixer matrix whose strictly lower and upper triangles correspond to forward and backward semiseparable (SSM) structure, while the diagonal remains free and acts as a learned residual pathway. Concretely, Hydra’s quasiseparable mixer admits the form (cf. Eq. (3) in [13])

$$m_{ij} = \begin{cases} \vec{\mathbf{c}}_i^\top \vec{\mathbf{A}}_{i:j}^\times \vec{\mathbf{b}}_j & i > j \\ \delta_i & i = j \\ \overleftarrow{\mathbf{c}}_j^\top \overleftarrow{\mathbf{A}}_{j:i}^\times \overleftarrow{\mathbf{b}}_i & i < j \end{cases} \quad (11)$$

where δ_i is a scalar, $b_i, c_i \in \mathbb{R}^{N \times 1}$, and $A_i \in \mathbb{R}^{N \times N}$. Since this matrix features non-zero upper triangle components, it enables bidirectional processing [13].

In our experiments, we use the released Hydra-BERT checkpoint at BERT scale (111M parameters, 23 layers) [13]. Hydra is a natural backbone for our setting because discrete diffusion language modeling is inherently bidirectional (tokens are refined using both left and right context), yet our objective requires timestep-dependent control over the internal temporal operator. MaRK targets exactly the parameters that define Hydra’s SSM dynamics (the continuous-time recurrence and discretization operators: A, B, C, D , and Δ underlying the quasi-separable factors) and makes them functions of the diffusion context $\mathbf{c}_t$. By doing so, we demonstrate the ability of MaRK to also function as a fine-tuning method to convert an existing bidirectional objective into a context-aware diffusion objective. Throughout, we treat the Hydra backbone as frozen and learn only the MaRK adapters, isolating the effect of dynamic operator modulation from changes in the base representation.

3.2 Discrete Diffusion Language Modeling

We evaluate MaRK in the masked discrete diffusion framework used by recent diffusion LLMs [5, 7, 8]. Let $\mathbf{x}_0 \in \mathcal{V}^L$ be a token sequence of length L from the data distribution, and let $\mathcal{M}$ denote an absorbing noise token. The forward process independently corrupts tokens according to a timestep t by masking each position with probability t (equivalently, keeping it unmasked with probability $\alpha(t) = 1 - t$) [7]:

$$q(\mathbf{x}_t | \mathbf{x}_0) = \prod_{i=1}^L q(x_{t,i} | x_{0,i}), \quad q(x_{t,i} = \mathcal{M} | x_{0,i}) = t, \quad q(x_{t,i} = x_{0,i} | x_{0,i}) = 1 - t. \quad (12)$$

The reverse model is a mask predictor, $p_\theta(\mathbf{x}_0 | \mathbf{x}_t, t)$, that predicts all masked positions in parallel using full bidirectional context. Concretely, we parameterize p_θ with the Hydra backbone, and provide the diffusion context (timestep embedding) as the conditioning vector $\mathbf{c}_t$ that drives MaRK operator modulation that conditions p_θ based on the diffusion context.

Following [7, 8], we train by sampling a clean sequence $\mathbf{x}_0$, sampling $t \sim \mathcal{U}(0, 1)$, then drawing $\mathbf{x}_t \sim q(\mathbf{x}_t | \mathbf{x}_0)$, and minimizing the negative log-likelihood on only the masked tokens:

$$\mathcal{L}_{\text{diff}}(\theta) = \mathbb{E}_{\mathbf{x}_0, t, \mathbf{x}_t} \left[w(t) \sum_{i=1}^L \mathbf{1}[x_{t,i} = \mathcal{M}] \left(-\log p_\theta(x_{0,i} | \mathbf{x}_t, t) \right) \right], \quad (13)$$

where $w(t)$ is a timestep-dependent reweighting term derived from the diffusion ELBO for particular schedules [7], and can be further adapted at the token level based on local context informativeness [8]. In our implementation, this context-adaptive reweighting is instantiated by CART (§3.3), which emphasizes masked tokens that are surrounded by more usable (unmasked) context.

3.3 Training Objective

To optimize training, we utilize **Context-Adaptive Token-level Noise Rescheduling (CART)**, following the token-level rescheduling paradigm introduced in Dream [8]. Rather than using a single global timestep-dependent weight $w(t)$ for all masked tokens at diffusion time t (Eq. (13)), CART applies a per-token weight $w_i(t, \mathbf{x}_t)$ that depends on how much usable (unmasked) context surrounds token i in the partially-masked sequence $\mathbf{x}_t$. This matters because two masked tokens at the same t can have very different predictability: a token with many nearby visible neighbors is effectively less ambiguous and should contribute a stronger learning signal, while a token whose neighborhood is mostly masked is inherently underdetermined and over-penalizing it injects gradient noise and encourages collapse toward “easy” tokens. CART operationalizes “how much context” as a distance-weighted sum of the visible tokens: each masked position receives one symmetric geometric kernel centered on every unmasked position, so nearer visible tokens contribute more and the weight grows with the amount of nearby visible context [8]. The resulting token-wise weights multiply the masked-token cross-entropy terms before averaging. We provide the full mathematical definition and its relationship to Eq. (13) in Appendix A.3.

3.4 MaRK Kernel Adapters

MaRK proposes the mapping $\psi(\cdot)$ that serves as the system’s scheduling function, translating high-dimensional semantic context into the low-dimensional manifold of stable dynamical parameters. By varying the basis of this mapping, we can control the spectral regularity of the adaptation process. The certified setting uses the bounded modulation structure from equations (3) and (4), and the three adapter geometries are ablated head-to-head in §4.4.

Hypernet Adapter: The Hypernet maps the hidden state h (obtained from the $\mathbf{c}_t$ via a mapping) directly to parameter shifts using linear projections W , to get the left and right low-rank matrices U and V , functioning as the most parameter-efficient baseline ($\approx 6.3\text{M}$ parameters):

$$\Theta_{\text{shift}} = W_{\Theta} h \quad (14)$$

$$UV_{\Phi_{\text{scale/shift}}} = W_{\Phi_{\text{scale/shift}}} h \quad (15)$$

The bounded scales control the gain amplification avoiding explosive states, whereas the shifts are kept optionally left unconstrained for the Hypernet kernel only, to be treated as a residual correction and act as the most expressive baseline.

Chebyshev Polynomial Adapter: To enforce smoother gradients, this kernel uses orthogonal polynomial basis functions evaluated at a latent coordinate $z = W_z h$. The Chebyshev polynomials $T_k(z)$ (up to degree K) determine the continuous shifts ($\approx 11\text{M}$ parameters):

$$\Theta_{\text{shift}} = \sum_{k=1}^K [W_{\Theta}(h)]_k \cdot T_k(z) \quad (16)$$

For matrix elements, we utilize the basis mean $\bar{T}_k = \frac{1}{n_{\text{heads}}} \sum_i T_k(z_i)$:

$$UV_{\Phi_{\text{scale/shift}}} = W_{\text{base}}(h) + \sum_{k=1}^K [W_{\text{basis}}(h)]_k \cdot \bar{T}_k \quad (17)$$

Discrete Cosine Transform (DCT) Kernel: The DCT kernel captures spectral dynamics across diffusion steps, it first forms a length- L signal u by combining frequency-damped spectral weights from h (with learnable decay exponent α via $1/(f+1)^{\text{softplus}(\alpha)+\epsilon}$ where $\epsilon \in (0, 1)$ is a constant added for numerical stability) with a fixed orthonormal Type-II DCT cosine basis $\{\phi_f\}$, where f indexes cosine frequency bins [15], then applies the same bounded structure as (3)–(4) with readouts from u , and for the B, C base path, from h as well ($\approx 7.1\text{M}$ parameters):

$$u = \sum_f \frac{[W_{\text{spec}} h]_f}{(f+1)^{\text{softplus}(\alpha)+\epsilon}} \phi_f. \quad \text{where } \epsilon = 1 \times 10^{-4} \quad (18)$$

Diagonal shifts and low-rank U, V factors are

$$\Theta_{\text{shift}} = W_{\Theta} u, \quad (19)$$

$$UV_{\Phi_{\text{scale/shift}}} = W_{\Phi_{\text{base}}} h + W_{\Phi_{\text{basis}}} u, \quad (20)$$

4 Experiments and Analysis

4.1 AQS Certificate and Diagnostics

The Affine Quadratic Stability (AQS) certificate provides an analytic stability check for the bounded diagonal recurrence induced by the MaRK kernels [16]. Rather than solving a generic Semi-Definite Program (SDP) to search for a common Lyapunov function, we exploit the constructive parameterization chain from §2.1 to obtain an analytical certificate with $P = I$.

We load the 23-layer pretrained Hydra base model and extract each layer’s diagonal recurrence parameters $A_{\log}$. Since the maximum modulation bound β_A is shared across the certified bounded variants, the bounded hyper-rectangle $[A_{\log} - \beta_A, A_{\log} + \beta_A]$ is kernel-independent within this setting.

For each layer ℓ and head i , we compute the worst-case discrete eigenvalue at the vertex $\delta = -\beta_A$, where $\bar{A}_i$ is closest to 1 (monotonicity of $\bar{A}_i(\delta)$ in δ , see Proposition 2.1), using the minimum discretization step $\Delta_{\min} = 0.001$. The energy function $V(x) = \|x\|^2$ (i.e., $P = I$) satisfies:

$$\bar{A}^\top I \bar{A} - I = \text{diag}(\bar{A}_i^2 - 1) \prec 0 \quad (21)$$

with per-layer stability margin $\varepsilon_\ell = 1 - \max_i \bar{A}_{i,\ell}^2 > 0$. All 23 layers certify with positive ε_ℓ under the stated modulation and discretization assumptions, supporting stability of the modulated Hydra recurrence across admissible conditioning contexts. The per-layer spectral radii and stability margins are visualized in Figure 1.

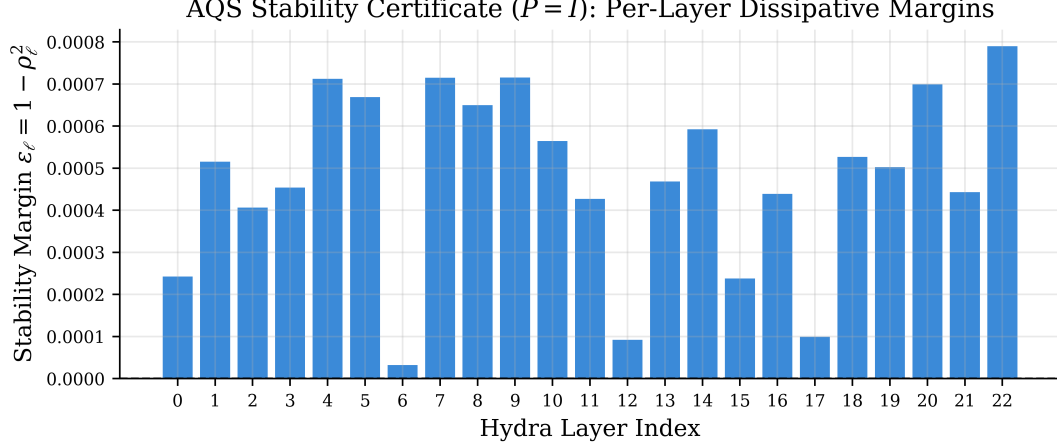


Figure 1: Stability margins (ε_ℓ) across all the layers of the pre-trained Hydra SSM at the extreme bounds of modulation by MaRK.

