WHAT ALGORITHMS DO PRE-TRAINED LLMS USE FOR HMM PREDICTION?

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) exhibit a striking ability to predict observations generated by Hidden Markov Models (HMMs) through in-context learning (ICL), yet the algorithms underlying this capability remain unclear. We investigate this question through a three-stage pipeline combining behavioral comparisons, constructive Transformer theory, and causal analysis of internal representations. First, we compare LLM behavior with a suite of candidate algorithms, including classical inference methods and algorithms motivated by empirical observations. Second, we establish theoretical connections among these algorithmic classes, construct in-context Transformer implementations, and validate a finite-window mechanism in a small trained Transformer. Third, we probe pre-trained LLMs for internal representations corresponding to the candidate algorithms. We identify low-dimensional linear representations that causally influence model predictions and whose strength tracks empirical ICL performance. These representations vary systematically across HMM regimes, with finite-window, gradient-descent-like computations predominating in most regimes. Together, our results connect the in-context behavior of LLMs to their internal algorithmic representations, providing a foundation for understanding how LLMs predict sequences governed by latent dynamics.

1 INTRODUCTION

Hidden Markov Models describe systems in which observations are generated by latent Markovian dynamics and arise in applications ranging from animal decision-making to ecological and climate processes (Glennie et al., 2023; McClintock et al., 2020; Zucchini & Guttorp, 1991). Recent work has shown that pre-trained LLMs can predict HMM-generated sequences from context: given a single emission sequence as a prompt, their next-token predictions approach Bayes-optimal predictions as context length increases, often outperforming classical inference algorithms applied to the same observations (Dai et al., 2026). Understanding this capability could help clarify how LLMs extract predictive structure from observations of latent dynamical systems and when such predictions may be useful in scientific applications.

How LLMs achieve this behavior remains unclear, and identifying the underlying mechanisms is more challenging than in simpler sequence-prediction settings. For fully observed Markov chains and fixed-order n-gram processes, optimal prediction depends on a bounded, observable history. This structure provides relatively direct algorithmic hypotheses, such as estimating conditional probabilities from transition counts, that can guide the identification of implementing circuits (Edelman et al., 2024; Makkuva et al., 2025; Rajaraman et al., 2024). HMM prediction requires accounting for an unobserved state through noisy emissions, and relevant information can extend beyond fixed observation window. When the HMM parameters are not supplied, the predictor must also extract predictive structure from the context. Although the prediction objective is well defined, the computational strategy an LLM might use is much less obvious. HMMs therefore offer a controlled bridge between simple algorithmic tasks and partially observed inference problems that arise in scientific applications.

Several candidate algorithms have been proposed for Transformer-based sequence modeling on hidden-Markovian data, including spectral methods, belief-state tracking, and gradient-based feature learning (Hu et al., 2024; Shai et al., 2024; Hao et al., 2025). Each is consistent with some subset of the observed behavior, but it remains unresolved which mechanisms pre-trained LLMs use.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

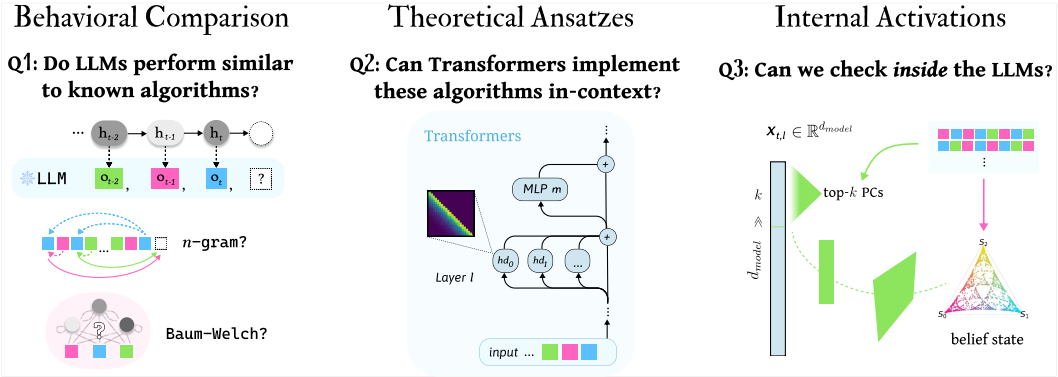


Figure 1: **Overview of our study.** We studied the behavioral comparison (Q1, Section 2), theoretical construction (Q2, Section 3), and internal causal representations (Q3, Section 4) of algorithms that LLMs might use for in-context learning HMM-generated sequences.

Our central question is: ***What algorithms underlie in-context prediction of HMM-generated sequences in LLMs, and how are these algorithms represented and used within the models?*** We investigate this question through a three-stage pipeline:

1. **Behavioral comparison.** We systematically benchmark 12 pre-trained LLMs against classical inference methods and algorithms motivated by empirical observations across 75 HMM configurations. No single baseline matches LLM behavior across all regimes, but these comparisons narrow the space of plausible mechanisms to three algorithmic classes, which we call *ansatzes*: Linear n-gram, Non-linear n-gram, and Spectral (Section 2).

2. **Theoretical constructions.** We show that Transformers are *capable* of implementing all three candidate ansatzes in context. We derive a unifying formulation in which next-token probabilities are linear functions of nonlinear features of past observations and use it to construct explicit Transformer implementations of each class. We further verify that one such mechanism, a finite-window predictor, arises naturally in a small Transformer trained on a single HMM (Section 3).

3. **Internal representations and causal use.** We introduce the *Principal Activations Probe* (PAP), a layer-wise probing-and-intervention framework that isolates low-dimensional algorithmic signals in the residual stream and tests whether they causally influence model predictions. PAP reveals that pre-trained LLMs recruit different algorithmic representations across HMM regimes, with finite-window, gradient-descent-like computations predominating in most regimes. Their strength tracks empirical ICL performance (Section 4).

2 Stage 1: Behavioral Comparison

We study how pre-trained LLMs perform in-context learning on synthetic HMM sequences. We first review key HMM properties (Section 2.1) and theoretical algorithms (Section 2.2). We then empirically demonstrate that the performance of LLMs consistently converges to the theoretical optimum, with nuances when compared against theoretical algorithms (Section 2.3).

2.1 Background: Hidden Markov Model

At each time step t, an HMM's *hidden state* $H_t \in \mathcal{H}$ emits an *observation* O_t from an *emission distribution*, then transitions to H_{t+1} according to *transition probabilities*. Implicit in this description are three standard assumptions: the *Markov property*, that each state depends only on its immediate predecessor; *output independence*, that each observation depends only on the current hidden state; and *stationarity*, that neither the transition nor the emission probabilities depend on t.

In the finite-alphabet setting, take $\mathcal{H} = \{1, \ldots, M\}$ and $\mathcal{O} = \{1, \ldots, N\}$. An HMM is specified by $\boldsymbol{\lambda} = (\boldsymbol{\pi}, \mathbf{A}, \mathbf{B})$, where $\boldsymbol{\pi} \in \mathbb{R}^M$ is the initial state distribution, $\mathbf{A} \in \mathbb{R}^{M \times M}$ the transition matrix, and $\mathbf{B} \in \mathbb{R}^{M \times N}$ the emission matrix. Under standard conditions, the chain converges to a unique *stationary distribution* $\boldsymbol{\mu} = \boldsymbol{\mu}\mathbf{A}$ (Ephraim & Merhav, 2002), governing long-run behavior of the hidden state and central to prediction and parameter learning. The rate of convergence is the *mixing rate*: the hidden-state distribution approaches $\boldsymbol{\mu}$ geometrically, with smaller rate meaning faster mixing. For finite-alphabet HMMs this rate equals λ_2, the second-largest eigen-

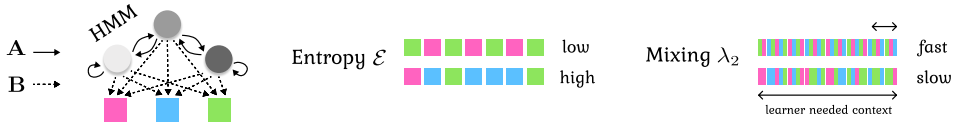


Figure 2: Properties of HMMs that impact the difficulty of next observation prediction task.

Class	Prediction approach
Oracle	Forward recursion with the true parameters $(\boldsymbol{\pi}, \mathbf{A}, \mathbf{B})$.
Spectral	Observation operators estimated from empirical uni-, bi-, and trigram statistics.
Linear n-gram	Linear map from the one-hot encodings of the last n observations to the next-observation distribution.
Non-linear n-gram	Linear map from the Kronecker product of the last n one-hot observation encodings to the next-observation distribution.
Baum–Welch	Prediction using HMM parameters estimated by expectation-maximization.

Table 1: HMM predictor classes benchmarked for next-observation prediction.

value of $\mathbf{A}$; we restrict to transition matrices with real positive eigenvalues (Appendix A). To quantify where an HMM sits on the deterministic-to-random spectrum, we use normalized entropy $\mathcal{E}(\mathbf{A}) = -\frac{1}{\log M}\sum_{i,j}\mu_i a_{ij}\log a_{ij}$, $\mathcal{E}(\mathbf{B},\boldsymbol{\mu}) = -\frac{1}{\log N}\sum_{j,k}\mu_j b_{jk}\log b_{jk}$, measuring stationary-distribution uncertainty in the next hidden state and in the emission; both lie in $[0,1]$, with smaller values indicating more predictable dynamics.

2.2 Setup: Candidate Algorithms to Compare Against

We benchmark pre-trained LLMs against five classes of HMM predictors, summarized in Table 1, with full specifications in Appendix B. Within the Spectral class (Hsu et al., 2012; Ma & Rodu, 2023), `Norm` and `SVD` differ in how they invert the empirical bigram statistic. For Linear n-gram predictors, `Gradient CE` and `Gradient MSE` use online gradient descent on cross-entropy and squared error, respectively, whereas `Ridge MSE` uses ridge regression. The Non-linear n-gram variants exploit the representation established in Lemma 3.1: `Kron` constructs the features explicitly, whereas `Kernel` avoids their exponential-in-n dimension through kernelization. Finally, `BW(EM)` (Baum et al., 1970) is given the number of hidden states and iteratively estimates the HMM parameters. The Oracle provides the Bayes-optimal reference.

2.3 Experiment: LLMs Empirical Performance

Experimental Setup. For each HMM configuration we sample observation sequences $\mathbf{o}_{1:T}$ from $\boldsymbol{\lambda} = (\boldsymbol{\pi}, \mathbf{A}, \mathbf{B})$ and evaluate next-observation prediction o_{t+1} given $\mathbf{o}_{1:t}$. We fix $M = 4$ and vary three controls across 75 configurations: transition and emission entropy $\mathcal{E}(\mathbf{A}), \mathcal{E}(\mathbf{B}) \in \{0,\ 0.25,\ 0.5,\ 0.75,\ 1\}$ (inducing mixing rates $|\lambda_2(\mathbf{A})| \in \{1, 0.9, 0.75, 0.5, 0\}$); and alphabet size $N \in \{2, 4, 8\}$, where $N < M$ induces hidden-state aliasing. We set $\boldsymbol{\pi} = \boldsymbol{\mu}$. Per configuration, we sample 4,096 sequences and evaluate at context lengths $\{4, 8, \ldots, 4096\}$.

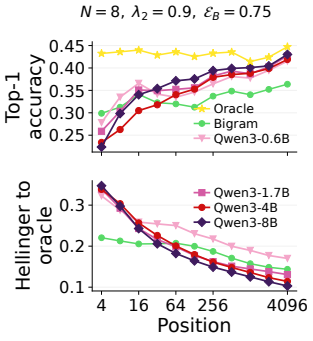


Figure 3: LLMs converges to Oracle consistently.

We report two metrics, averaged over 4,096 samples per HMM. *Accuracy* is the fraction of positions whose argmax prediction matches the truth. The Hellinger distance between predicted next-token distributions plays two roles: *Hellinger-to-oracle* measures distance to the Bayes-optimal posterior; *Hellinger-to-baseline* measures distance to each baseline. We evaluate 12 open-weight pre-trained LLMs from the Qwen, Llama, Gemma, and OLMo families (0.6B–8B parameters), without any fine-tuning. Each model sees raw observation tokens via a fixed alphabet-to-vocabulary map; at every position, we restrict the next-token logits to the observation vocabulary and renormalize via softmax.

LLMs converge to Bayes-optimal predictor. Across HMM configurations, LLM predictions approach Oracle as context length grows, and this trend is consistent across model sizes (Figure 3). Appendix F provides analysis of more model families and HMM configurations, including a discussion of cases where convergence fails or unexpected behavior arises.

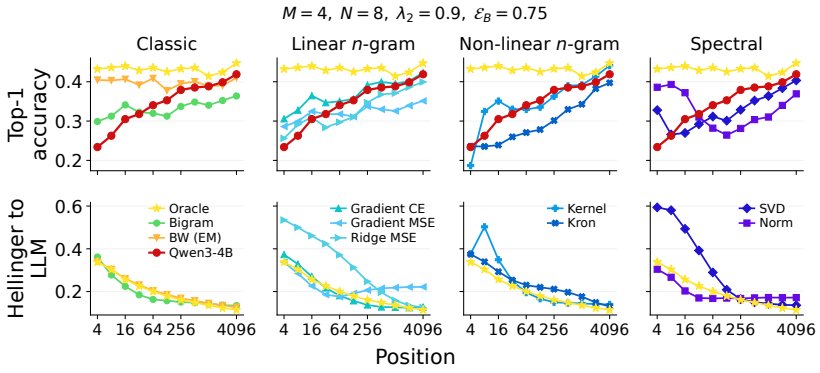


Figure 4: LLMs empirical comparisons to candidate algorithms, which are specified in Section 2.2.

LLM behavior resembles several algorithmic baselines. Figure 4 shows that Qwen steadily approaches Oracle as context length increases, a pattern shared by several baseline classes. `BW(EM)` (Baum–Welch) exhibits a different pattern, achieving strong performance even at short context lengths. It explicitly estimates HMM parameters, providing a distinct algorithmic route to prediction. We therefore focus the subsequent theoretical analysis on baselines whose context-length dependence more closely resembles Qwen's. Appendix F.3 provides detailed comparisons across all baselines.

3 Stage 2: Theoretical Constructions

In Section 2.3, we found that Linear n-gram, Non-linear n-gram, and Spectral predictors empirically resemble LLM behavior. We refer to these predictors as *ansatzes* and now study their implementation by Transformers through in-context learning.

3.1 Theoretical Results: Construction of Each Ansatz

Our constructions target the next-observation probability vector $q_{t+1} := \mathbb{P}\left(O_{t+1} \mid O_{1:t} = o_{1:t}\right)$. We begin with known HMM parameters $(\boldsymbol{\pi}, \mathbf{A}, \mathbf{B})$ and show how to express q_{t+1} using non-linear features constructed from the one-hot encodings of past observations. This representation provides the starting point for constructing the ansatzes described in Section 2.2.

Lemma 3.1. *Consider an HMM specified by $\boldsymbol{\lambda} = (\boldsymbol{\pi}, \mathbf{A}, \mathbf{B})$. Let $\mathbf{A}_k := \operatorname{diag}(\mathbf{B}[:, k])\, \mathbf{A}^\top$ for all $k \in [N]$, where $\mathbf{B}[:, k]$ denotes the k-th column of the emission matrix $\mathbf{B}$. Let*

$$\phi_t^{(n)} := u_t \otimes u_{t-1} \otimes \cdots \otimes u_{t-n+1} \in \mathbb{R}^{N^n}, \quad \psi_t^{(n)} := [u_t^\top \ u_{t-1}^\top \ \cdots \ u_{t-n+1}^\top]^\top \in \mathbb{R}^{nN} \tag{1}$$

denote a nonlinear and the corresponding linear feature vector, constructed from n past observations, for $u_t \in \mathbb{R}^N$ one-hot vector encoding o_t. Then, there exists a matrix $\mathcal{G}_{t,n} \in \mathbb{R}^{N \times N^n}$ such that $q_{t+1} = \mathcal{G}_{t,n}\, \phi_t^{(n)}$. Moreover, there exists matrices $\mathbf{W}_{lin} \in \mathbb{R}^{nN \times N^n}$, $\mathbf{W}_{kron} \in \mathbb{R}^{N^n \times nN}$, and a vector $v \in \mathbb{R}^{N^n}$ such that, $\psi_t^{(n)} = \mathbf{W}_{lin} \phi_t^{(n)}$, and $\phi_t^{(n)} = \mathrm{ReLU}(\mathbf{W}_{kron} \psi_t^{(n)} + v)$. where $\mathrm{ReLU}(x) := x^+ = \max(0, x)$, and is applied entry-wise to vectors.

The proof of Lemma 3.1 is deferred to the Appendix C. Under suitable mixing conditions ensuring exponential forgetting of the initial belief, Lemma 3.1 implies that, when $(\boldsymbol{\pi}, \mathbf{A}, \mathbf{B})$ are unknown, the prediction problem reduces to learning a linear operator, with an approximation error ,that decays exponentially with n. Concretely, we can treat the observed sequence $\{o_t\}_{t \geq n}$ as training data and minimize the cross-entropy loss of predicting o_{t+1} from either the nonlinear feature $\phi_t^{(n)}$ or the linear feature $\psi_t^{(n)}$. Since both mappings from feature to prediction are linear, this amounts to fitting a single linear matrix via standard regression or online gradient descent. Our next result shows that Transformers can learn the features $\phi_t^{(n)}$ and $\psi_t^{(n)}$ from raw observation sequences (one-hot), and can emulate GD algorithm to learn a linear map from $\phi_t^{(n)}$ and $\psi_t^{(n)}$ to the probability distribution q_{t+1}.

Theorem 3.2. *Consider the same setup of Lemma 3.1. Furthermore, consider a dataset*

$$\mathcal{D}_m = \{(\mathfrak{h}_t, y_t)\}_{t=n}^{n+m-1}, \qquad \mathfrak{h}_t = (u_t, u_{t-1}, \ldots, u_{t-n+1}), \qquad y_t \in \{e_1, \ldots, e_N\}. \tag{2}$$

Let $\mathbf{W} \in \mathbb{R}^{N \times N^n}$ be the parameter of multinomial logistic regression, $q_{\mathbf{W}}(\cdot \mid \phi) = \mathrm{softmax}(\mathbf{W}\phi)$, trained by full-batch gradient descent on the empirical cross-entropy loss

$$\mathcal{L}_m(\mathbf{W}) := \frac{1}{m} \sum_{t=n}^{n+m-1} \ell(\mathbf{W}\phi_t^{(n)}, y_t), \qquad \ell(z, y) := -y^\top \log \mathrm{softmax}(z),$$

If the number of GD iterations satisfies $T = \mathcal{O}(\log(n))$, then there exists a Transformer of depth $\mathcal{O}(\log(n))$ that emulates T steps of full-batch GD on the CE loss of the logistic-regression model $q_{\mathbf{W}}(\cdot \mid \phi) = \mathrm{softmax}(\mathbf{W}\phi)$.

The proof of Theorem 3.2 is deferred to Appendix C. Our next result shows that a sufficiently deep Transformer can also emulate the spectral learning algorithm in-context, with proof in Appendix D.

Remark 3.3. Consider the setup of Lemma 3.1 and the dataset in equation 2. The spectral learning algorithm estimates low-order moment statistics P_1, P_2, P_3 from the observation prefix, constructs observation operators $B_o := P_2^{-1} P_3(o)$ for each $o \in \mathcal{O}$, and predicts O_{t+1} by recursively updating a belief state via $b_t \propto B_{o_t} b_{t-1}$. By combining several existing results (see Appendix D), there exists a Transformer of depth $\mathcal{O}(\log(n+m))$ that emulates this algorithm in-context.

3.2 Experiment: Which Algorithm does a Small Trained Transformer use?

Training on sequences from a single fixed HMM lets us examine which computation a small Transformer learns when the transition and emission structure is shared across training examples. In this setting, the model can encode the HMM parameters in its weights, reducing the need for in-context parameter estimation. This simpler setting provides a controlled test of the algorithm learned by the architecture (Edelman et al., 2024).

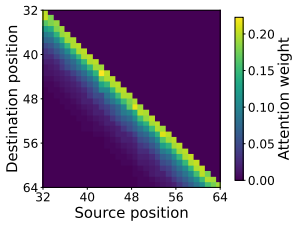


Figure 5: Attention pattern for the trained one layer one head Transformer.