4.2 Synthetic LPV Identification

This experiment evaluates the learnability and theoretical identifiability of the MaRK architecture. While the internal state matrices inhabit an unidentifiable manifold due to non-unique factorizations, the underlying input-output dynamics are estimable under the bounded LPV assumptions below. We evaluate recovery using the coordinate-invariant Markov Operator Error (MOE): $\|\mathcal{H}_{\text{true}} - \hat{\mathcal{H}}_{\text{est}}\|_F$ (where $\mathcal{H}_{\text{true}}$ is the true Markov Parameter Sequence and $\hat{\mathcal{H}}_{\text{est}}$ is the estimated Markov Parameter Sequence by MaRK). We train synthetic Hypernet, Chebyshev, and DCT adapter variants on $N \in [50, 1600]$ trajectories with Gaussian output noise, keeping the backbone frozen to mirror our training regime; full synthetic protocol details are given in Appendix C.

Proposition 4.1 (Operator Identifiability and Convergence). *Let $\mathcal{M}$ be an Affine Quadratically Stable LPV-SSM with strictly bounded representations, observed under standard excitation and mixing conditions. The precise statistical model, including the frozen-context identification protocol, bounded-density innovation assumptions, and operator-level persistent excitation, is specified in Appendix B.2. Then (i) the input-output operator defined by the frozen-time Markov sequence $\mathcal{H}(\mathbf{c}_t) = \{\bar{C}'(\mathbf{c}_t)A'^k(\mathbf{c}_t)\bar{B}'(\mathbf{c}_t)\}_{k=0}^{\infty}$ is identifiable up to a structural equivalence class. (ii) Under squared loss, the empirical estimation error converges at the optimal parametric rate $O(1/\sqrt{N})$, where N is the total number of samples.*

Proof Sketch. We define model equivalence by equality of the input-output Markov sequence, which removes the non-identifiability of internal state coordinates while preserving the observable operator. The convergence argument then follows three standard steps. (i) **Mixing:** AQS gives geometric decay of the Markov operators; with persistently exciting conditioning and sub-Gaussian inputs/noise, as used in the synthetic recovery protocol, this places the induced temporal process in the exponentially β -mixing regime [21]. (ii) **Complexity:** The tanh-bounded parameterization makes the operator class uniformly bounded and Lipschitz, so Talagrand’s contraction lemma yields Rademacher complexity scaling as $O(1/\sqrt{N})$ under the mixing relaxation of i.i.d. sampling [22]. (iii) **Generalization:** Standard squared-loss bounds for dependent sequences transfer this capacity bound to the empirical operator-estimation error [23]. Appendix B.2 gives the full proof, including the structural equivalence relation and the cited statistical premises. $\square$

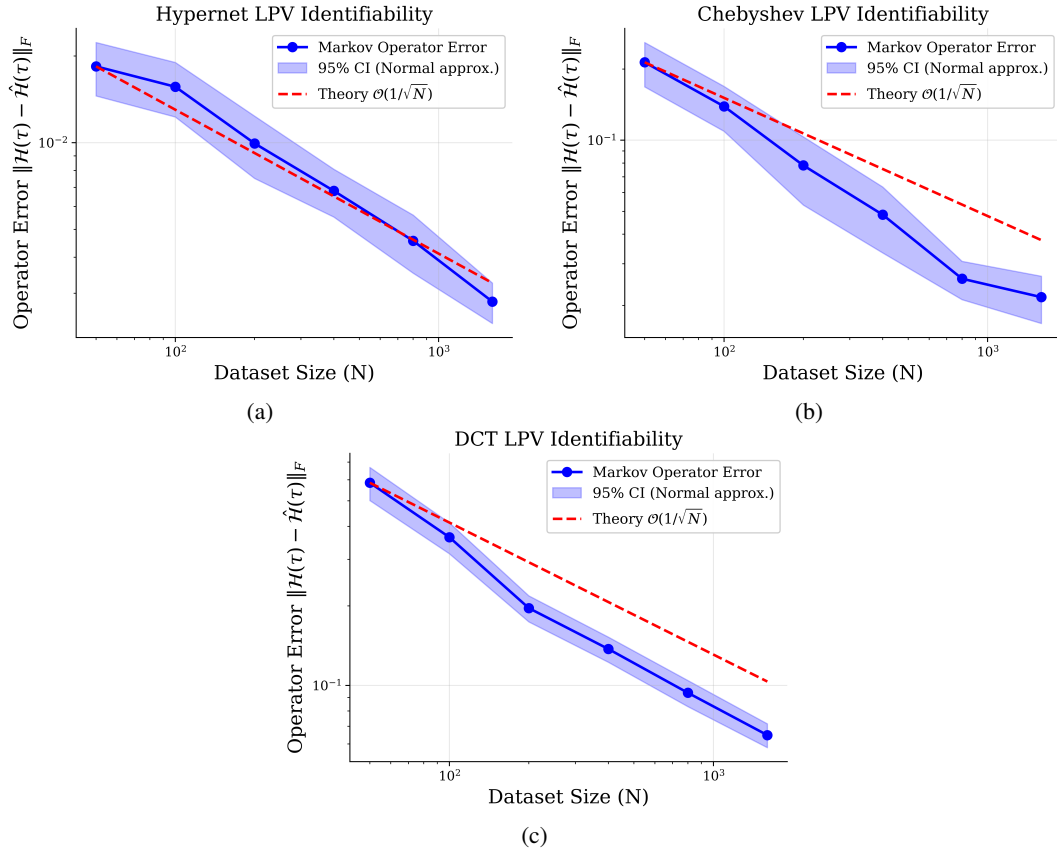


Figure 2: Markov Operator Error vs Dataset Size for synthetic Hypernet, Chebyshev, and DCT LPV recovery. Solid lines denote 15-seed means with normal-approximation 95% confidence intervals (shaded). Red dashed lines show an $O(N^{-1/2})$ reference normalized at the smallest dataset size.

Analysis of Convergence. Figure 2 illustrates the log-log scaling of the MOE. All three synthetic kernels exhibit systematic operator-error decay as the number of trajectories increases, supporting the

predicted $O(1/\sqrt{N})$ estimation trend in Proposition 4.1 under matched LPV recovery conditions. Fitting the endpoints over $N = 50$ to 1600 gives approximate empirical rates of $N^{-0.54}$ for Hypernet, $N^{-0.66}$ for Chebyshev, and $N^{-0.64}$ for DCT, consistent with parametric convergence up to kernel-dependent constants. Chebyshev achieves the lowest structured-basis error after the smallest sample regime, matching its smooth polynomial bias. DCT has a larger absolute constant, plausibly reflecting the additional bandlimited spectral restriction, but its monotone decay indicates that this restriction does not destroy operator-level recoverability. The experiment therefore supports the ability of MaRK kernel geometries to recover the coordinate-invariant Markov operator in a well-specified synthetic setting, while the theorem supplies a conditional statistical guarantee.

4.3 Dynamics Analysis

To test whether MaRK conditions the temporal operator, we measure the Markov parameter sequence induced by each frozen diffusion timestep. For each adapter family, we evaluate the head-averaged Markov norm $\|h_k(t)\|_2$ over lags $k \leq 4096$ at $t \in \{0, 0.25, 0.5, 0.75, 1.0\}$, and compare it with the unmodulated Hydra backbone. Since $\mathcal{H}(t) = \{C(t)A^k(t)B(t)\}_{k \geq 0}$ is the input-output memory profile of the SSM, separation between these curves provides evidence of timestep-dependent temporal kernels rather than merely timestep-dependent activations. The experiment therefore asks two questions: whether MaRK can reshape the Markov sequence as the diffusion process changes, and whether the resulting kernels remain dissipative at long lags.

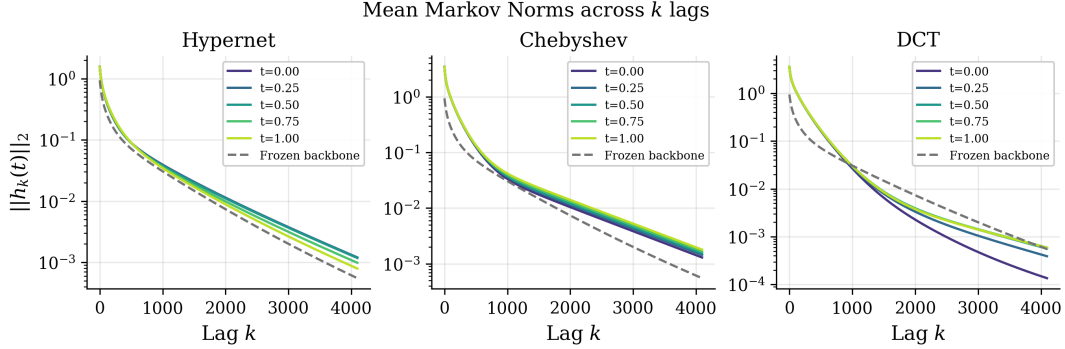


Figure 3: Head-resolved mean Markov parameter norm versus lag for Hypernet, Chebyshev, and DCT MaRK adapters at different diffusion timesteps t . The dashed curve denotes the frozen Hydra backbone without timestep-conditioned operator modulation.

Analysis of Markov Dynamics. Figure 3 shows that all three adapter geometries preserve a stable decay envelope while producing distinct timestep-conditioned Markov sequences. The curves remain approximately geometric over thousands of lags, consistent with the AQS construction, but they do not collapse onto the frozen backbone: MaRK changes how much input-output energy is retained at each temporal scale. Hypernet produces the broadest upward shift relative to the backbone, especially at medium and long lags, suggesting an expressive increase in memory persistence. Chebyshev yields a tightly ordered family of smooth decay curves, matching the low-oscillation bias of polynomial modulation and indicating controlled, stable variation across diffusion time. DCT exhibits the most selective redistribution of energy: high-noise timesteps amplify early-lag response, while lower timesteps attenuate more rapidly and cross the backbone at intermediate lags, consistent with spectral shaping rather than uniform gain scaling. Together, these signatures support the claim that MaRK conditions the Markov parameter sequence, giving the diffusion timestep explicit influence over the SSM memory kernel while preserving long-horizon dissipation.

Remark on Operator Adaptation. Figure 3 also illustrates all variants of MaRK possess a distinct signature compared to the frozen Hydra backbone. Since the Markov parameter sequence $\{\mathcal{H}_k\}_{k=0}^{\infty}$ is the core characteristic of the SSM, it serves as a canonical operator for the system’s dynamics. This is because the sequence uniquely determines the transfer function and the impulse response of the model [18], remaining invariant under any similarity transformation of the state-space coordinates [24]. Consequently, the observed divergence in the norm signature across k -lags is consistent with a shift in the model’s impulse response. This shift suggests that MaRK does not merely perform a

linear re-projection of features, but also adapts the underlying dynamical operators (specifically the spectral properties and memory decay) to align with the temporal dependencies of the target task. This supports viewing our approach as a principled mechanism for operator-level fine-tuning.

4.4 Ablations

We ablate the three MaRK adapter geometries under the same frozen Hydra backbone, diffusion training objective, and CART-weighted validation protocol. The goal is to isolate the effect of the operator-modulation map $\psi(\cdot)$: direct Hypernet projections, smooth Chebyshev polynomial bases, or bandlimited DCT bases. Table 1 reports $\mathcal{L}_{\text{CART}}$, where lower values indicate better reconstruction of masked tokens under the same context-adaptive weighting used for training. We include trainable adapter size to separate geometry effects from parameter count, and interpret the results as an internal MaRK design ablation rather than as a full scaling comparison against separately trained diffusion language models.

Table 1: CART losses ($\mathcal{L}_{\text{CART}} \downarrow$) across 10 seeds (mean $\pm$ 95% CI, $1.96 \times \text{SEM}$) and trainable adapter sizes across completed validation datasets. **Bold** = Best. Underline = Second Best.