Following the trained-HMM setup of Hu et al. (2024), we vary the number of layers and heads, model dimension, and feedforward dimension to identify a minimal configuration that reliably solves the task (Appendix E.1). For an HMM with $M = N = 3$, $\mathcal{E}(\mathbf{A}) = 0.25$, and $\mathcal{E}(\mathbf{B}) = 0.75$, one layer with one attention head, model dimension 64, and feedforward dimension 256 is sufficient. For comparison, the theoretical Spectral construction of Hu et al. (2024) uses $\mathcal{O}(\log(n+m))$ layers when supplied with the theoretical operators, without in-context learning. Our result establishes sufficiency for this fixed HMM.

Figure 5 visualizes the learned attention pattern. The attention mass is concentrated on a small number of recent tokens rather than spreading over the full history, which is consistent with finite-window prediction—namely, the Linear and Non-linear n-gram algorithms.

Lemma 3.1 establishes a linear representation of the Bayes-optimal prediction in Kronecker features. This motivates a two-part investigation of whether the trained Transformer constructs finite-window features and applies a corresponding optimal linear readout. Let z_t denote either the Linear n-gram feature $\psi_t^{(n)}$ or the Non-linear n-gram feature $\phi_t^{(n)}$, and let d_z be its dimension. First, we test whether the hidden state $x_t \in \mathbb{R}^{d_{\text{model}}}$ linearly encodes z_t by fitting a probe $R \in \mathbb{R}^{d_z \times d_{\text{model}}}$ such that $Rx_t \approx z_t$. Second, we test whether the learned hidden-to-logit map $W \in \mathbb{R}^{N \times d_{\text{model}}}$ is consistent with applying the optimal readout to the decoded features. Specifically, we compare W with $W_{\text{th}}R$, where $W_{\text{th}} \in \mathbb{R}^{N \times d_z}$ is the cross-entropy-optimal linear readout on z_t.

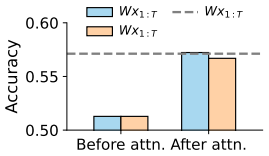


Figure 6: Prediction accuracy using hidden states before and after attention.

Attention makes finite-window features accessible to a linear readout. Figure 6 compares predictions from the learned readout Wx_t, the probe-based readout $W_{\text{th}}Rx_t$, and the feature-based predictor $W_{\text{th}}z_t$. Before attention, the learned and probe-based readouts both achieve low accuracy. After attention, both improve substantially and closely match the feature-based predictor. These results support the view that attention aggregates recent observations into a representation from which a linear readout can recover the predictive information in the finite window.

The learned readout closely matches the optimal finite-window readout. Figure 7 directly compares W and $W_{\text{th}}R$ for $n = 8$, using post-attention hidden states. Their near-zero difference supports a two-step account of the learned computation: attention constructs a finite-window representation, and the output projection applies an approximately optimal linear readout to it. Appendices E.2 and E.3 provide additional ablations.

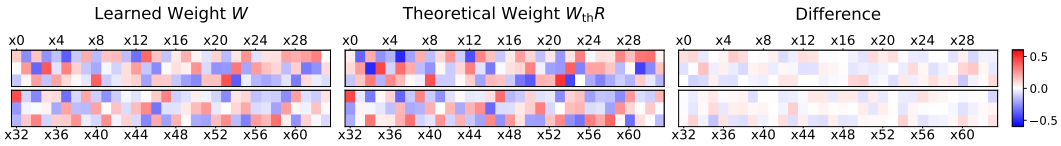


Figure 7: Learned and theoretical readouts for window size $n = 8$. *Left:* the learned hidden-to-logit map $W \in \mathbb{R}^{3\times 64}$. *Middle:* the composed map $W_{\text{th}}R$, where R probes Linear n-gram features from post-attention hidden states and W_{th} is the cross-entropy-optimal readout on those features. *Right:* the difference $W - W_{\text{th}}R$, showing close agreement between the two maps.

4 Stage 3: Internal Algorithmic Representations

Section 3.1 establishes that Transformers can implement each ansatz, and our small-Transformer experiments provide evidence for Linear n-gram computation. We now examine pre-trained LLM activations to test which candidate algorithmic representations are relevant to its in-context prediction.

4.1 Methods

We analyze the residual stream layer by layer, treating each activation as the accumulated computation up to that depth (Elhage et al., 2021). At each layer, we test whether candidate algorithmic representations are linearly decodable and whether interventions along the decoded directions affect prediction. Our probe combines linear probing (Alain & Bengio, 2017; Jiang et al., 2024) with projection onto leading principal components, allowing us to examine representations at different subspace dimensions (Nanda et al., 2023a; Piotrowski et al., 2025).

Principal Activations Probe (PAP). For an observation sequence $\mathbf{o}_{1:T} = (o_1, \ldots, o_T)$, we record the residual-stream activation $x_{\ell,t} \in \mathbb{R}^{d_{\text{model}}}$ at each layer ℓ and position t. In parallel, we compute an *algorithmic representation* $r_t \in \mathbb{R}^{d_r}$ from a candidate algorithm applied to the same prefix $\mathbf{o}_{1:t}$. Section 4.2 defines the candidate representations.

PAP assumes that the train and test sequences be sampled from a single distribution. Since $x_{\ell,t}$ and r_t are both deterministic functions of $\mathbf{o}_{1:t}$, this ensures the activation–representation pairs $(x_{\ell,t}, r_t)$ are identically distributed across the two splits, so that test performance reflects whether r_t is decodable from $x_{\ell,t}$ rather than distribution shift between them. PAP then fits two stages independently per layer on the training split. First, PCA on centered activations yields a top-k orthonormal basis $U_\ell \in \mathbb{R}^{d_{\text{model}}\times k}$ and mean $\bar{x}_\ell$, with $k \ll d_{\text{model}}$. Second, ridge regression maps the k leading PCA coordinates to r_t, yielding $M_\ell \in \mathbb{R}^{d_r \times k}$ and $c_\ell \in \mathbb{R}^{d_r}$. The probe's prediction is $\hat{r}_{\ell,t} = M_\ell U_\ell^\top (x_{\ell,t} - \bar{x}_\ell) + c_\ell$. And the rank k is a hyperparameter we vary in our experiments.

Probe evaluation metrics. Our primary metric is the coefficient of determination, $R^2 = 1 - \frac{\text{MSE}}{\text{Var}(r_t)}$, which measures how much of the variance in the target representation r_t is explained by the probe. Higher R^2 therefore indicates greater linear decodability of the algorithmic representation from the residual stream. The metric is evaluated on the held-out test split.

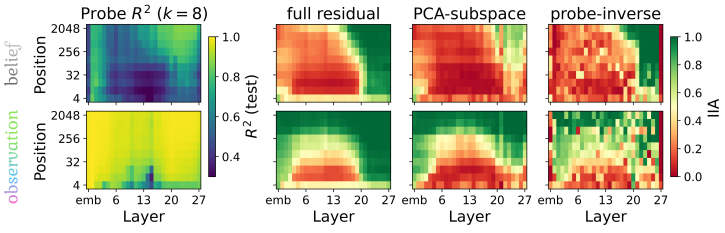


Figure 8: Oracle belief probing and causal patching across layers and positions for the belief-dominated (top) and observation-dominated (bottom) regimes. Three patches of increasing specificity: full residual, PCA-subspace, probe-inverse. $R^2\uparrow$, IIA$\uparrow$.

Causal Intervention with PAP. Decodability does not establish that the model uses a representation for prediction (Belinkov, 2022). We therefore complement probing with activation patching (Zhang & Nanda, 2023). Given source and target prefixes $\mathbf{o}^{\text{src}}_{1:t}$ and $\mathbf{o}^{\text{tgt}}_{1:t}$, we run the model on the source and modify its activation at (ℓ, t) using information from the target. We test whether the patched prediction agrees with the next-token distribu-

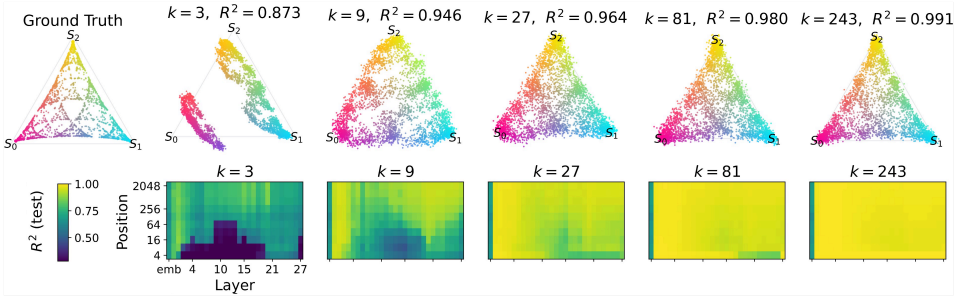


Figure 9: Quality of the belief-geometry mapping for an HMM with $M = 3$, $N = 3$, $\mathcal{E}(\mathbf{A}) = 0.5$ ($\lambda_2 = 0.75$), and $\mathcal{E}(\mathbf{B}) = 0.75$. *(top)* Probe reconstruction of the belief simplex at Layer 20, position 2048, for varying PCA rank k. *(bottom)* Probe quality measured by R^2 on the test set, with each subplot corresponding to a different PCA rank k.

tion implied by the target representation. We use three interventions of increasing specificity. The **full residual patch** replaces the entire activation, $x^{\text{src}}_{\ell,t} \leftarrow x^{\text{tgt}}_{\ell,t}$, and serves as the least restrictive reference intervention. The **PCA-subspace patch** replaces only the top-k PCA component, $x^{\text{src}}_{\ell,t} \leftarrow U_\ell U_\ell^\top (x^{\text{tgt}}_{\ell,t} - \bar{x}_\ell) + (\mathbf{I} - U_\ell U_\ell^\top)(x^{\text{src}}_{\ell,t} - \bar{x}_\ell) + \bar{x}_\ell$, testing whether the low-dimensional subspace identified by PAP is causally relevant. Finally, the **probe-inverse patch** applies the minimum-norm perturbation $\Delta x = U_\ell M_\ell^+ \left(r^{\text{tgt}}_t - \hat{r}^{\text{src}}_{\ell,t}\right)$, where M_ℓ^+ is the pseudoinverse. This restricts the intervention to directions read out by the probe, providing the most targeted test of causal relevance.

Intervention metrics. We evaluate interventions against the target next-token distribution p^{tgt}_{t+1} implied by r^{tgt}_t. *Interchange intervention accuracy* (IIA) (Geiger et al., 2021; 2024) is the fraction of source–target pairs for which $\arg\max y^{\text{patched}}_{t+1} = \arg\max p^{\text{tgt}}_{t+1}$, measuring whether the intervention shifts the model's discrete prediction to match the target.

The components above: layer-wise linear probing (Alain & Bengio, 2017), PCA projection, and activation patching (Zhang & Nanda, 2023; Geiger et al., 2021) are individually standard. Our contribution lies in probing on candidate algorithmic representations drawn from the predictor families of Section 3.1. This lets us ask which algorithm's intermediate state best explains and causally drives the model's predictions.

4.2 Setup: Algorithmic Representations in LLMs on HMM Tasks

Candidate representations. We focus on the Linear n-gram, Non-linear n-gram, and Spectral ansatzes studied in Sections 2.3 and 3.1. We group the two n-gram classes under *Soft n-gram* because both predict through a learned map applied to finite-window features.

The two representation types capture complementary aspects of the algorithm's behavior over the observation sequence. The *belief* r^{state}_t tracks the current estimate of the hidden state and updates with each new observation. The *operator* r^{op}_t encodes the estimated parameters of the prediction rule; it is updated incrementally and converges toward the ground-truth values as the sequence grows. Thus, the belief reflects both the accumulated observations and the current quality of the operator estimate, becoming more accurate as the operator converges.

The Oracle provides a reference representation computed using the true HMM parameters $\boldsymbol{\lambda}$; we do not probe an Oracle operator. For Soft n-gram, $z_t \in \{\psi^{(n)}_t, \phi^{(n)}_t\}$ is the corresponding finite-window feature, and $W_t \in \mathbb{R}^{N \times \dim(z_t)}$ is the weight learned from the prefix. For Spectral methods (Hsu et al., 2012), $b_t \in \mathbb{R}^M$ is updated recursively through observation operators, $b_t \propto B_{o_t} b_{t-1}$. The operator representation $B_t := \{B_x\}_{x \in \mathcal{O}}$ collects the operators estimated from the prefix's empirical bigram and trigram statistics (Appendix B.2).

Table 2: The representations to be probed for each ansatz algorithm class.

Algorithm	Belief r^{state}_t	Operator r^{op}_t
Oracle	$\mathbb{P}(h_t \mid \mathbf{o}_{1:t}, \boldsymbol{\lambda})$	–
Soft n-gram	$W_t z_t$	W_t
Spectral	b_t	B_t

Experimental Setup. We use Qwen-3-1.7B with 28 layers and $d_{\text{model}} = 2048$, and reuse the 75 HMM configurations from Section 2.3. For each HMM $\boldsymbol{\lambda} = (\boldsymbol{\pi}, \mathbf{A}, \mathbf{B})$, we sample 4,096 sequences of length 2,049 and randomly split them by sequence into 80% training and 20% test

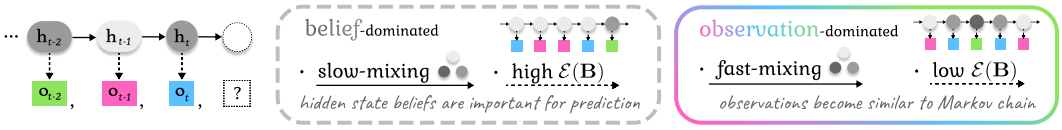


Figure 10: Two HMM regimes: *belief*-dominated, where latent-state beliefs drive prediction, and *observation*-dominated, where observations are approximately Markovian.

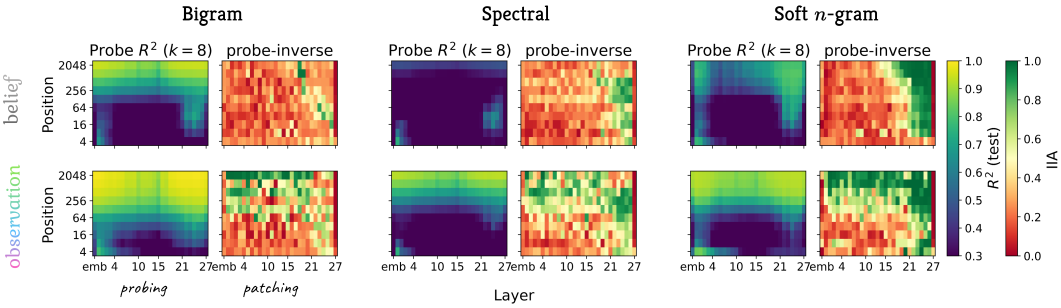


Figure 11: Decodability and intervention effects of algorithmic predictive states in two representative HMM regimes (one per row). Soft n-gram (right panel), instantiated by Linear n-gram with $n = 4$, achieves the highest IIA among the tested candidates in both regimes. Higher R^2 indicates better probe reconstruction; higher IIA indicates greater agreement with the target prediction after intervention.

data. At positions $t \in \{4, 8, \ldots, 2048\}$, we compute the candidate representations and record the residual-stream activations. We fit PCA and ridge regression on the training split and evaluate reconstruction on the test split, varying the retained PCA rank. For interventions, we sample 1,024 source–target pairs whose algorithmic states at position t have different maximizing coordinates, $\arg\max r_t^{\text{src}} \neq \arg\max r_t^{\text{tgt}}$.

4.3 Results

Oracle beliefs are linearly decodable from the residual stream. Figure 9 reports Oracle-belief probes for a three-state HMM. At PCA rank $k = 9$, reconstruction R^2 is high in early (0–3) and late (20–26) layers. At $k = 243$, the belief is nearly perfectly decodable. Following Shai et al. (2024), we also visualize the reconstructed belief simplex. The probe recovers its fractal-like geometry in the pre-trained LLM, although visible reconstruction noise remains even at high R^2.

We next examine interventions in two representative HMM regimes (Figure 10). In the *belief-dominated* regime, accurate prediction requires integrating observations over longer contexts to track uncertainty over hidden states. In the *observation-dominated* regime, recent observations provide more direct predictive information, and is closer to Markovian. Figure 8 reports Oracle-belief probing and intervention results across layers and positions; full results appear in Appendix G.

Decodability and intervention effects differ across regimes. We compare the predictive states in Table 2 and additionally include Bigram, which predicts using first-order transition statistics (Figure 11). At PCA rank $k = 8$, the Bigram representation is highly decodable in both regimes, but its interventions are effective only in the observation-dominated regime. This contrast illustrates why decodability alone is insufficient to establish a representation's role in prediction. The intervention results also align with the LLM's empirical prediction performance across regimes (Appendix G.4).

Soft n-gram interventions have the strongest causal effects in both representative regimes. Among the candidates tested, interventions on the Soft n-gram representation are the most causally effective in both regimes (Figure 11). These results support the causal relevance of directions encoding finite-window predictions, including in the belief-dominated regime where accurate prediction benefits from longer-context integration. We also probe the Linear n-gram operator W_t in the belief-dominated regime (Fig-

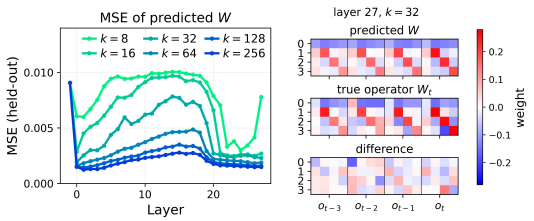


Figure 12: Probing Linear n-gram operator W_t.

ure 12). Reconstruction follows a U-shaped pattern across layers, with higher decodability in early and late layers. The right panel shows reconstructed operator weights at a representative layer.

These mechanistic findings favor Soft n-gram as an account of the model's in-context predictions. They complement the behavioral similarities (Section 2.3) and Transformer constructions (Section 3), providing converging evidence for finite-window computation in the LLM.

5 Related Works

Algorithmic Identification in LLMs. Recent theoretical results establish that Transformers can implement a broad range of ICL algorithms Bai et al. (2023); Garg et al. (2022); Akram & Vikalo (2026). Most of this work infers algorithms from input–output behavior or explicit constructions rather than internal activations. Exceptions include probing for implicit model updates across layers Akyürek et al. (2023), fully reverse-engineering learned algorithms through weights, activations, and ablations Nanda et al. (2023a), and distinguishing between competing internal algorithms for state-tracking tasks Li et al. (2025). Related theory also connects in-context prediction to mirror descent in latent mixture models D'Angelo & Flammarion (2026).

Mechanistic Interpretability and Probing. Linear probing has long been used to characterize intermediate neural representations Alain & Bengio (2017); Adi et al. (2017); Hupkes & Zuidema (2018); Conneau et al. (2018). More recent work applies this framework to Transformers and synthetic sequence tasks, revealing linearly decodable world-model structure and other task-relevant variables Li et al. (2023); Nanda et al. (2023b); Karvonen (2024). Activation patching further complements probing by testing whether decoded representations are causally used, including in state-tracking and structured world-model settings Vafa et al. (2025); Liu et al. (2026).

Transformers on HMMs. HMMs provide a controlled setting for studying the capabilities and internal computations of Transformers. Hu et al. (2024) identify learning difficulties relative to recurrent networks and establish logarithmic-depth constructions for approximating HMMs. Complementary work demonstrates in-context HMM prediction in pre-trained LLMs (Dai et al., 2026) and studies feature learning and multi-task generalization in Transformers trained on HMMs (Hao et al., 2025). Mechanistic studies show that hidden-state beliefs are linearly represented in the residual stream (Shai et al., 2024) and explain how attention implements constrained Bayesian belief updates (Piotrowski et al., 2025). Contemporaneous work by Balcells et al. (2026) finds linearly decodable belief-state geometry in pre-trained LLMs and tests its functional relevance through activation interventions. Our work complements these studies by comparing multiple candidate algorithms through their predictive behavior, Transformer implementations, and internal representations.