Kernel	Params	PTB	WikiText	Lambda	AG News	Arxiv	Avg.
Hypernet	6.3M	3.39 ± 0.31	3.73 ± 0.23	3.70 ± 0.17	3.91 ± 0.20	4.12 ± 0.10	3.77
Chebyshev	11.0M	2.42 ± 0.26	<u>2.55 ± 0.16</u>	2.40 ± 0.10	2.85 ± 0.14	2.55 ± 0.05	2.55
DCT	7.1M	<u>2.48 ± 0.27</u>	2.55 ± 0.14	<u>2.47 ± 0.11</u>	<u>2.86 ± 0.14</u>	<u>2.61 ± 0.07</u>	<u>2.59</u>

Both structured bases improve on the direct Hypernet projection on every completed validation dataset. Chebyshev gives the lowest average loss (2.55) and leads DCT on four of the five datasets, tying it on WikiText; DCT is 0.04 nats behind on the average (2.59). The cause is structural rather than statistical. The timestep is a continuous scalar and adjacent timesteps define nearly identical denoising tasks, so the modulation map should vary smoothly, but the Hypernet MLP consumes each timestep as an independent input and must learn that smoothness from data. It can therefore fit the training timesteps accurately while varying freely between them, spending capacity on high-frequency structure the diffusion process does not need. Chebyshev and DCT encode the smoothness in the basis itself: five global polynomials, or eight frequencies with a learned spectral decay that attenuates the high end. Projecting the modulation map onto a compact basis functions set acts as a strong structural regularizer, constraining the functional space without compromising expressive capacity. Table 1 shows the effect at equal budget: Hypernet and DCT carry 6.3M and 7.1M adapter parameters and still differ by 1.18 nats, while the further 3.9M that separate DCT from Chebyshev (11.0M) buys an additional 0.04 nats. Hypernet does not overfit; it lacks the prior. Chebyshev then separates from DCT only on Arxiv (2.55 vs. 2.61), Lambda (2.40 vs. 2.47) and PTB (2.42 vs. 2.48), with WikiText and AG News inside seed noise, and DCT is the cheaper of the two at inference, so Chebyshev is the accuracy ceiling and DCT the better parameter and latency trade.

Table 2: Parameter ablation: CART validation loss ($\mathcal{L}_{\text{CART}} \downarrow$) on the WikiText dataset across 10 seeds (mean $\pm$ 95% CI, $1.96 \times \text{SEM}$). "Full" modulates all five parameters; "None" modulates none (frozen Hydra). **Bold** = Best per kernel.

Kernel	Full	All Except A	A Only	Δ Only	B,C Only	D Only	All Except Δ	None
Hypernet	3.73 ± 0.26	3.91 ± 0.27	4.07 ± 0.29	4.04 ± 0.29	4.08 ± 0.29	3.78 ± 0.26	3.74 ± 0.26	3.99 ± 0.29
Chebyshev	2.55 ± 0.18	2.78 ± 0.19	3.44 ± 0.22	3.92 ± 0.26	3.06 ± 0.21	3.90 ± 0.26	2.56 ± 0.18	4.00 ± 0.27
DCT	2.55 ± 0.16	2.80 ± 0.16	3.24 ± 0.17	3.99 ± 0.24	3.05 ± 0.17	3.90 ± 0.24	2.59 ± 0.16	4.00 ± 0.24

A full leave-one-out parameter ablation across all five state-space model (SSM) parameters (A, B, C, D, Δ) isolated the performance contributions of each operator-modulation component. Modulating matrix A proves necessary across all settings; freezing A (All Except A) increases validation loss by +0.18 to +0.25 points. This performance gap confirms that MaRK cannot be reduced to standard Mamba-style selection (B, C, Δ), as the B, C -Only variant leaves a substantial loss degradation of +0.34 to +0.51 points. This result aligns with the functional distinction where

input gating controls instantaneous throughput while A governs persistent memory horizon [1, 12]. Furthermore, modulating Δ yields marginal independent gains due to the redundancy in spectral transformation paths: the operator dynamics can be shifted directly via A or indirectly via Δ through zero-order hold discretization $\bar{A} = \exp(A_{\text{cont}} \cdot \Delta)$. The direct spectral path dominates empirically, as A -Only outperforms Δ -Only by up to 0.75 points, and All Except Δ achieves the second-best score across every kernel. Nevertheless, single-parameter configurations trail the Full baseline by up to +0.50 points, demonstrating that optimal performance requires joint parameter modulation.

Table 3: Inference performance and relative latency overhead over Frozen Hydra at batch size 8 on an NVIDIA A100-SXM4-80GB with the same frozen 23-layer backbone. Each variant: 15 seeds, 100 forward passes per seed, means $\pm$ 95% CI (t-distribution, 14df). **Bold** = MaRK (Ours)

Variant	Latency (ms) seq=4096	Memory (MiB) seq=4096	Total Params (M)	Additional Params (M)	Overhead seq=512	Overhead seq=4096
Frozen Hydra	267.43 $\pm$ 0.06	4446.58 $\pm$ 0.06	111.68	0.00	-	-
Input Injection	270.86 $\pm$ 0.03	4455.25 $\pm$ 0.06	113.96	2.28	+3.6%	+1.3%
AdaLN Zero	276.84 $\pm$ 0.02	4474.52 $\pm$ 0.60	118.51	6.84	+7.6%	+3.5%
Hypernet	304.75 $\pm$ 0.03	4470.62 $\pm$ 0.06	117.95	6.27	+65.6%	+14.0%
Chebyshev	309.74 $\pm$ 0.10	4488.61 $\pm$ 0.06	122.65	10.97	+104.0%	+15.8%
DCT	307.28 $\pm$ 0.05	4473.93 $\pm$ 0.06	118.75	7.07	+88.7%	+14.9%

At sequence length 4096, MaRK incurs a modest 14% to 16% latency increase while maintaining memory overhead below 1% relative to the frozen Hydra backbone. The per-layer conditioning computation introduces a constant temporal additive cost (~ 37 ms to 42 ms) that remains independent of sequence length. Consequently, while conditioning overhead dominates leading to 65% to 104% additional latency at length 512, this cost amortizes rapidly as sequence length scales to 4096 and beyond. Because MaRK scales $O(1)$ with sequence length and $O(N)$ with diffusion sampling steps, its asymptotic runtime trajectory aligns with standard AdaLN Zero and Input-Injection baselines.

5 Conclusion

We introduced **Dynamic Operator Modulation**, an operator-level conditioning perspective that shifts SSM adaptation from external signal injection toward direct reconfiguration of the recurrent operator. By mapping context to bounded, low-rank modulations of the frozen backbone, MaRK induces a context-indexed family of stable input-output dynamics. This operator-level perspective is our central contribution: adapting an SSM for iterative tasks like diffusion can be understood as a controlled deformation of the model’s underlying Markov parameters.

Our framework provides theoretical support (through Affine Quadratic Stability certificates and Rademacher complexity bounds) and empirical evidence across multiple adapter geometries. The observed shift in Markov signatures suggests that parameter-efficient fine-tuning can arise from the structure of this modulation, rather than only from an additive patch. Overall, these results suggest that the temporal logic of an SSM can be made context-aware in the studied setting while preserving stability and recoverability under the stated assumptions, opening new avenues for principled adaptation in non-stationary sequence modeling. Systematic empirical comparison against established conditioning frameworks on downstream benchmarks, and how MaRK holds under scaling laws remain important future directions; we discuss these limitations further in Appendix A.5.

6 Acknowledgements

This project was made possible by the generous support provided by the NVIDIA AI Technology Center (NVAITC). Additionally, we gratefully acknowledge Dr. Yuhao Wang from the Department of Data Science at City University of Hong Kong for his assistance during the final stages of this project.

References

- [1] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024.
- [2] Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 10041–10071, 2024.
- [3] Aakash Lahoti, Kevin Y. Li, Berlin Chen, Caitlin Wang, Aviv Bick, J. Zico Kolter, Tri Dao, and Albert Gu. Mamba-3: Improved sequence modeling using state space principles. *CoRR*, abs/2603.15569, 2026.
- [4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.
- [5] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In Marc’Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 17981–17993, 2021.
- [6] Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. Diffusion-lm improves controllable text generation. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.
- [7] Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, JUN ZHOU, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *ICLR 2025 Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*, 2025.
- [8] Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *CoRR*, abs/2508.15487, 2025.
- [9] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 4172–4182. IEEE, 2023.
- [10] Jing Nathan Yan, Jiatao Gu, and Alexander M. Rush. Diffusion models without attention. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 8239–8249. IEEE, 2024.
- [11] Hao Phung, Quan Dao, Trung Tuan Dao, Hoang Phan, Dimitris N. Metaxas, and Anh Tuan Tran. DiMSUM: Diffusion mamba - a scalable and unified spatial-frequency method for image generation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [12] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022.
- [13] Sukjun Hwang, Aakash Sunil Lahoti, Ratish Puduppully, Tri Dao, and Albert Gu. Hydra: Bidirectional state space models through generalized matrix mixers. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

- [14] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022.
- [15] Nasir Ahmed, T. Natarajan, and Kamisetty R. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, 100(1):90–93, 1974.
- [16] Pepijn Bastiaan Cox. *Towards Efficient Identification of Linear Parameter-Varying State-Space Models*. PhD thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, March 2018. Proefschrift.
- [17] Jeff S. Shamma. *Analysis and Design of Gain Scheduled Control Systems*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1988. The LPV-SSM equation is introduced on page 42.
- [18] T. Kailath. *Linear Systems*. Information and System Sciences Series. Prentice-Hall, 1980. Markov parameter sequence $H_k = CA^{k-1}B$ (for $k \geq 1$) defined on pp. 70, 113 and 354.
- [19] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 62429–62442, 2024.
- [20] Wenchao Chen, Liqiang Niu, Ziyao Lu, Fandong Meng, and Jie Zhou. Maskmamba: A hybrid mamba-transformer model for masked image generation. *CoRR*, abs/2409.19937, 2024.
- [21] Stephen Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), 1994.
- [22] Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: Isoperimetry and Processes*. Springer-Verlag, Berlin, New York, 1991.
- [23] Peter L. Bartlett and Shahr Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.
- [24] Chi-Tsong Chen. *Linear System Theory and Design*. The Oxford Series in Electrical and Computer Engineering. Oxford University Press, New York, NY, 3rd edition, 1998.
- [25] Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dallar-torre, Bernardo P de Almeida, Alexander M Rush, Thomas PIERROT, and Volodymyr Kuleshov. Simple guidance mechanisms for discrete diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [26] David Ha, Andrew M. Dai, and Quoc V. Le. Hypernetworks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.
- [27] Eldad Haber and Lars Ruthotto. Stable architectures for deep neural networks. *Inverse Problems*, 34(1):014004, December 2017.
- [28] Tomer Galanti and Lior Wolf. On the modularity of hypernetworks. In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.
- [29] T. D. Pham and L. T. Tran. Some mixing properties of time series models. *Stochastic Processes and their Applications*, 19(2):297–303, 1985.
- [30] Bin Yu. Rates of convergence for empirical processes of stationary mixing sequences. *The Annals of Probability*, 22(1):94–116, 1994.

- [31] Mehryar Mohri and Afshin Rostamizadeh. Rademacher complexity bounds for non-i.i.d. processes. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pages 1097–1104. Curran Associates, Inc., 2008.
- [32] Vitaly Kuznetsov and Mehryar Mohri. Generalization bounds for non-stationary mixing processes. *Machine Learning*, 106(1):93–117, 2017.

A Discussions

A.1 Notation Summary

Table 4: Compact symbol guide for the main text.