6 Conclusions and Limitations

We study in-context HMM prediction in pre-trained LLMs through empirical, theoretical, and mechanistic analysis. Empirically, no single classical predictor explains LLM behavior across regimes, but Linear n-gram, Non-linear n-gram, and Spectral predictors capture its main patterns. Theoretically, Transformers can implement all three through a shared nonlinear-feature construction, and a small Transformer trained on a fixed HMM closely recovers a finite-window Linear n-gram predictor. Mechanistically, PAP distinguishes decodability from causal relevance: Bigram representations are decodable in both representative regimes but have effective interventions only in the observation-dominated regime, whereas Soft n-gram interventions are the most effective in both.

Together, these findings favor Soft n-gram as an account of in-context prediction. They provide converging evidence for finite-window computation even in regimes where prediction benefits from longer-range integration. However, they do not uniquely identify the implemented algorithm, establish gradient descent as its learning rule, or rule out alternative inference procedures.

Our study is limited to synthetic HMMs with small finite alphabets. Linear probes restricted to leading principal components may miss information encoded nonlinearly or outside the retained subspace, and interventions along decoded directions do not establish a complete algorithmic implementation. While our work focuses on synthetic HMMs to produce controllable and grounded insights, our work builds the foundation for studying the latent structure and resolving circuits underlying representations on natural-language tasks in the future.

AI USE STATEMENT

In this work, we used generative AI tools to generate synthetic data sets, assist in the writing of proofs, design or provide feedback on research experiments, implementing methods, support qualitative and thematic data analysis, and clean and reformat dataset. We have not used generative AI tools for helping to develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, proposing or refining hypotheses, designing or providing feedback on research methodology, interpreting results and assisting with translation are not applicable to this work. Additionally, we used generative AI tools to create or modify scientific figures or images, sourcing/searching for information, and edit a research paper to improve readability. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We include the synthetic dataset generation details in Section 2.3 and Appendix B, the small trained Transformer experiment details in Section 3.2 and Appendix E, the probing construction and experiment details in Section 4 and Appendix G. The proof of theoretical results are in Appendix C and D. We will include the code for synthetic dataset generation, pre-trained LLMs evaluation, small Transformer training, and probing in the supplementary materials.

REFERENCES

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=BJh6Ztuxl.

Usman Akram and Haris Vikalo. Transformers as implicit state estimators: In-context learning in dynamical systems. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=hIMK5MvGkP.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=0g0X4H8yN4I.

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes, 2017. URL https://openreview.net/forum?id=ryF7rTqgl.

Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=liMSqUuVg9.

Daniel Balcells, Andrew Jun Lee, Chirag Rastogi, Paul M Riechers, Adam Shai, and Xavier Poncini. Large language models develop belief state geometry in-context. *arXiv preprint arXiv:2609.17376*, 2026.

Leonard E Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, 41(1):164–171, 1970.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2126–2136, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1198. URL https://aclanthology.org/P18-1198/.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006. ISBN 0471241954.

Yijia Dai, Zhaolin Gao, Yahya Sattar, Sarah Dean, and Jennifer J. Sun. Pre-trained large language models learn to predict hidden markov models in-context. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=btBqWTbf6q.

Francesco D'Angelo and Nicolas Flammarion. Transformers learn latent mixture models in-context via mirror descent. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=SHidElLSVt.

Ezra Edelman, Nikolaos Tsilivis, Benjamin L. Edelman, eran malach, and Surbhi Goel. The evolution of statistical induction heads: In-context learning markov chains. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=qaRT6QTIqJ.

Chanakya Ekbote, Marco Bondaschi, Nived Rajaraman, Jason D Lee, Michael Gastpar, Ashok Vardhan Makkuva, and Paul Pu Liang. What one cannot, two can: Two-layer transformers provably represent induction heads on any-order markov chains. *arXiv preprint arXiv:2508.07208*, 2025.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

Y. Ephraim and N. Merhav. Hidden markov processes. *IEEE Transactions on Information Theory*, 48(6):1518–1569, 2002. doi: 10.1109/TIT.2002.1003838.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=flNZJ2eOet.

Atticus Geiger, Hanson Lu, Thomas F Icard, and Christopher Potts. Causal abstractions of neural networks. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=RmuXDtjDhG.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah Goodman. Finding alignments between interpretable causal variables and distributed neural representations. In Francesco Locatello and Vanessa Didelez (eds.), *Proceedings of the Third Conference on Causal Learning and Reasoning*, volume 236 of *Proceedings of Machine Learning Research*, pp. 160–187. PMLR, 01–03 Apr 2024. URL https://proceedings.mlr.press/v236/geiger24a.html.

Richard Glennie, Timo Adam, Vianey Leos-Barajas, Théo Michelot, Theoni Photopoulou, and Brett T McClintock. Hidden markov models: Pitfalls and opportunities in ecology. *Methods in Ecology and Evolution*, 14(1):43–56, 2023.

Yifan Hao, Chenlu Ye, Chi Han, and Tong Zhang. Transformers as multi-task learners: Decoupling features in hidden markov models. *arXiv preprint arXiv:2506.01919*, 2025.

Daniel Hsu, Sham M Kakade, and Tong Zhang. A spectral algorithm for learning hidden markov models. *Journal of Computer and System Sciences*, 78(5):1460–1480, 2012.

Jiachen Hu, Qinghua Liu, and Chi Jin. On limitation of transformer for learning hmms. *arXiv preprint arXiv:2406.04089*, 2024.

Dieuwke Hupkes and Willem Zuidema. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure (extended abstract). In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pp. 5617–5621. International Joint Conferences on Artificial Intelligence Organization, 7 2018. doi: 10.24963/ijcai.2018/796. URL https://doi.org/10.24963/ijcai.2018/796.

Yibo Jiang, Goutham Rajendran, Pradeep Kumar Ravikumar, Bryon Aragam, and Victor Veitch. On the origins of linear representations in large language models. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=otuTw4Mghk.

Adam Karvonen. Emergent world models and latent variable estimation in chess-playing language models. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=PPTrmvEnpW.

Belinda Z. Li, Zifan Carl Guo, and Jacob Andreas. (how) do language models track state? In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=8SXosAVIFH.

Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=DeG07_TcZvT.

Ziming Liu, Sophia Sanborn, Surya Ganguli, and Andreas Tolias. From kepler to newton: Inductive biases guide learned world models in transformers, 2026. URL https://arxiv.org/abs/2602.06923.

Xiaoyuan Ma and Jordan Rodu. Bridging the usability gap: Theoretical and methodological advances for spectral learning of hidden markov models. *arXiv preprint arXiv:2302.07437*, 2023.

Ashok Vardhan Makkuva, Marco Bondaschi, Adway Girish, Alliot Nagle, Martin Jaggi, Hyeji Kim, and Michael Gastpar. Attention with markov: A curious case of single-layer transformers. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=SqZ0KY4qBD.

Brett T McClintock, Roland Langrock, Olivier Gimenez, Emmanuelle Cam, David L Borchers, Richard Glennie, and Toby A Patterson. Uncovering ecological state dynamics with hidden markov models. *Ecology letters*, 23(12):1878–1903, 2020.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=9XFSbDPmdW.

Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world models of self-supervised sequence models. In Yonatan Belinkov, Sophie Hao, Jaap Jumelet, Najoung Kim, Arya McCarthy, and Hosein Mohebbi (eds.), *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 16–30, Singapore, December 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.blackboxnlp-1.2. URL https://aclanthology.org/2023.blackboxnlp-1.2/.

Mateusz Piotrowski, Paul M. Riechers, Daniel Filan, and Adam Shai. Constrained belief updates explain geometric structures in transformer representations. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=f6Hl60FBFU.

Nived Rajaraman, Marco Bondaschi, Ashok Vardhan Makkuva, Kannan Ramchandran, and Michael Gastpar. Transformers on markov data: Constant depth suffices. In *ICML 2024 Workshop on Mechanistic Interpretability*, 2024. URL https://openreview.net/forum?id=jnCM5EHd2H.

Yahya Sattar, Yassir Jedra, Maryam Fazel, and Sarah Dean. Finite sample identification of partially observed bilinear dynamical systems. *arXiv preprint arXiv:2501.07652*, 2025.

Adam Shai, Paul M. Riechers, Lucas Teixeira, Alexander Gietelink Oldenziel, and Sarah Marzen. Transformers represent belief state geometry in their residual stream. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=YIB7REL8UC.

Keyon Vafa, Peter G. Chang, Ashesh Rambachan, and Sendhil Mullainathan. What has a foundation model found? inductive bias reveals world models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=i9npQatSev.

Aditya Varre, Gizem Yüce, and Nicolas Flammarion. Learning in-context n-grams with transformers: Sub-n-grams are near-stationary points. *arXiv preprint arXiv:2508.12837*, 2025.

Itay Yona, Ilia Shumailov, Jamie Hayes, Federico Barbero, and Yossi Gandelsman. Interpreting the repeated token phenomenon in large language models. *arXiv preprint arXiv:2503.08908*, 2025.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. *arXiv preprint arXiv:2309.16042*, 2023.

Walter Zucchini and Peter Guttorp. A hidden markov model for space-time precipitation. *Water Resources Research*, 27(8):1917–1923, 1991.

Appendices

Table of Contents

- Appendix A: Additional HMM Background
- Appendix B: Benchmark Algorithmic Details
- Appendix C: n-gram Prediction Constructions
- Appendix D: Spectral Learning for Prediction
- Appendix E: Details of the Small Trained Transformer
- Appendix F: LLMs Empirical Performance
- Appendix G: Principal Activation Probe (PAP) and Results

A Additional HMM Background

In this section, we present the HMM setting in detail, along with the conditions under which the underlying Markov chain converges to a unique stationary distribution. Recall that an HMM is characterized by the Markov chain's *initial state distribution* and *state transitions*, together with the *emission probabilities* of observations given hidden states. We focus on the finite-alphabet setting, taking states in $\mathcal{X} = \{1, 2, \ldots, M\}$ and observations in $\mathcal{O} = \{1, 2, \ldots, N\}$ without loss of generality. The model parameters comprise an initial distribution $\boldsymbol{\pi} \in \mathbb{R}^M$, with π_j the probability of starting in state j; a transition matrix $\mathbf{A} \in \mathbb{R}^{M \times M}$, with a_{ij} the probability of moving from state i to state j; and an emission matrix $\mathbf{B} \in \mathbb{R}^{M \times N}$, with b_{jk} the probability of emitting observation k from state j. The triple $\lambda = (\boldsymbol{\pi}, \mathbf{A}, \mathbf{B})$ fully parameterizes a finite-alphabet HMM.