Symbol	Meaning
t, τ	diffusion timestep and internal continuous time
$\mathbf{c}_t$	context/timestep embedding used as the LPV scheduling variable
Q, Q'	base and modulated SSM parameters, $Q \in \{A, B, C, D, \Delta\}$
$A(\mathbf{c}_t), B(\mathbf{c}_t)$	LPV-SSM matrix-valued functions for recurrence and read-in parameters
$C(\mathbf{c}_t), D(\mathbf{c}_t)$	LPV-SSM matrix-valued functions for read-out and skip parameters
$x(\tau), u(\tau), y(\tau)$	LPV-SSM vector-valued functions for state, input and output
A, B, C, D, Δ	SSM recurrence, read-in, read-out, skip, and discretization parameters
$A'_{\log}, \Delta'_{\text{input}}, \Delta'_{\text{bias}}$	modulated log-recurrence, discretization input, and modulated discretization bias
Θ, Φ	diagonal parameters $\{A, \Delta, D\}$ and dense projections $\{B, C\}$
$\psi_{\text{scale}}, \psi_{\text{shift}}$	adapter maps that emit bounded scales and shifts
$\alpha_{\Phi}, \beta_{\Phi}, \beta_{\Theta}$	modulation magnitudes for scales and shifts
P, I, ε	Lyapunov matrix, identity matrix, and AQS stability margin
$\bar{A}, \bar{B}$	ZOH-discretized recurrence and read-in parameters
$\mathcal{H}(\mathbf{c}_t), h_k$	context-indexed Markov parameter sequence and its k th lag term
$\mathcal{V}, \mathcal{M}, \theta$	vocabulary, absorbing mask token, and trainable model parameters
h, W, z, T_k, ϕ_f	adapter hidden state, learned readouts, Chebyshev coordinate/polynomial, and DCT basis
$w(t), w_i(t, \mathbf{x}_t)$	global diffusion weight and CART token-level weight
U, V, r	low-rank factors and rank used for dense-parameter modulation

A.2 The MaRK Kernel Classes

We propose three principal variants of the MaRK kernels: Hypernet, Chebyshev, and DCT. Each kernel structurally modifies the recurrent dynamics (A, Δ) and input/output projections (B, C) of the base State Space Model. In the certified bounded setting, shifts are modulated by bounded activation functions (hyperbolic tangent gating) to support AQS and prevent unbounded activation growth.

The three variants should be read as different parameterizations of the same bounded operator-modulation map rather than as unrelated architectures. Hypernet places the weakest structural prior on the context-to-operator map and is therefore useful as a flexible baseline. Chebyshev and DCT instead restrict the modulation to structured bases: Chebyshev favors globally smooth variation over the conditioning coordinate, while DCT favors bandlimited spectral variation with explicit attenuation of higher-frequency components. This distinction is important because the empirical comparison in Table 1 tests not only whether MaRK can modulate the SSM operator, but also which inductive bias makes that modulation reliable under a frozen backbone and finite adaptation budget.

- **Expressivity.** Hypernet can represent the least constrained modulation among the three variants, but this flexibility may allocate capacity to changes that are not necessary for the task. Chebyshev and DCT reduce this search space by forcing modulations to pass through low-dimensional basis expansions.
- **Optimization and parameter efficiency.** The basis kernels trade some pointwise flexibility for smoother parameter trajectories. This can make the adapter easier to regularize and interpret, although the actual compute and parameter cost still depends on basis degree, frequency budget, and low-rank factor sizes.
- **Interpretability.** Chebyshev coefficients can be interpreted as polynomial components of the operator path, while DCT coefficients separate low- and high-frequency modulation content. Hypernet is less directly interpretable, since the MLP output mixes these effects before the bounded shifts are applied.

A.2.1 Hypernet Adapter

The Hypernet acts as the fundamental MaRK baseline, employing standard linear projections directly from the hidden conditioning variable to the factorized shift components. This variant serves as the most expressive baseline because the MLP can learn a comparatively unconstrained mapping from the conditioning vector to the modulations. Its main role in the study is therefore diagnostic: it asks whether direct learned projections are sufficient, before imposing the smoother basis restrictions used by Chebyshev or DCT.

Algorithm 1 Hypernet MaRK Kernel

- 1: **Input:** Context vector $\mathbf{c}_t$, Base parameters $A_{\log}, B, C, \Delta, D$
 - 2: **Hyperparameters:** Factorization constraints α, β
 - 3: $h \leftarrow \text{MLP}(\mathbf{c}_t)$ ▷ Extract hidden representation
 - 4: $A_{\text{shift}}, \Delta_{\text{shift}}, D_{\text{shift}} \leftarrow \text{Linear}_{A, \Delta, D}(h)$
 - 5: $B_{\text{scale}}, B_{\text{shift}} \leftarrow \text{LowRankProject}(h)$
 - 6: $C_{\text{scale}}, C_{\text{shift}} \leftarrow \text{LowRankProject}(h)$
 - 7: **Modulation:**
 - 8: $A'_{\log} \leftarrow A_{\log} + \beta_A \tanh(A_{\text{shift}})$
 - 9: $\Delta' \leftarrow \Delta + \beta_{\Delta} \tanh(\Delta_{\text{shift}})$
 - 10: $B' \leftarrow B \odot (1 + \alpha_B \tanh(B_{\text{scale}})) + \beta_B \tanh(B_{\text{shift}})$
 - 11: $C' \leftarrow C \odot (1 + \alpha_C \tanh(C_{\text{scale}})) + \beta_C \tanh(C_{\text{shift}})$
 - 12: **Return** modulated parameters
-

A.2.2 Chebyshev Polynomial Adapter

The Chebyshev polynomial adapter imposes structural smoothness by mapping conditional modulations onto orthogonal polynomial bases. Rather than predicting unbounded shifts directly, the hidden dimension governs a latent evaluation coordinate $z \in [-1, 1]$ and generates series coefficients. This creates a low-degree global approximation to the operator path, which is appropriate when the desired recurrent dynamics vary smoothly with context. The same global support also means that changing one coefficient can affect the modulation over the full coordinate range, so the degree K controls a bias-variance tradeoff rather than merely increasing capacity.

Algorithm 2 Chebyshev MaRK Kernel

- 1: **Input:** Context vector $\mathbf{c}_t$, Base parameters $A_{\log}, B, C, \Delta, D$
 - 2: **Hyperparameters:** Degree K , Scaling constraints α, β
 - 3: $h \leftarrow \text{MLP}(\mathbf{c}_t)$
 - 4: $z \leftarrow \text{Linear}_z(h)$ ▷ Project hidden state to latent basis coordinate
 - 5: $T(z) \leftarrow [T_0(z), T_1(z), \dots, T_{K-1}(z)]$ ▷ Evaluate Chebyshev polynomials up to degree K
 - 6: $\theta_{\text{coeffs}} \leftarrow \text{Linear}_{\theta}(h)$ for $\theta \in \{A, \Delta, D\}$
 - 7: $\theta_{\text{shift}} \leftarrow \sum_{k=0}^{K-1} \theta_{\text{coeffs}}^{(k)} \cdot T_k(z)$ for $\theta \in \{A, \Delta, D\}$
 - 8: $\bar{T} \leftarrow \frac{1}{n_{\text{heads}}} \sum_i T(z_i)$ ▷ Calculate shared basis mean for broad matrix shifts
 - 9: $X_{\text{scale}} \leftarrow \text{UV}_{X_{\text{scale_base}}}(h) + \sum_{k=0}^{K-1} \text{UV}_{X_{\text{scale_basis}}}(h)^{(k)} \cdot \bar{T}_k$ for $X \in \{B, C\}$
 - 10: $X_{\text{shift}} \leftarrow \text{UV}_{X_{\text{shift_base}}}(h) + \sum_{k=0}^{K-1} \text{UV}_{X_{\text{shift_basis}}}(h)^{(k)} \cdot \bar{T}_k$ for $X \in \{B, C\}$
 - 11: **Modulation:** Evaluate modulated representations using base parameters and $X_{\text{scale/shift}}$ via bounded tanh saturation (Step 8-11 in Algorithm 1)
 - 12: **Return** modulated parameters
-

A.2.3 Discrete Cosine Transform (DCT) Kernel

The DCT kernel imposes a spectral inductive bias by casting modulations as bandlimited real-valued basis transformations. Rather than assuming that high-frequency modulation is always harmful, the parameterization makes high-frequency content explicit and attenuates it through a learned monotone decay. Drawing inspiration from the Inverse Real Fast Fourier Transform (IRFFT) without requiring unconstrained complex arithmetic, we formulate a deterministic real-valued basis grid. Contextual

representations evaluate spectral components that decay across higher frequencies before being mapped back into parameter shifts. This makes DCT most natural when the useful operator variation is expected to be coarse-to-fine or spectrally organized; if the true modulation requires sharp localized changes, the frequency budget and decay prior can become limiting factors.

Algorithm 3 DCT MaRK Kernel

- 1: **Input:** Context vector $\mathbf{c}_t$, Base parameters $A_{\log}, B, C, \Delta, D$
 - 2: **Hyperparameters:** NumFreqs n_f , Basis grid timepoints L , Decay bound α_{decay}
 - 3: $h \leftarrow \text{MLP}(\mathbf{c}_t)$
 - 4: $w_{\text{decay}}(f) \leftarrow \frac{1}{(f+1)^{\text{softplus}(\alpha)+\epsilon}}$ $\triangleright$ Evaluate learned frequency attenuation
 - 5: $\text{Spec}_{\text{weights}} \leftarrow \text{Linear}_{\text{spec}}(h) \odot w_{\text{decay}}$
 - 6: $\text{Time}_{\text{vec}} \leftarrow \text{Spec}_{\text{weights}} \times \mathcal{B}_{\text{DCT-II}}$ $\triangleright$ Project bandlimited spectral amplitudes into temporal structure
 - 7: $\theta_{\text{shift}} \leftarrow \text{Linear}_{\theta}(\text{Time}_{\text{vec}})$ for $\theta \in \{A, \Delta, D\}$
 - 8: $X_{\text{scale}} \leftarrow \text{UV}_{X_{\text{scale_base}}}(h) + \text{UV}_{X_{\text{scale_basis}}}(\text{Time}_{\text{vec}})$ for $X \in \{B, C\}$
 - 9: $X_{\text{shift}} \leftarrow \text{UV}_{X_{\text{shift_base}}}(h) + \text{UV}_{X_{\text{shift_basis}}}(\text{Time}_{\text{vec}})$ for $X \in \{B, C\}$
 - 10: **Modulation:** Evaluate modulated representations using base parameters and $X_{\text{scale/shift}}$ via bounded tanh saturation (Step 8-11 in Algorithm 1)
 - 11: **Return** modulated parameters
-

Overall, the kernel choice determines how much structure is imposed before the bounded AQS-compatible shifts are applied. Hypernet may overfit or produce rapidly varying modulations when the available data do not justify its flexibility. Chebyshev may oversmooth localized changes because its polynomial basis has global support. DCT may underfit non-bandlimited operator paths if too few frequencies are retained or if the learned decay is too strong. These are not separate stability assumptions, since all variants still use bounded shifts, but they are practical modeling assumptions that affect how efficiently MaRK uses its trainable degrees of freedom.

A.3 CART Objective

To rigorously adapt the training criteria against distributional mode collapse in heavily masked regimes, we replace standard Categorical Cross-Entropy (CCE) masked language modeling loss with the Context-Adaptive Token-Level Noise Rescheduling (CART) objective. In conventional sequence modeling, heavy masking at early diffusion steps often degenerates the output tensor into identical redundant tokens predicting the global marginal distribution rather than contextualized completions. To prevent mode collapse, the objective introduces differentiated spatial penalty terms, adapted from Dream [8]. As a consequence, when evaluating with a global context such as MDLM ELBO [25], the perplexities may seem inflated.