Let $\{X_1, X_2, \ldots\}$ be a discrete-time Markov chain on $\mathcal{X}$ with transition matrix $\mathbf{A}$, and let $p_{ij}^{(n)} = \mathbb{P}(X_{t+n} = j \mid X_t = i)$ denote the n-step transition probability from i to j. State j is *accessible* from state i if $p_{ij}^{(n)} > 0$ for some $n \geq 1$, and a subset $\mathcal{C} \subseteq \mathcal{X}$ is *irreducible* if every pair of states in $\mathcal{C}$ is mutually accessible. The *period* of state i is $c(i) = \gcd\{n \geq 1 : p_{ii}^{(n)} > 0\}$, the greatest common divisor of its possible return times; state i is *aperiodic* when $c(i) = 1$. A Markov chain is *geometrically ergodic* if it is irreducible and aperiodic, in which case it converges to a unique *stationary distribution* $\boldsymbol{\mu} \in \mathbb{R}^M$ satisfying $\boldsymbol{\mu} = \boldsymbol{\mu}\mathbf{A}$. Its *mixing rate* is the smallest $\rho \in [0, 1)$ for which $|p_{ij}^{(n)} - \mu_j| \leq C\rho^n$ holds with some constant $C \geq 0$, uniformly over $i, j \in \mathcal{X}$ and $n \geq 1$; for a finite-alphabet HMM, ρ equals λ_2, the second-largest eigenvalue of $\mathbf{A}$ in modulus. To avoid dependence on the initial state, the bulk of our HMMs use ergodic transitions, though we also include a handful of non-ergodic cases in our experiments.

The *entropy* of a discrete random variable X is $H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x)$. A fundamental property is that conditioning reduces uncertainty: $H(X \mid Y) \leq H(X)$ for any random variables X and Y, with equality if and only if they are independent (Cover & Thomas, 2006). The chain rule then expresses the joint entropy of a stochastic process as $H(X_1, X_2, \ldots, X_n) = \sum_{i=1}^{n} H(X_i \mid X_{i-1}, \ldots, X_1)$. For a Markov chain with stationary distribution $\boldsymbol{\mu}$, the *entropy rate* simplifies to $H(\mathcal{X}) = \lim_{n \to \infty} \frac{1}{n} H(X_1, \ldots, X_n) = -\sum_{i,j} \mu_i a_{ij} \log a_{ij}$, depending only on the transition matrix $\mathbf{A}$. By analogy, we define the entropy of the emission matrix as $-\sum_{j,k} \mu_j b_{jk} \log b_{jk}$, which captures the average uncertainty in observations given the underlying state. The entropy rate of the observation process in an HMM admits no closed form, but it is sandwiched between $H(O_n \mid O_{n-1}, X_{n-1}, \ldots, O_1, X_1) \leq H(\mathcal{O}) \leq H(O_n \mid O_{n-1}, \ldots, O_1)$. Since $\mathbf{A}$ governs the transitions $X_t \to X_{t+1}$ and $\mathbf{B}$ governs the emission of O_t from X_t, the entropies of $\mathbf{A}$ and $\mathbf{B}$ jointly control a lower bound on the entropy of the sampled HMM sequence.

B Benchmark Algorithm Details

This appendix details the implementation of each benchmark predictor used in Section 2.2 and Section 2.3. Throughout, $u_t = e_{o_t} \in \mathbb{R}^N$ denotes the one-hot encoding of observation o_t, and $q_{t+1} \in \Delta^{N-1}$ denotes the predicted next-observation distribution.

B.1 Oracle: Forward Algorithm

Given the true HMM parameters $\boldsymbol{\lambda} = (\boldsymbol{\pi}, \mathbf{A}, \mathbf{B})$, the Bayes-optimal predictor is computed by the standard forward recursion. The belief state $b_t \in \Delta^{M-1}$ is updated as

$$b_t = \frac{\operatorname{diag}(\mathbf{B}u_t)\,\mathbf{A}^\top b_{t-1}}{\|\operatorname{diag}(\mathbf{B}u_t)\,\mathbf{A}^\top b_{t-1}\|_{\ell_1}}, \qquad b_0 = \boldsymbol{\pi},$$

and the next-observation distribution is read out as $q_{t+1} = \mathbf{B}^\top \mathbf{A}^\top b_t$.

B.2 Spectral Methods

Spectral methods estimate empirical low-order moments $\hat{P}_1 \in \mathbb{R}^N$, $\hat{P}_2 \in \mathbb{R}^{N\times N}$, and $\hat{P}_3(o) \in \mathbb{R}^{N\times N}$ for each $o \in \mathcal{O}$ from observation triples in the prefix $\mathbf{o}_{1:t}$, then form an observation-operator predictor that bypasses iterative parameter estimation. The observation operator for symbol o is $\hat{B}_o := \hat{P}_2^{+}\hat{P}_3(o)$, so the belief state $b_t \in \mathbb{R}^M$ evolves recursively as $b_t \propto \hat{B}_{o_t} b_{t-1}$, with $b_0 \propto \hat{P}_2^{+}\hat{P}_1$. The resulting predictor takes the form

$$\mathbb{P}(O_{t+1} = o \mid \mathbf{o}_{1:t}) \;\propto\; \hat{P}_1^\top\, \hat{B}_{o_t} \cdots \hat{B}_{o_1}\, \hat{P}_2^{+}\hat{P}_1,$$

where the operator $\hat{B}_t := \{\hat{B}_o\}_{o\in\mathcal{O}}$ converges to its ground-truth value as more observations are used to estimate the moments.

`Spectral`. Standard implementation following Hsu et al. (2012), using the rank-M truncated pseudoinverse of $\hat{P}_2$ via SVD as above.

`Spectral-Norm`. Replaces $\hat{P}_2^{+}$ with $\hat{P}_2^\top$, defining $\hat{B}_o = \hat{P}_2^\top \hat{P}_3(o)$:

$$\mathbb{P}(O_{t+1} = o \mid \mathbf{o}_{1:t}) \;\propto\; \hat{P}_1^\top\, \hat{P}_2^\top \hat{P}_3(o_t) \cdots \hat{P}_2^\top \hat{P}_3(o_1)\, \hat{P}_2^\top \hat{P}_1.$$

This avoids ill-conditioning from small singular values of $\hat{P}_2$ at the cost of biased moment matching.

B.3 Linear n-gram Methods

Let $\psi_t^{(n)} := [u_t^\top, u_{t-1}^\top, \ldots, u_{t-n+1}^\top]^\top \in \mathbb{R}^{nN}$ be the stacked one-hot feature of the n most recent observations. We fit a linear predictor $W \in \mathbb{R}^{N\times nN}$ in three variants, corresponding to `Gradient CE`, `Gradient MSE`, and `Ridge MSE` in Table 1.

`Gradient CE`. Online gradient descent on the cross-entropy loss

$$\ell_{\mathrm{CE}}(W; t) = -u_{t+1}^\top \log \operatorname{softmax}(W\psi_t^{(n)}),$$

with predictor $q_{t+1} = \operatorname{softmax}(W_t\psi_t^{(n)})$ and update $W_{t+1} = W_t - \eta\,(\operatorname{softmax}(W_t\psi_t^{(n)}) - u_{t+1})\,(\psi_t^{(n)})^\top$.

`Gradient MSE`. Online gradient descent on the squared error

$$\ell_{\mathrm{MSE}}(W; t) = \tfrac{1}{2}\|W\psi_t^{(n)} - u_{t+1}\|_2^2,$$

with $q_{t+1} = W_t\psi_t^{(n)}$ projected onto the simplex.

`Ridge MSE`. At each evaluation position, we solve the closed-form ridge regression

$$\hat{W}_t = \arg\min_W \sum_{s=n}^{t-1} \|W\psi_s^{(n)} - u_{s+1}\|_2^2 \; + \; \lambda\|W\|_F^2,$$

and predict $q_{t+1} = \hat{W}_t\psi_t^{(n)}$ projected onto the simplex. Regularization λ is selected on a held-out validation split.

For the `Soft` n`-gram` entry of Table 2 (Section 4), we use `Ridge MSE` weights at position t: the operator representation is W_t and the belief representation is $W_t\psi_t^{(n)}$.

B.4 Non-linear n-gram Methods

Let $\phi_t^{(n)} := u_t \otimes u_{t-1} \otimes \cdots \otimes u_{t-n+1} \in \mathbb{R}^{N^n}$ be the Kronecker feature. By Lemma 3.1, $q_{t+1} = \mathcal{G}_{t,n}\,\phi_t^{(n)}$ for some $\mathcal{G}_{t,n} \in \mathbb{R}^{N\times N^n}$.

`Kron`. Ridge regression of the next observation on $\phi_t^{(n)}$. Because $\phi_t^{(n)}$ is an indicator for a single n-gram (exactly one entry equals one, the rest zero), ridge regression reduces to bucketing the prefix by its n-gram suffix and tabulating smoothed empirical conditional distributions.

`Kernel`. Kernel ridge regression on the n-observation window, capturing the same nonlinear interactions as $\phi_t^{(n)}$ without materializing the $\mathcal{O}(N^n)$ feature vector, which permits larger n.

B.5 Baum–Welch

We run the standard Baum–Welch EM algorithm (Baum et al., 1970) with M assumed known. Rows of $\mathbf{A}$ and $\mathbf{B}$ are initialized by drawing uniformly from the simplex (i.e., symmetric Dirichlet with concentration 1); we run to convergence in negative log-likelihood (tolerance 10^{-6}, max 200 iterations). At each evaluation position, the converged parameters are plugged into the forward recursion of Appendix B.1.

C n-GRAM PREDICTION

C.1 NONLINEAR FEATURES FOR n-GRAM PREDICTION

Recall that, an HMM is specified by the triple $\boldsymbol{\lambda} = (\boldsymbol{\pi}, \mathbf{A}, \mathbf{B})$, where $\boldsymbol{\pi} \in \mathbb{R}^M$ is the initial state distribution (π_j the probability of starting in state j), $\mathbf{A} \in \mathbb{R}^{M \times M}$ is the transition matrix (a_{ij} the probability of moving from state i to j), and $\mathbf{B} \in \mathbb{R}^{M \times N}$ is the emission matrix (b_{jk} the probability of emitting k from state j). Let u_t denote the one-hot vector in $\mathbb{R}^N$ corresponding to the observation o_t, that is, $u_t = e_{o_t}$, where $e_1, \ldots, e_N$ denote the standard basis vectors of $\mathbb{R}^N$. Let $q_t \in \mathbb{R}^N$ denote the probability distribution of next observation given the history of observations, that is, $q_{t+1}[i] := \mathbb{P}\left(O_{t+1} = i \;\middle|\; O_{1:t} = o_1, o_2, \ldots, o_t\right)$. Another important quantity is the belief state $b_t \in \mathbb{R}^M$ which is defined as the posterior probability of the HMM being at each state given the history of observations, that is, $b_t[i] := \mathbb{P}\left(X_{t+1} = i \;\middle|\; O_{1:t} = o_1, o_2, \ldots, o_t\right)$. Using Bayes' rule, we have

$$\begin{aligned} b_t &= \frac{\operatorname{diag}(\mathbf{B}u_t)\mathbf{A}^\top b_{t-1}}{\|\operatorname{diag}(\mathbf{B}u_t)\mathbf{A}^\top b_{t-1}\|_{\ell_1}}, \\ q_{t+1} &= \mathbf{B}^\top \mathbf{A}^\top b_t \end{aligned} \tag{3}$$

Opening up the recursion till $t-n$, and defining the scaling $\alpha_{t,n} := \prod_{i=1}^{n} \left\|\operatorname{diag}(\mathbf{B}u_{t-i+1})\mathbf{A}^\top b_{t-i}\right\|_{\ell_1}$, we have

$$q_{t+1} = \mathbf{B}^\top \mathbf{A}^\top \cdot \operatorname{diag}(\mathbf{B}u_t)\mathbf{A}^\top \cdot \operatorname{diag}(\mathbf{B}u_{t-1})\mathbf{A}^\top \cdots \operatorname{diag}(\mathbf{B}u_{t-n+1})\mathbf{A}^\top b_{t-n}/\alpha_{t,n} \tag{4}$$

Let $\mathbf{B}[:, k]$ denote the k-th column of the emission matrix $\mathbf{B}$, and let $\mathbf{A}_k := \operatorname{diag}\left(\mathbf{B}[:, k]\right)\mathbf{A}^\top$. Then equation 3, and equation 4 can be alternately expressed as as a bilinear dynamical equation Sattar et al. (2025) as follows,

$$\begin{aligned} b_t &= \operatorname{diag}\left(\sum_{k=1}^{N} u_t[k]\mathbf{B}[:, k]\right)\mathbf{A}^\top b_{t-1}/\alpha_{t,1} = \sum_{k=1}^{N} u_t[k] \operatorname{diag}\left(\mathbf{B}[:, k]\right)\mathbf{A}^\top b_{t-1}/\alpha_{t,1} \\ &= \sum_{k=1}^{N} u_t[k]\mathbf{A}_k b_{t-1}/\alpha_{t,1} \\ q_{t+1} &= \mathbf{B}^\top \mathbf{A}^\top b_t \end{aligned} \tag{5}$$

Again, opening up the recursion equation 5 till $t - n$, we have

$$q_{t+1} = \mathbf{B}^\top \mathbf{A}^\top \left(\sum_{k=1}^{N} u_t[k]\mathbf{A}_k\right)\left(\sum_{k=1}^{N} u_{t-1}[k]\mathbf{A}_k\right) \cdots \left(\sum_{k=1}^{N} u_{t-n+1}[k]\mathbf{A}_k\right) b_{t-n}/\alpha_{t,n} \tag{6}$$

Using equation 6, we can write the next observation prediction as a function of a nonlinear feature vector as follows.

Lemma C.1 (n-gram prediction)**.** *Consider an HMM specified by $\boldsymbol{\lambda} = (\boldsymbol{\pi}, \mathbf{A}, \mathbf{B})$. Let $\mathbf{A}_k := \operatorname{diag}\left(\mathbf{B}[:, k]\right)\mathbf{A}^\top$ for all $k \in [N]$, where $\mathbf{B}[:, k]$ denotes the k-th column of the emission matrix $\mathbf{B}$. Let*

$$\phi_t^{(n)} := u_t \otimes u_{t-1} \otimes \cdots \otimes u_{t-n+1} \in \mathbb{R}^{N^n}, \tag{7}$$

denote a nonlinear feature of n past observations. Let b_t denotes the belief state at time t, and $\alpha_{t,0} > 0$ be a normalizing scalar. Then, there exists a matrix $\mathcal{G}_{t,n} \in \mathbb{R}^{N \times N^n}$ such that

$$q_{t+1} = \mathcal{G}_{t,n}\, \phi_t^{(n)}, \tag{8}$$

where the column of $\mathcal{G}_{t,n}$ indexed by $(k_0, \ldots, k_{n-1}) \in [N]^n$ is given by

$$\mathcal{G}_{t,n}[:, (k_0, \ldots, k_{L-1})] = \mathbf{B}^\top \mathbf{A}^\top \mathbf{A}_{k_0}\mathbf{A}_{k_1} \cdots \mathbf{A}_{k_{n-1}}\, b_{t-n}/\alpha_{t,n}. \tag{9}$$

Proof. From equation equation 6, we have

$$q_{t+1} = \mathbf{B}^\top \mathbf{A}^\top \left(\sum_{k=1}^{N} u_t[k]\mathbf{A}_k\right)\left(\sum_{k=1}^{N} u_{t-1}[k]\mathbf{A}_k\right) \cdots \left(\sum_{k=1}^{N} u_{t-n+1}[k]\mathbf{A}_k\right) b_{t-n}/\alpha_{t,n}.$$

Distributing the product over the sums yields

$$q_{t+1} = \sum_{k_0=1}^{N} \sum_{k_1=1}^{N} \cdots \sum_{k_{n-1}=1}^{N} \left(\mathbf{B}^\top \mathbf{A}^\top \mathbf{A}_{k_0} \mathbf{A}_{k_1} \cdots \mathbf{A}_{k_{n-1}} \frac{b_{t-n}}{\alpha_{t,n}} \right) \left(u_t[k_0] u_{t-1}[k_1] \cdots u_{t-n+1}[k_{n-1}] \right).$$

Next, note that the coordinate of $\phi_t^{(n)} = u_t \otimes u_{t-1} \otimes \cdots \otimes u_{t-n+1}$ indexed by $(k_0, \ldots, k_{n-1})$ is precisely $\phi_t^{(n)}[(k_0, \ldots, k_{n-1})] = u_t[k_0] u_{t-1}[k_1] \cdots u_{t-n+1}[k_{n-1}]$. Therefore, defining $\mathcal{G}_{t,n}$ columnwise by

$$\mathcal{G}_{t,n}[:, (k_0, \ldots, k_{n-1})] = \mathbf{B}^\top \mathbf{A}^\top \mathbf{A}_{k_0} \mathbf{A}_{k_1} \cdots \mathbf{A}_{k_{n-1}} \, b_{t-n} / \alpha_{t,n} \tag{10}$$

we obtain $q_{t+1} = \mathcal{G}_{t,n} \phi_t^{(n)}$. This completes the proof. □

Remark C.2. Note that, since each u_{t-j} is a one-hot vector, the tensor product $u_t \otimes u_{t-1} \otimes \cdots \otimes u_{t-n+1}$ is also one-hot. Hence q_{t+1} is obtained by selecting exactly one column of $\mathcal{G}_{t,n}$.

One of the drawbacks of the nonlinear feature $\phi_t^{(n)}$ is that the size of the unknown matrix/operator $\mathcal{G}_{t,n}$ grows exponentially with the history length n. Hence, the sample complexity of learning $\mathcal{G}_{t,n}$ also scales with exponentially with n. In order to mitigate this issue, we note that the column space of $\mathcal{G}_{t,n}$ lies in a low dimensional subspace space, because the effective dimensionality of the unknown parameter space is $M^2 + MN$ (instead of N^{n+1}).

C.2 Linear Features for n-gram Prediction

In this section, we show that, we can learn an efficient n-gram prediction function for q_{t+1} by exploiting the special structure of the one-hot nonlinear feature $\phi_t^{(n)}$ in equation 7. Specifically, we will show how the nonlinear features $\phi_t^{(n)}$ encodes the linear features studied in Hao et al. (2025).

Lemma C.3 (Features relation)**.** *Consider the same setup of Lemma C.1. Furthermore, let* $\psi_t^{(n)} := [u_t^\top \;\; u_{t-1}^\top \;\; \cdots \;\; u_{t-n+1}^\top]^\top \in \mathbb{R}^{nN}$ *denote a linear feature of* n *past observations. Then, there exists matrices* $\mathbf{W}_{lin} \in \mathbb{R}^{nN \times N^n}$, $\mathbf{W}_{kron} \in \mathbb{R}^{N^n \times nN}$, *and a vector* $v \in \mathbb{R}^{N^n}$ *such that,*

$$\psi_t^{(n)} = \mathbf{W}_{lin} \phi_t^{(n)}, \qquad \textit{and} \qquad \phi_t^{(n)} = \mathrm{ReLU}\left(\mathbf{W}_{kron} \psi_t^{(n)} + v \right) \tag{11}$$

where $\mathrm{ReLU}(x) := x^+ = \max(0, x)$*, and is applied entry-wise to vectors.*

Proof. We first show the existence of a matrix $\mathbf{W}_{lin} \in \mathbb{R}^{nN \times N^n}$ such that, $\psi_t^{(n)} = \mathbf{W}_{lin} \phi_t^{(n)}$. To begin, let $e_1, \ldots, e_N$ denote the standard basis vectors of $\mathbb{R}^N$. If we set the column of $\mathbf{W}_{lin}$ indexed by $(k_0, \ldots, k_{n-1}) \in [N]^n$ to be

$$\mathbf{W}_{lin}[:, (k_0, \ldots, k_{n-1})] = \begin{bmatrix} e_{k_0} \\ e_{k_1} \\ \vdots \\ e_{k_{n-1}} \end{bmatrix} \in \mathbb{R}^{nN} \tag{12}$$

then, it is easy to see that $\phi_t^{(n)}$ will pick the column of $\mathbf{W}_{lin}$ which is exactly $\psi_t^{(n)}$, that is, we have $\psi_t^{(n)} = \mathbf{W}_{lin} \phi_t^{(n)}$. Next, we show the existence of a matrix $\mathbf{W}_{kron} \in \mathbb{R}^{N^n \times nN}$, and a vector $v \in \mathbb{R}^{N^n}$ such that, $\phi_t^{(n)} = \mathrm{ReLU}\left(\mathbf{W}_{kron} \psi_t^{(n)} + v \right)$. Let $\varphi_t^{(n)}$ be a vector in $\mathbb{R}^{N^n}$ such that, its entries indexed by $(k_0, \ldots, k_{n-1}) \in [N]^n$ are given by

$$\varphi_t^{(n)}[(k_0, \ldots, k_{n-1})] = \sum_{j=0}^{n-1} u_{t-j}[k_j]. \tag{13}$$