We provide the full mathematical specification of the Context-Adaptive Token-level Noise Rescheduling (CART) objective described in §3.3, including the token-wise weighting function and its substitution into the masked diffusion loss.

Let $m_i = \mathbf{1}[x_{t,i} = \mathcal{M}]$ indicate which tokens are masked at timestep t . Following Dream [8], CART replaces the single sequence-level weight with a per-token weight built from a symmetric geometric kernel over token distance. For a masked token i ,

$$w_i(t, \mathbf{x}_t) = \frac{1}{2} \sum_{j=1}^L (1 - m_j) \text{Geo}(p, |i - j| - 1), \quad \text{Geo}(p, d) = p(1 - p)^d, \quad (22)$$

where the sum runs over the visible (unmasked) tokens and $p \in (0, 1]$ controls the kernel sharpness. A small p spreads each clean token’s influence almost uniformly across the sequence, while a large p concentrates it on nearby masked tokens. Because the weight is the sum of one geometric kernel per visible token, a masked position surrounded by many nearby visible tokens receives a larger weight; this is the mixture of geometric kernels over clean anchors used in Dream. We use $p = 0.45$ (Appendix C.2). Substituting $w_i(t, \mathbf{x}_t)$ into Eq. (13) yields the CART-weighted training objective:

$$\mathcal{L}_{\text{CART}}(\theta) = \mathbb{E}_{\mathbf{x}_0, t, \mathbf{x}_t} \left[\sum_{i=1}^L m_i w_i(t, \mathbf{x}_t) \left(-\log p_{\theta}(x_{0,i} \mid \mathbf{x}_t, t) \right) \right]. \quad (23)$$

A.4 Scope of Contribution

Before discussing limitations, we clarify the scope of what this paper establishes. Our contribution is a theoretically grounded formulation of **Dynamic Operator Modulation** for SSMs: instead of injecting conditioning information only through inputs or activation scaling, MaRK maps a context vector into bounded changes of the recurrent operator itself. The comparison to Adaptive Layer Normalization (AdaLN) and input-stream injection in Section 1 is a structural motivation, not a claim that MaRK dominates those mechanisms in all empirical settings.

Scope of the conditioning signal. The operator-conditioning claim is scoped to a continuous, scalar, ordered scheduling variable, which in this work is the diffusion timestep $t \in [0, 1]$. This setting matters because adjacent timesteps index nearly identical denoising tasks, so the context-to-operator map should vary smoothly with the coordinate. Chebyshev and DCT encode that smoothness structurally, which is the mechanism behind their advantage over the Hypernet adapter (Section 4.4). The MaRK architecture accepts any embedding as $\mathbf{c}_t$, but the smooth-operator prior is specific to a continuous and ordered coordinate, so we do not claim transfer to autoregressive next-token modeling, which has no denoising coordinate to schedule against, or to high-dimensional prompt conditioning, where the signal is discrete and carries no proximity metric. Those regimes are left to future work.

The central technical challenge is that Dynamic Operator Modulation is a meta-function learning problem: a conditioning vector $\mathbf{c}_t$ must determine continuous-time system parameters (A, B, C, D, Δ) that control long-range temporal behavior. This resembles the setting studied by hypernetworks [26], where a side network produces parameters for a primary model. In recurrent or state-space systems, such context-to-operator maps raise three practical risks:

- (i) **Training instability and divergence.** If generated recurrence parameters are unconstrained, small context-dependent errors can move eigenvalues toward unstable regions, producing unstable forward dynamics or poorly conditioned gradients [26, 27].
- (ii) **Identifiability collapse.** A low-dimensional conditioning signal can map to many internally different state-space parameterizations that represent the same input-output behavior, and poorly structured maps can become nearly insensitive to $\mathbf{c}_t$ [28].
- (iii) **Optimization burden.** Emitting large dense parameter tensors from a small context vector can introduce high-variance gradients and make the context-to-operator map difficult to train reliably [26].

MaRK addresses these risks through a constrained architecture rather than by generating unrestricted recurrent weights.

- **Stability** is controlled by bounded log-space modulations and positive discretization, yielding the constructive AQS certificate in Section 4.1 and Appendix B under the stated parameterization [16].
- **Operator recovery** is studied at the level of the coordinate-invariant Markov parameter sequence. The synthetic LPV experiments support recovery of this input-output operator under matched assumptions, while the theory states the corresponding assumptions explicitly.
- **Trainability** is encouraged by low-dimensional outputs: the adapters predict scalar shifts and low-rank scales rather than full dense matrices, using 6.3–11M trainable parameters on top of a frozen 111M-parameter backbone.

Thus, the contribution of this work is the formulation and initial validation of an operator-level conditioning paradigm for SSMs. The parameter ablation in Table 2, the inference-overhead comparison in Table 3, and the Markov-operator diagnostics in Section 4.3 provide mechanistic and systems-level evidence that the conditioning signal reaches the recurrent operator rather than only external activations. None of these is a matched-budget quality comparison against external conditioning methods, and Appendix A.5 states what remains open.

A.5 Limitations and Future Directions

This work establishes MaRK as a stable and identifiable operator-level conditioning framework for SSMs, but its empirical scope is intentionally narrower than the full space of conditioning methods. Our experiments validate internal MaRK design choices (kernel geometry in Table 1,

SSM-parameter coverage in Table 2), compare systems cost against external conditioning methods (Table 3), and diagnose changes in the Markov parameter sequence (Figure 3). The comparison against AdaLN and input-stream injection is limited to inference latency, memory, and parameter count. We do not provide a controlled head-to-head baseline against those methods under matched training compute, objective, and frozen-backbone constraints, so we make no claim that operator modulation outperforms them on reconstruction quality.

The present results also do not establish scaling laws for Dynamic Operator Modulation. All reported language-model experiments use a 111M-parameter Hydra-BERT backbone with 6.3–11M trainable adapter parameters. This setting is sufficient to test whether bounded operator modulation is feasible, stable, and trainable, but it does not determine how MaRK behaves as model depth, hidden width, adapter rank, sequence length, training tokens, or pretrained SSM family vary. In particular, the Markov-dynamics diagnostics in Figure 3 suggest that MaRK can reshape long-lag memory profiles, but they should be interpreted as mechanistic evidence rather than as a completed long-context scaling study.

Our evaluation is likewise centered on the claims of this paper: AQS certificates, synthetic LPV recovery, Markov-operator diagnostics, and CART-weighted validation loss. These measurements support stability, recoverability under matched assumptions, and timestep-conditioned operator adaptation. We now report inference systems metrics (Table 3), but the evaluation does not cover generation quality, downstream task accuracy, or robustness across long-context language benchmarks. Future work should test MaRK on controlled long-range suites such as LRA and language long-context evaluations, and should report both quality and systems metrics there.

The theoretical guarantees depend on assumptions that may be violated in less structured settings. The identifiability and generalization arguments assume a well-specified bounded LPV-SSM class, stable Markov operators, sufficiently exciting inputs, sub-Gaussian noise, and mixing conditions. The written stability proof relies on the bounded MaRK parameterization and constructive ZOH chain, while the statistical-learning proof composes standard concentration, mixing, and Rademacher-complexity results cited from the literature. These assumptions are appropriate for the synthetic recovery protocol and for the stability claims made here, but they should not be read as unconditional guarantees for arbitrary pretrained sequence models or arbitrary conditioning distributions.

Finally, the bounded parameterization that makes MaRK stable may also constrain expressivity. Performance can plausibly depend on the modulation bounds, the low-rank dimension, the Chebyshev degree, the DCT frequency count, and the diffusion weighting. A useful next stage is a systematic robustness study over these hyperparameters. The conditioning regime itself is also open: within this paper the scheduling variable is the diffusion timestep, and whether operator-level conditioning helps when the conditioning signal is not a continuous and ordered coordinate remains future work (Appendix A.4).

B Proofs

This section gives the written theoretical arguments for the strictly bounded MaRK kernels. We first state the AQS stability argument for bounded affine parameter variations, then prove the coordinate-invariant operator identifiability and statistical convergence guarantee for Proposition 4.1.

B.1 Proof of Affine Quadratic Stability (AQS)

Let the base matrix $A \in \mathbb{R}^{n \times n}$ and perturbed system $A(v) = A + \Delta A(v)$, where parameter variations v lie in a hyper-rectangle Ψ .

Theorem. The LPV system is Affine Quadratically Stable if there exists a matrix $P \succ 0$ and scalar $\varepsilon > 0$ such that for all vertices $v_k \in \Psi$, $A(v_k)^\top P A(v_k) - P \preceq -\varepsilon I$.

Proof. Because the mapping $\mathbf{c}_t \mapsto A(v)$ is modulated via strictly bounded differentiable activations ($\tanh$), the range of possible evaluations falls strictly within the convex hull defined by vertices v_k . This vertex argument applies to affine dependence on the bounded modulation variables; in the diagonal Hydra ZOH instantiation, the certificate is obtained directly with $P = I$ because $\bar{A}_i = \exp(-\exp(A'_{\log,i})\Delta_i) \in (0, 1)$. For any affine interpolation $A(\lambda) = \sum_k \lambda_k A_k$, matrix

convexity of $A \mapsto A^\top P A$ for $P \succ 0$ gives

$$A(\lambda)^\top P A(\lambda) \preceq \sum_k \lambda_k A_k^\top P A_k.$$

Thus, if the strictly negative definite LMI $A_k^\top P A_k - P \preceq -\varepsilon I$ is satisfied uniformly at all finite vertices, every interpolated system inherits the same Lyapunov decrease. Hence $V(x) = x^\top P x$ decays exponentially. $\square$

B.2 Proof of Operator Identifiability (Proposition 4.1)

We prove the result for the coordinate-invariant input-output operator. The direct term $D_\theta(c)$ is not part of the state-coordinate ambiguity, but it is part of the one-step input-output map; therefore the statistical bound is stated for the extended Markov operator including D . If D is fixed by the model class, the D -term below simply drops out.

Let $\mathcal{C} \subset \mathbb{R}^{d_c}$ be compact, let $\Theta \subset \mathbb{R}^d$ be the compact trainable parameter set, and write

$$G_\theta(c) = (A_\theta(c), B_\theta(c), C_\theta(c), D_\theta(c)).$$

All matrix norms are operator norms unless $\|\cdot\|_F$ is written. Define

$$M_B := \sup_{\theta, c} \|B_\theta(c)\|, \quad M_C := \sup_{\theta, c} \|C_\theta(c)\|, \quad M_D := \sup_{\theta, c} \|D_\theta(c)\|,$$

$$D_\Theta := \sup_{\theta, \theta' \in \Theta} \|\theta - \theta'\|_2.$$

The tanh-bounded adapter is Lipschitz on $\Theta \times \mathcal{C}$. Let

$$L_A := \sup_{\theta \neq \theta', c} \frac{\|A_\theta(c) - A_{\theta'}(c)\|}{\|\theta - \theta'\|_2}, \quad L_B, L_C, L_D \text{ be defined analogously.}$$

These constants are finite because Θ and $\mathcal{C}$ are compact, the adapter is a finite composition of affine maps and tanh, and $\|\tanh'\|_\infty \leq 1$. Let n_x, n_u, n_y denote the state, input, and output dimensions.

AQS constants. Affine quadratic stability gives $P = P^\top \succ 0$ and $\varepsilon > 0$ such that, for every $\theta \in \Theta$ and $c \in \mathcal{C}$,

$$A_\theta(c)^\top P A_\theta(c) - P \preceq -\varepsilon I. \quad (24)$$

Let

$$\lambda_- := \lambda_{\min}(P), \quad \lambda_+ := \lambda_{\max}(P), \quad \rho := \sqrt{1 - \varepsilon/\lambda_+} \in (0, 1), \quad M_A := \sqrt{\lambda_+/\lambda_-}.$$

The exogenous inputs $u_t \in \mathbb{R}^{n_u}$ and output noise $\xi_t \in \mathbb{R}^{n_y}$ are centered and sub-Gaussian:

$$\|u_t\|_{\psi_2} \leq K_u, \quad \|\xi_t\|_{\psi_2} \leq K_\xi.$$

For the bounded squared-loss theorem used below, assume the synthetic identification protocol has almost-sure radii $\|u_t\|_2 \leq R_u$ and $\|\xi_t\|_2 \leq R_\xi$. For unbounded sub-Gaussian inputs, the same proof applies after the standard truncation argument with R_u, R_ξ replaced by the corresponding high-probability sub-Gaussian radii.

For a frozen context c , define

$$H_{\theta, k}(c) := C_\theta(c) A_\theta^k(c) B_\theta(c), \quad k \geq 0,$$

and the extended Markov operator

$$\tilde{\mathcal{H}}_\theta(c) := (D_\theta(c), H_{\theta, 0}(c), H_{\theta, 1}(c), \dots).$$

The well-specified observation model is

$$y_t = D_{\theta_*}(c) u_t + \sum_{k=0}^{\infty} H_{\theta_*, k}(c) u_{t-k-1} + \xi_t, \quad \theta_* \in \Theta. \quad (25)$$

The context is frozen for the local Markov operator in (25). If contexts vary inside the recurrence, the identified object is instead the time-varying product $C(c_t) A(c_t) \cdots A(c_{t-k+1}) B(c_{t-k})$.

Persistent excitation is required at the same operator level: there exists $\kappa_{\text{pe}} > 0$ such that, for every square-summable measurable sequence $\{G_k(c)\}_{k \geq -1}$,

$$\mathbb{E} \left\| G_{-1}(c) u_t + \sum_{k=0}^{\infty} G_k(c) u_{t-k-1} \right\|_2^2 \geq \kappa_{\text{pe}} \mathbb{E}_c \left[\|G_{-1}(c)\|_F^2 + \sum_{k=0}^{\infty} \|G_k(c)\|_F^2 \right]. \quad (26)$$

Lemma B.1 (Coordinate-invariant equivalence class). *Let $\mathcal{V}$ be the context index set. For two LPV realizations (A, B, C, D) and (A', B', C', D') , define*

$$(A, B, C, D) \sim (A', B', C', D')$$

if and only if $D(v) = D'(v)$ and

$$C(v)A^k(v)B(v) = C'(v)A'^k(v)B'(v), \quad \forall v \in \mathcal{V}, k \in \mathbb{N}_0.$$

Then $\sim$ is an equivalence relation, and each equivalence class is uniquely represented by the extended Markov operator $\tilde{\mathcal{H}}(v) = (D(v), \{C(v)A^k(v)B(v)\}_{k \geq 0})$.

Proof. Reflexivity and symmetry are immediate. Transitivity follows because equality of matrices is transitive for every v and k . Hence the quotient identifies exactly those state-coordinate realizations that induce the same input-output Markov parameters. Conversely, equality of input-output maps for all finitely supported inputs identifies $D(v)$ from the instantaneous response and identifies $C(v)A^k(v)B(v)$ by applying an impulse at lag $k + 1$. Under the stochastic experiment, the same implication holds in L_2 by persistent excitation (26). Thus the identifiable object is the structural equivalence class represented by $\tilde{\mathcal{H}}(v)$, not a particular choice of state coordinates. $\square$

Lemma B.2 (AQS implies geometric Markov decay). *For every $\theta \in \Theta$, $c \in \mathcal{C}$, and $k \geq 0$,*

$$\|A_\theta^k(c)\| \leq M_A \rho^k, \quad \|H_{\theta,k}(c)\| \leq M_C M_A M_B \rho^k. \quad (27)$$

The same bound holds for time-varying products $\Phi_{\theta,t,k} = A_\theta(c_t)A_\theta(c_{t-1}) \cdots A_\theta(c_{t-k+1})$.

Proof. Let $\|x\|_P^2 = x^\top P x$. From (24),

$$\|A_\theta(c)x\|_P^2 \leq \|x\|_P^2 - \varepsilon \|x\|_2^2 \leq (1 - \varepsilon/\lambda_+) \|x\|_P^2 = \rho^2 \|x\|_P^2.$$

Iterating gives $\|A_\theta^k(c)x\|_P \leq \rho^k \|x\|_P$, and converting between the Euclidean and P -norms yields $\|A_\theta^k(c)\| \leq \sqrt{\lambda_+/\lambda_-} \rho^k = M_A \rho^k$. The common Lyapunov matrix P gives the identical estimate for arbitrary time-varying products. Multiplying by the uniform B - and C -bounds gives (27). $\square$

Lemma B.3 (Geometric decay implies exponential β -mixing). *Let $Z_t = (c_t, u_t, \xi_t, y_t)$ be the stationary process generated by (25). Suppose the exogenous process (c_t, u_t, ξ_t) is a causal process of innovations with density bounded by $K_p < \infty$ and absolute-regularity coefficients $\beta_{\text{exo}}(q) \leq b_0 e^{-a_0 q}$; the independent-exogenous case is obtained by taking $b_0 = 0$. Then Z_t is exponentially β -mixing; there exist constants $b_\beta < \infty$ and $\rho_\beta \in (0, 1)$, depending only on $(M_A, M_B, M_C, M_D, \rho, R_u, R_\xi, b_0, a_0, K_p, n_u, n_y, d_c)$, such that*

$$\beta_Z(q) \leq b_\beta \rho_\beta^q, \quad q \geq 1. \quad (28)$$

One admissible choice is

$$\rho_\beta := \max\{\rho^{1/2}, e^{-a_0/2}\}, \quad b_\beta := b_0 + C_{\text{PT}}(K_p, n_u, n_y, d_c) \left(1 + \frac{M_C M_A M_B R_u}{1 - \rho} + M_D R_u + R_\xi \right)$$

where C_{PT} is the finite constant in the Pham–Tran absolute-regularity theorem for Volterra/Bernoulli-shift processes [29].

Proof. Let $y_t^{(q)}$ be the output obtained by truncating (25) to inputs no older than q lags. By Lemma B.2,

$$\|y_t - y_t^{(q)}\|_2 \leq \sum_{k > q} \|H_{\theta,k}(c_t)\| \|u_{t-k-1}\|_2 \leq \frac{M_C M_A M_B R_u}{1 - \rho} \rho^{q+1}. \quad (29)$$

Thus Z_t is a causal Bernoulli-shift, equivalently a Volterra process, whose remote-past influence is exponentially summable. The theorem of Pham and Tran [29] states that a Volterra process with bounded-density innovations and summable kernel tails is absolutely regular, with mixing coefficients controlled by the innovation mixing coefficient plus the kernel tail. Combining $\beta_{\text{exo}}(q) \leq b_0 e^{-a_0 q}$ with (29) and enlarging constants gives (28). $\square$

Lemma B.4 (Dependent Rademacher complexity). *For $z_t = (c_t, \{u_{t-j}\}_{j \geq 0}, y_t)$, define*

$$f_\theta(z_t) = D_\theta(c_t)u_t + \sum_{k=0}^{\infty} H_{\theta,k}(c_t)u_{t-k-1}, \quad \ell_\theta(z_t) := \|y_t - f_\theta(z_t)\|_2^2.$$

Let $\mathcal{L} := \{\ell_\theta : \theta \in \Theta\}$. Define

$$B_f := \left(M_D + \frac{M_C M_A M_B}{1 - \rho} \right) R_u, \quad B_y := B_f + R_\xi, \quad B_\ell := 4B_y^2,$$

and

$$L_f := R_u \left[L_D + \frac{L_C M_A M_B + M_C M_A L_B}{1 - \rho} + \frac{M_C M_B L_A M_A^2}{(1 - \rho)^2} \right].$$

Finally let

$$\Gamma_\beta := 1 + 2 \sum_{q=1}^{\infty} \beta_Z(q) \leq 1 + \frac{2b_\beta \rho_\beta}{1 - \rho_\beta},$$

$$J_d := \int_0^1 \sqrt{d \log \left(1 + \frac{3L_f D_\Theta}{B_f s} \right)} ds.$$

Then the empirical Rademacher complexity of the squared-loss class satisfies

$$\mathfrak{R}_N^\beta(\mathcal{L}) \leq \frac{C_{\text{rad}}}{\sqrt{N}}, \quad (30)$$

where

$$C_{\text{rad}} := 96 B_y B_f \sqrt{n_y \Gamma_\beta} J_d.$$

Proof. The geometric series in Lemma B.2 gives $\|f_\theta(z_t)\|_2 \leq B_f$ and $\|y_t\|_2 \leq B_y$, hence $0 \leq \ell_\theta \leq B_\ell$. For any θ, θ' ,

$$\begin{aligned} \|H_{\theta,k}(c) - H_{\theta',k}(c)\| &\leq \left(L_C M_A M_B \rho^k + M_C M_A L_B \rho^k \right. \\ &\quad \left. + M_C M_B L_A M_A^2 k \rho^{k-1} \right) \|\theta - \theta'\|_2, \end{aligned}$$

and summing over $k \geq 0$, adding the D -term, and multiplying by R_u gives

$$\|f_\theta(z_t) - f_{\theta'}(z_t)\|_2 \leq L_f \|\theta - \theta'\|_2.$$

Thus the covering number of each output coordinate satisfies

$$\log \mathcal{N}(\epsilon, \{f_{\theta,j}\}, L_2) \leq d \log \left(1 + \frac{3L_f D_\Theta}{\epsilon} \right).$$

Dudley's entropy integral gives

$$\mathfrak{R}_N(\{f_{\theta,j}\}) \leq \frac{12B_f}{\sqrt{N}} J_d.$$

Summing over n_y output coordinates and using Cauchy-Schwarz gives the $\sqrt{n_y}$ factor. Since $\ell_\theta(z) = \|y - f_\theta(z)\|_2^2$ is $4B_y$ -Lipschitz in $f_\theta(z)$ on the bounded range, Talagrand's contraction principle [22, 23] transfers the predictor bound to the squared-loss class. For the dependent sequence, the independent-block coupling of Yu and the Mohri-Rostamizadeh blocking lemma for β -mixing sequences inflate the independent complexity by at most $\sqrt{\Gamma_\beta}$ [30, 31]. This yields (30). $\square$

Proposition B.1 (Excess-risk and operator convergence). *Let $\hat{\theta}$ be an empirical risk minimizer over Θ :*

$$\hat{\theta} \in \arg \min_{\theta \in \Theta} \widehat{\mathcal{R}}_N(\theta), \quad \widehat{\mathcal{R}}_N(\theta) := \frac{1}{N} \sum_{t=1}^N \ell_\theta(z_t),$$

and let $\mathcal{R}(\theta) := \mathbb{E} \ell_\theta(Z_t)$. For any $\delta \in (0, 1)$, with probability at least $1 - \delta$,

$$\mathcal{R}(\hat{\theta}) - \mathcal{R}(\theta_\star) \leq \frac{C_{\text{ex}}(\delta)}{\sqrt{N}}, \quad (31)$$

where

$$C_{\text{ex}}(\delta) := 4C_{\text{rad}} + 2B_\ell \sqrt{2\Gamma_\beta \log(2/\delta)}.$$

Moreover, the squared extended Markov-operator error obeys

$$\mathbb{E}_c \left[\|D_{\hat{\theta}}(c) - D_{\theta_*}(c)\|_F^2 + \sum_{k=0}^{\infty} \|H_{\hat{\theta},k}(c) - H_{\theta_*,k}(c)\|_F^2 \right] \leq \frac{C_{\text{ex}}(\delta)}{\kappa_{\text{pe}} \sqrt{N}}. \quad (32)$$

Proof. For bounded losses generated by an exponentially β -mixing process, the uniform convergence theorem for absolutely regular sequences gives [31, 32]

$$\sup_{\theta \in \Theta} |\mathcal{R}(\theta) - \hat{\mathcal{R}}_N(\theta)| \leq 2\mathfrak{R}_N^\beta(\mathcal{L}) + B_\ell \sqrt{\frac{2\Gamma_\beta \log(2/\delta)}{N}}$$

with probability at least $1 - \delta$. Substituting Lemma B.4 gives the right-hand side

$$\Delta_N(\delta) \leq \frac{2C_{\text{rad}} + B_\ell \sqrt{2\Gamma_\beta \log(2/\delta)}}{\sqrt{N}}.$$

Since $\hat{\theta}$ minimizes the empirical risk and $\theta_* \in \Theta$,

$$\mathcal{R}(\hat{\theta}) - \mathcal{R}(\theta_*) \leq 2\Delta_N(\delta) \leq \frac{C_{\text{ex}}(\delta)}{\sqrt{N}},$$

which proves (31).

It remains to connect the statistical risk to the coordinate-invariant operator. Because the noise is centered and independent of the inputs, and the model is well specified, the noise variance cancels in the excess risk:

$$\mathcal{R}(\theta) - \mathcal{R}(\theta_*) = \mathbb{E} \left\| \Delta D_\theta(c_t)u_t + \sum_{k=0}^{\infty} \Delta H_{\theta,k}(c_t)u_{t-k-1} \right\|_2^2.$$

Here $\Delta D_\theta = D_\theta - D_{\theta_*}$ and $\Delta H_{\theta,k} = H_{\theta,k} - H_{\theta_*,k}$. Applying persistent excitation (26) with $G_{-1} = \Delta D_\theta$ and $G_k = \Delta H_{\theta,k}$ yields

$$\mathcal{R}(\theta) - \mathcal{R}(\theta_*) \geq \kappa_{\text{pe}} \mathbb{E}_c \left[\|\Delta D_\theta(c)\|_F^2 + \sum_{k=0}^{\infty} \|\Delta H_{\theta,k}(c)\|_F^2 \right].$$

Taking $\theta = \hat{\theta}$ and using (31) proves (32). $\square$

Combining Lemma B.1 with Proposition B.1 proves Proposition 4.1. The internal matrices are identifiable only modulo the structural equivalence relation $\sim$, while the observable Markov operator $\mathcal{H}$ is identifiable. Under squared loss, both the excess prediction risk and the squared extended Markov-operator error are bounded by $C/\sqrt{N}$ with high probability, where C is an explicit function of the AQS certificate (P, ε) , representation bounds (M_A, M_B, M_C, M_D) , excitation and tail constants $(K_u, K_\xi, R_u, R_\xi, \kappa_{\text{pe}})$, the parameter dimension d , the tanh-parameterization Lipschitz constants (L_A, L_B, L_C, L_D) , the mixing constants (b_β, ρ_β) , and the confidence level δ , but not of N . The unsquared Frobenius norm is controlled by taking a square root of (32); the rate stated in Proposition 4.1 is the squared-loss estimation rate.

C Experiment Details

C.1 Training Details

Training adapts the frozen Hydra-BERT backbone to the masked diffusion objective on an internal Hydra corpus constructed from the source mixture in Table 5. The reported train and validation portions contain 4,295,110,656 tokens in total: 3,537,149,952 training tokens and 757,960,704 validation tokens. The validation split is used for monitoring and reporting the held-out losses in the experiments, while all model updates use only the training split.

Table 5: Source mixture used to construct the Hydra train/validation corpus. Percentages are the relative shard-pool weights of the reported training runs and normalize at sampling.

Source	Ratio (%)
common-pile StackExchange	10.00
common-pile LibreTexts	1.67
common-pile YouTube	22.50
common-pile PubMed	7.50
common-pile CCCC	10.00
common-pile Project Gutenberg	5.00
common-pile arXiv papers	7.50
common-pile News	1.67
common-pile DOAB	15.00
common-pile Pressbooks	1.67
WikiText-103 raw	2.50
common-pile Data Provenance Initiative	8.33
common-pile Wikimedia	8.33
common-pile arXiv abstracts	3.33

For all reported experiments, only the MaRK adapter and timestep-embedding parameter groups are trainable. The Hydra backbone parameters and masked-LM classification head remain fixed, isolating the effect of dynamic operator modulation. The objective is the masked-token diffusion negative log-likelihood with CART weighting enabled using parameter $p = 0.45$. Training uses Distributed Data Parallel (DDP) on two NVIDIA GH200 GPUs with bfloat16 precision, gradient checkpointing, high-precision matrix multiplication, and gradient clipping at norm 1.0.

Table 6: Training settings shared by all three MaRK kernels. Values are taken from the training configurations and the corresponding optimization implementation.

Setting	Value
Trainable parameters	MaRK adapters and timestep embeddings only
Backbone	Frozen 23-layer Hydra-BERT SSM
Sequence length	4096 tokens
Epochs	3
Optimizer	Fused AdamW, $\beta = (0.9, 0.999)$, weight decay 0.01
MaRK learning rate	6.0×10^{-4}
Learning-rate schedule	Step-wise linear warmup, then cosine decay
Minimum LR ratio	0.1
Precision	bfloat16
Training strategy	Distributed Data Parallel
Workers	8
Matrix multiply precision	high
Hydra hidden size / vocabulary	768 / 30,522
SSM state / convolution / head dim	64 / 7 / 64
Expansion / chunk size	2 / 256

The variants are initialized from their corresponding kernel-specific checkpoints. Apart from the kernel geometry, all reported variants use the same batch size, warmup length, total training horizon, optimizer, and base learning rate. The 6.0×10^{-4} learning rate is linearly warmed up for 720 steps and then cosine-decayed over the remaining steps of the 14,400-step schedule to a minimum value of $0.1 \times$ the base rate. The kernel-specific entries in Table 7 specify only the adapter geometry: Chebyshev uses a degree-5 polynomial basis, while DCT uses a 256-point basis grid with 8 retained frequencies.

Table 7: Kernel-specific training settings. Shared kernel dimensions are rank $r = 2$, adapter MLP dimension 256, and timestep embedding dimension 128.

Kernel	Batch size	Warmup / total steps	Kernel-specific parameters
Hypernet	112	720 / 14,400	–
Chebyshev	112	720 / 14,400	Polynomial degree = 5
DCT	112	720 / 14,400	Timepoints = 256, frequencies = 8

C.2 Modulation Bounds and CART Settings

The bounded shifts and scales of Section 2.1 use the magnitudes in Table 8. Diagonal parameters use $\beta_{\Theta} \tanh(\cdot)$, with the A -shift bound capped at $\beta_A \leq 0.5$ and the remaining magnitudes learned; dense parameters use α_{Φ} and β_{Φ} on the low-rank factors. Every magnitude is initialized to 0.2. The kernel dimensions (low-rank size, adapter MLP, timestep embedding, Chebyshev degree, and DCT frequency budget) are listed in Table 7.

Table 8: Modulation bounds and CART settings, identical across the three kernels.

Parameter	Value
A -shift bound β_A	≤ 0.5
Scale and shift initial magnitude α, β	0.2 (learned)
CART p / scale	0.45 / 1.0

The A -shift bound is the only constraint that enters the stability certificate, so it is capped while the remaining magnitudes are learned freely; B , C , and D never enter the Lyapunov condition because they are not raised to powers. Because $A < 0$ and $\Delta > 0$, every discrete eigenvalue stays in $(0, 1)$ under any finite bound, so the modulated recurrence is Schur-stable for each individual parameter value and the Markov sequence always decays. The finite bound buys a uniform positive margin rather than stability itself: the worst case sits at the vertex $\delta = -\beta_A$, and as β_A grows, that eigenvalue approaches 1 from below while the margin $\varepsilon = 1 - \max_i \bar{A}_i^2$ shrinks toward zero. An unbounded shift would therefore not destabilize any single realization, but it would remove the uniform guarantee that Appendix B relies on, which is why the bound is kept finite. The bounds themselves were set conservatively to keep the modulated operators near the frozen backbone, and the CART sharpness $p = 0.45$ follows Dream.

C.3 Evaluation Protocol and Statistics

Every language-model number in this paper is a CART-weighted validation loss, read as `weighted_nll` from the evaluation harness; perplexity is only the exponential of that quantity and is not the reported metric. Each cell is a mean over 10 seeds for the accuracy tables (Tables 1 and 2) and over 15 seeds for the inference table (Table 3), with seed i fixed at $42 + 100i$. Intervals are standard-error-of-the-mean bands over seeds (sample standard deviation with $\text{ddof} = 1$ divided by $\sqrt{n}$): Tables 1 and 2 report the normal approximation (mean $\pm 1.96 \times \text{SEM}$ over 10 seeds), while the inference table (Table 3) reports the Student- t interval with 14 degrees of freedom over 15 seeds. The synthetic LPV curves in Figure 2 use the same seed and interval convention.

Validation consumes the global random-number stream on every batch, sampling both the diffusion timestep and the masking pattern, and building the model also draws from it because Hydra initializes its discretization bias randomly. The harness therefore captures the RNG state immediately after the model is built and restores it before each dataset, so the CART suite reproduces the leave-one-out draws and the WikiText loss matches the leave-one-out full row to the bfloat16 run-to-run noise. Because the largest validation split (arXiv) contains tens of thousands of packed sequences at length 4096, its evaluation is capped at 500 validation batches, while smaller datasets are evaluated in full.

C.4 AQS Certificate Evaluation

We evaluate the AQS certificate by directly checking discrete-time stability margins implied by the ZOH discretization for the learned diagonal recurrence. Using the identity Lyapunov function

$P = I$, this yields a closed-form certificate that avoids SDP solving. We load the pretrained diagonal log-parameters $A_{\log}^{(\ell)}$ from the 23-layer Hydra checkpoint and evaluate the worst-case (least stable) parameter vertex by monotonicity of the map $A_{\log} \mapsto \bar{A} = \exp(-\exp(A_{\log})\Delta)$ under the bounded shift $A_{\log} \pm \beta_A$, using $\Delta_{\min} = 0.001$:

Algorithm 4 Constructive AQS Certificate via $P = I$ Lyapunov Function

```

1: Input: Pretrained  $A_{\log}^{(\ell)}$  for layers  $\ell = 1, \dots, 23$ ; bound  $\beta_A = 0.5$ ;  $\Delta_{\min} = 0.001$ 
2: Set  $P \leftarrow I$  ▷ Identity matrix (valid Lyapunov function for diagonal  $A$  with  $\rho < 1$ )
3: for each layer  $\ell$ , each head  $i$  do
4:    $A_{\log,i}^{\text{worst}} \leftarrow A_{\log,i}^{(\ell)} - \beta_A$  ▷ Worst-case vertex (least decay)
5:    $A_i \leftarrow -\exp(A_{\log,i}^{\text{worst}})$  ▷ Strictly negative
6:    $\bar{a}_i \leftarrow \exp(A_i \cdot \Delta_{\min})$  ▷ Discrete eigenvalue  $\in (0, 1)$ 
7: end for
8:  $\rho_{\max}^{(\ell)} \leftarrow \max_i |\bar{a}_i|$  ▷ Spectral radius at worst-case vertex
9:  $\varepsilon_\ell \leftarrow 1 - (\rho_{\max}^{(\ell)})^2$  ▷ Lyapunov stability margin
10: Verify:  $\varepsilon_\ell > 0$  for all  $\ell$  ▷  $\bar{A}^\top I \bar{A} - I \prec 0$  (neg. definite)
11: Return  $P = I, \{\varepsilon_\ell\}_{\ell=1}^{23}$  ▷ AQS Certificate with per-layer margins

```

The procedure is conservative in three ways: it fixes the Lyapunov function to $P = I$, evaluates the least stable vertex $\delta = -\beta_A$, and uses the smallest discretization $\Delta_{\min} = 0.001$, which places the discrete eigenvalue closest to one. Because β_A is shared across the bounded kernels, the certificate depends only on the pretrained $A_{\log}$ and not on the adapter geometry, so a single pass over the backbone covers all three variants. The reported margins are the distance of the worst-case eigenvalue from the unit circle.

C.5 Synthetic LPV Identifiability Algorithms

For identifiability tests, both ground-truth generator models and target estimators use diagonal LPV-SSMs with state dimension $d_{\text{state}} = 4$, factorization rank $r = 2$, sequence length $L = 20$, and datasets $N \in \{50, 100, 200, 400, 800, 1600\}$ over 15 random seeds. Inputs are sampled as $u \sim \mathcal{N}(0, 1)$ and diffusion coordinates as $\tau \sim \mathcal{U}(0, 1)$; scalar outputs are generated by the ground-truth system and corrupted with Gaussian noise $\sigma = 0.1$. Chebyshev and DCT use a fixed readout $C = \mathbf{1}$ to remove final rotation/scaling ambiguity, while the Hypernet synthetic kernel also modulates $C(\tau)$ with bounded low-rank scale/shift terms. This choice is without loss of generality for scalar readouts: any nonzero scalar C can be absorbed by a linear reparameterization of the latent state, so fixing $C = \mathbf{1}$ isolates operator recovery rather than an arbitrary output scaling.

The continuous matrices returned by Algorithms 5–7 are rolled out with the same scalar ZOH discretization used in the scripts:

$$\bar{A}(\tau) = \exp(A(\tau)\Delta t), \quad \bar{B}(\tau) = \frac{\bar{A}(\tau) - 1}{A(\tau) + 10^{-7}} \odot B(\tau), \quad \Delta t = 0.1.$$

The synthetic sequence is then generated by

$$x_s = \bar{A}(\tau_s) \odot x_{s-1} + \bar{B}(\tau_s)u_s, \quad y_s = C(\tau_s)^\top x_s.$$

Estimators are initialized from noisy copies of the ground-truth parameters and trained with Mean-Squared Error (MSE). Recovery is measured by frozen-time Markov operator error on a 200-point τ grid over lags $k = 0, \dots, 14$, using $M_k(\tau) = C(\tau)^\top (\bar{A}^k(\tau) \odot \bar{B}(\tau))$; for Chebyshev and DCT, $C(\tau) = \mathbf{1}$ is constant.

The Hypernet generator is the most flexible synthetic parameterization: a sinusoidal embedding of τ is mapped by a four-layer SiLU MLP to bounded shifts of the recurrence, input map, and readout. The negative exponential form enforces $A(\tau) < 0$ componentwise, and sigmoid/tanh bounds keep the modulation amplitude controlled. Unlike the basis-kernel variants below, Hypernet returns a time-varying $C(\tau)$ in addition to $A(\tau)$ and $B(\tau)$.

Algorithm 5 Hypernet Synthetic LPV-SSM

1: **Input:** Diffusion coordinate $\tau \in [0, 1]$
2: **Parameters:** $A_{\log_base}$, low-rank base U_{base}, V_{base} , scalars $\beta_A^{\text{param}}, \alpha_B, \beta_B, \alpha_C, \beta_C$, four-layer SiLU MLP, linear heads
3: $\text{cond}(\tau) \leftarrow [\sin(2\pi m\tau), \cos(2\pi m\tau)]_{m=0}^{63}$ $\triangleright$ 128-dimensional conditioning signal
4: $h \leftarrow \text{MLP}_{4 \times \text{SiLU}}(\text{cond}(\tau))$
5: $\text{uvflat}(q) \leftarrow \text{vec}(U_q V_q)_{1:d_{\text{state}}}$, where q is split into rank- r factors (U_q, V_q) $\triangleright$ matches the low-rank head flattening
6: $\beta_A^{\text{eff}} \leftarrow 0.5 \sigma(\beta_A^{\text{param}})$, $a_{\text{shift}} \leftarrow W_A h$ $\triangleright$ bounds the recurrence modulation
7: $A(\tau) \leftarrow -\exp(A_{\log_base} + \beta_A^{\text{eff}} \tanh(a_{\text{shift}}))$
8: $b_{\text{base}} \leftarrow \text{vec}(U_{\text{base}} V_{\text{base}})_{1:d_{\text{state}}}$
9: $b_{\text{scale}}(\tau) \leftarrow 1 + \alpha_B \tanh(\text{uvflat}(W_{B,\text{scale}} h))$, $b_{\text{shift}}(\tau) \leftarrow \beta_B \tanh(\text{uvflat}(W_{B,\text{shift}} h))$ $\triangleright$ bounded scale/shift for B
10: $B(\tau) \leftarrow b_{\text{base}} \odot b_{\text{scale}}(\tau) + b_{\text{shift}}(\tau)$
11: $c_{\text{scale}}(\tau) \leftarrow 1 + \alpha_C \tanh(\text{uvflat}(W_{C,\text{scale}} h))$, $c_{\text{shift}}(\tau) \leftarrow \beta_C \tanh(\text{uvflat}(W_{C,\text{shift}} h))$
12: $C(\tau) \leftarrow \mathbf{1} \odot c_{\text{scale}}(\tau) + c_{\text{shift}}(\tau)$ $\triangleright$ learned readout for the Hypernet variant
13: **Return** $A(\tau), B(\tau), C(\tau)$

Hypernet is the only synthetic generator with a learned, time-varying readout, so its recovery error also reflects the readout path and not just the recurrence and input maps. It is therefore the hardest of the three synthetic targets, and the basis kernels are compared against it rather than against an easier diagonal reference.

The Chebyshev generator uses a fixed polynomial basis after mapping τ to $z \in [-1, 1]$. All time variation enters through bounded Chebyshev shifts of the log-recurrence and of the low-rank factors defining $B(\tau)$. The readout is fixed to $C = \mathbf{1}$, so the reported recovery error measures the induced Markov operator rather than an arbitrary rescaling of the output map.

Algorithm 6 Chebyshev Synthetic LPV-SSM

1: **Input:** Diffusion coordinate $\tau \in [0, 1]$
2: **Parameters:** $A_{\log_base}, U_{\text{base}}, V_{\text{base}}$, coefficients $A_{\text{coeffs}}, U_{\text{coeffs}}, V_{\text{coeffs}}$, bounds β_A, β_B , basis count $K_{\text{basis}} = 3$
3: $z \leftarrow 2\tau - 1$ $\triangleright$ Chebyshev domain
4: $T_0(z) \leftarrow 1, T_1(z) \leftarrow z$
5: **for** $k = 2$ **to** $K_{\text{basis}} - 1$ **do**
6: $T_k(z) \leftarrow 2z \cdot T_{k-1}(z) - T_{k-2}(z)$
7: **end for**
8: $A_{\text{shift}} \leftarrow \sum_{k=0}^{K_{\text{basis}}-1} T_k(z) A_{\text{coeffs}}^{(k)}$ $\triangleright$ basis modulation of the log recurrence
9: $U_{\text{shift}} \leftarrow \sum_{k=0}^{K_{\text{basis}}-1} T_k(z) U_{\text{coeffs}}^{(k)}$, $V_{\text{shift}} \leftarrow \sum_{k=0}^{K_{\text{basis}}-1} T_k(z) V_{\text{coeffs}}^{(k)}$
10: $A(\tau) \leftarrow -\exp(A_{\log_base} + \beta_A \tanh(A_{\text{shift}}))$ $\triangleright$ negative diagonal continuous dynamics
11: $U(\tau) \leftarrow U_{\text{base}} + \beta_B \tanh(U_{\text{shift}})$, $V(\tau) \leftarrow V_{\text{base}} + \beta_B \tanh(V_{\text{shift}})$
12: $B(\tau) \leftarrow U(\tau) V(\tau)$, $C \leftarrow \mathbf{1}$ $\triangleright$ fixed readout removes scaling ambiguity
13: **Return** $A(\tau), B(\tau), C$

The synthetic Chebyshev generator uses a degree-3 basis to keep the target low-dimensional, whereas the language-model Chebyshev adapter uses degree 5 (Table 7). The synthetic target is therefore a smoother, lower-capacity member of the same polynomial family.

The DCT generator replaces the Chebyshev basis with cosine modes and a learned monotone spectral decay. The normalization constants match the $L_{\text{time}} = 256$ DCT-II basis used in the experiment, while tanh-bounded shifts keep the recurrence and input map controlled. As in the Chebyshev setting, $C = \mathbf{1}$ is fixed so the comparison focuses on the recovered operator.

Algorithm 7 Discrete Cosine Transform (DCT) Synthetic LPV-SSM

```

1: Input: Diffusion coordinate  $\tau \in [0, 1]$ 
2: Parameters:  $A_{\log\_base}, U_{base}, V_{base}$ , spectral coefficients  $A_{spec}, U_{spec}, V_{spec}$ , bounds  $\beta_A, \beta_B$ ,
   bandwidth  $\alpha, n_{freqs} = 8, L_{time} = 256$ 
3: for  $k = 0$  to  $n_{freqs} - 1$  do
4:    $w_k \leftarrow (k + 1)^{-(\text{softplus}(\alpha) + 10^{-4})}$  ▷ learned spectral attenuation
5:    $c_k \leftarrow 1/\sqrt{L_{time}}$  if  $k = 0$ , else  $\sqrt{2/L_{time}}$  ▷ DCT-II normalization
6:    $b_k(\tau) \leftarrow c_k \cos(\pi k \tau)$ 
7: end for
8:  $A_{shift} \leftarrow \sum_{k=0}^{n_{freqs}-1} b_k(\tau) w_k A_{spec}^{(k)}$  ▷ attenuated spectral recurrence shift
9:  $U_{shift} \leftarrow \sum_{k=0}^{n_{freqs}-1} b_k(\tau) w_k U_{spec}^{(k)}, \quad V_{shift} \leftarrow \sum_{k=0}^{n_{freqs}-1} b_k(\tau) w_k V_{spec}^{(k)}$ 
10:  $A(\tau) \leftarrow -\exp(A_{\log\_base} + \beta_A \tanh(A_{shift}))$  ▷ negative diagonal continuous dynamics
11:  $U(\tau) \leftarrow U_{base} + \beta_B \tanh(U_{shift}), \quad V(\tau) \leftarrow V_{base} + \beta_B \tanh(V_{shift})$ 
12:  $B(\tau) \leftarrow U(\tau)V(\tau), \quad C \leftarrow \mathbf{1}$  ▷ fixed readout isolates operator recovery
13: Return  $A(\tau), B(\tau), C$ 

```

The synthetic DCT generator keeps the basis length and frequency count of the language-model adapter so that the target uses the same bandlimited parameterization it is meant to probe. The learned attenuation only reshapes how quickly the high-frequency coefficients decay, so a miss in recovery is attributable to the bandlimited basis rather than to a mismatch in basis definition.