Since $\{u_t\}_{t \geq 0}$ are one-hot vectors, we have

$$\varphi_t^{(n)}[(k_0, \ldots, k_{n-1})] \quad \begin{cases} = n, & \text{if } u_t = e_{k_0}, u_{t-1} = e_{k_1}, \ldots, u_{t-n+1} = e_{k_{n-1}}, \\ < n, & \text{otherwise.} \end{cases} \tag{14}$$

Applying ReLU function after subtracting $n-1$ from each entry of $\varphi_t^{(n)}$, we have

$$\text{ReLU}\left(\varphi_t^n[(k_0,\ldots,k_{n-1})]-n+1\right)=\begin{cases}1, & \text{if } u_t{=}e_{k_0}, u_{t-1}{=}e_{k_1},\ldots,u_{t-n+1}{=}e_{k_{n-1}},\\ 0, & \text{otherwise.}\end{cases} \tag{15}$$

Comparing equation 15 with equation 7, we have

$$\text{ReLU}\left(\varphi_t^{(n)}-(n-1)\cdot\mathbf{1}\right)=u_t\otimes u_{t-1}\otimes\cdots\otimes u_{t-n+1}=\phi_t^{(n)}, \tag{16}$$

where $\mathbf{1}$ is a vector of all ones in $\mathbb{R}^{N^n}$. To complete the proof, we need to show the existence of a matrix $\mathbf{W}_{kron}\in\mathbb{R}^{N^n\times nN}$ such that $\varphi_t^{(n)}=\mathbf{W}_{kron}\psi_t^{(n)}$. If we set the row of $\mathbf{W}_{kron}$ indexed by $(k_0,\ldots,k_{n-1})\in[N]^n$ to be

$$\mathbf{W}_{kron}[(k_0,\ldots,k_{n-1}),:]=\begin{bmatrix}e_{k_0}^\top & e_{k_1}^\top & \cdots & e_{k_{n-1}}^\top\end{bmatrix}\in\mathbb{R}^{nN} \tag{17}$$

then, it is easy to see that $\mathbf{W}_{kron}\psi_t^{(n)}=\varphi_t^{(n)}$. Combining this with equation 16, we have

$$\text{ReLU}\left(\mathbf{W}_{kron}\psi_t^{(n)}-(n-1)\cdot\mathbf{1}\right)=u_t\otimes u_{t-1}\otimes\cdots\otimes u_{t-n+1}=\phi_t^{(n)}. \tag{18}$$

This completes the proof. □

Remark C.4. From the Proof of Lemma C.3, it is easy to see that $\mathbf{W}_{kron}=\mathbf{W}_{lin}^\top$.

Lemma C.3 shows that the nonlinear features $\phi_t^{(n)}$ can be obtained from the linear features $\psi_t^{(n)}$ after passing it through a ReLU network. However, this increases the dimension of the feature space exponentially. In the following, we will identify conditions, under which linear features $\psi_t^{(n)}$ can be used to approximate the prediction q_{t+1}.

C.3 When does Linear Features $\psi_t^{(n)}$ provide accurate prediction?

First, we show that the matrix $\mathbf{W}_{lin}$ in Lemma C.3 is not invertible (rank-deficient). Hence, we cannot construct the nonlinear features $\phi_t^{(n)}$ from the linear features $\psi_t^{(n)}$ by simple matrix inversion.

Proposition C.5 (Linear approximation)**.** *Consider the same setup of Lemmas C.1, and C.3. We have, (a)* $\text{rank}(\mathbf{W}_{lin})<nN$*; and (b) The linear features* $\psi_t^{(n)}$ *can be used to approximate the prediction* q_{t+1}*, if there exists a matrix/operator* $\mathcal{K}_{t,n}\in\mathbb{R}^{N\times nN}$ *such that* $\mathcal{G}_{t,n}\approx\mathcal{K}_{t,n}\mathbf{W}_{lin}$*.*

Proof. Since every column of $\mathbf{W}_{lin}$ contains exactly one 1 in each of the n blocks, the block sums are identical. Thus there are $n-1$ linear dependencies among the rows. This implies that

$$\text{rank}(\mathbf{W}_{lin})=\text{rank}(\mathbf{W}_{lin}\mathbf{W}_{lin}^\top)<nN \qquad \text{for } n>1.$$

To prove the second statement of Proposition C.5, recall from Lemma C.1, that we have, $q_{t+1}=\mathcal{G}_{t,n}\,\phi_t^{(n)}$. Hence, if there exists a matrix/operator $\mathcal{K}_{t,n}\in\mathbb{R}^{N\times nN}$ such that $\mathcal{G}_{t,n}\approx\mathcal{K}_{t,n}\mathbf{W}_{lin}$, then we have

$$q_{t+1}\approx\mathcal{K}_{t,n}\mathbf{W}_{lin}\,\phi_t^{(n)}=\mathcal{K}_{t,n}\,\psi_t^{(n)}, \tag{19}$$

where we get the last equality from Lemma C.3. This completes the proof. □

C.4 Transformer Construction for $\phi_t^{(n)}$ and $\psi_t^{(n)}$

In this section, we can show that we can construct a transformer architecture with either $\log(n)$ layers or a single layer with n attention heads to extract the nonlinear feature $\phi_t^{(n)}$ (hence the linear feature $\psi_t^{(n)}$) from the one-hot vectors $u_t, u_{t-1},\ldots,u_0$.

Lemma C.6 (Transformer realization of $\phi_t^{(n)}$)**.** *Fix the history length* $n\geq 1$*. Suppose the lagged observations (one-hot)* $u_t,u_{t-1},\ldots,u_0$ *are provided to a transformer as separate tokens together with positional encoding that uniquely identify the lags* $0,1,\ldots,t$*. Let* $\phi_t^{(n)}$*,* $\psi_t^{(n)}$ *be as defined in Lemma C.3. Then, there exists, either (a) finite-depth multi-head transformer (*n *heads), or (b)* $\mathcal{O}(\log(n))$*-depth transformer, with feed-forward MLP sub-layers that computes* $\phi_t^{(n)}$ *(and* $\psi_t^{(n)}$ *) exactly.*

Proof. First, we provide the proof of claim (a). Our main observation is that attention can gather the lagged symbols into $\psi_t^{(n)}$ as follows: Introduce a dedicated summary token s. Using the positional encoding, assign one attention head to each lag $j \in \{0, \dots, n-1\}$. Choose the query of head j at the summary token so that it attends only to the token carrying lag j. Choose the value projection of that head to copy the content vector of that token into a reserved N-dimensional slot of the summary token. After concatenating the outputs of the n heads, the summary token contains $\psi_t^{(n)} := [u_t^\top \;\; u_{t-1}^\top \;\; \cdots \;\; u_{t-n+1}^\top]^\top \in \mathbb{R}^{nN}$. Combining this with Lemma C.3, it is evident that, we need one feed-forward MLP layer to map $\psi_t^{(n)}$ to $\phi_t^{(n)}$. Thus, one multi-head attention layer suffices to recover the stacked lagged history exactly from the raw tokens, provided the positions are identifiable. Moreover, one multi-head attention layer with one feed-forward (ReLU) layer suffices to recover the Kronecker products of lagged history exactly.

To provide the proof of claim (b), we construct $\phi_t^{(n)}$ by a binary tree as follows: For simplicity, first we assume that $n = 2^r$ for some integer $r > 0$. This is without loss of generality and changes the depth only by a constant. At layer 0, the Transformer stores the n vectors

$$v_j^{(0)} := u_{t-j} \in \mathbb{R}^N, \qquad j = 0, 1, \dots, n-1.$$

Assuming that a single attention head followed by a feed-forward MLP later can perform a tensor product (note that $u \otimes v = \mathrm{vec}(uv^\top)$) of two adjacent tokens, at layer 1, it performs pairwise tensor products

$$v_j^{(1)} := v_{2j}^{(0)} \otimes v_{2j+1}^{(0)} \in \mathbb{R}^{N^2}, \qquad j = 0, \dots, n/2 - 1.$$

Similarly, at layer 2, it performs pairwise tensor products

$$v_j^{(2)} := v_{2j}^{(1)} \otimes v_{2j+1}^{(1)} \in \mathbb{R}^{N^4}, \qquad j = 0, \dots, n/4 - 1$$

Continuing in this way, after $r = \log_2(n)$ layers, there is a single vector

$$v_0^{(r)} = u_t \otimes u_{t-1} \otimes \cdots \otimes u_{t-n+1} = \phi_t^{(n)}.$$

Lastly, we will prove our claim that a Transformer layer can perform pairwise tensor products as follows: Given two adjacent tokens u, v, from Lemma C.3 a single ReLU-layer maps (u, v) to $u \otimes v$. Hence, using an ReLU network in each layer of the transformer, we can guarentee that each Transformer layer can perform pairwise tensor products. This proves our claim that an $\mathcal{O}(\log(n))$-depth transformer, with feed-forward MLP sub-layers (specifically ReLU) can compute $\phi_t^{(n)}$ (hence, $\psi_t^{(n)}$ through a linear map) exactly. This completes the proof. □

Remark C.7 (Role of attention vs. MLP). The attention layer performs routing, that is, it collects the lagged one-hot vectors into a common summary representation. The nonlinear interaction across positions is carried out by the feed-forward ReLU layer. In particular, the exact computation of $\phi_t^{(n)}$ is not a purely additive single-head attention phenomenon, it relies on higher-order cross-position interactions.

C.5 Transformer can Implement Gradient Descent on the Cross-Entropy Loss

In the previous section, we showed that both linear and nonlinear features can be realized by Transformers. In this section, we will build on this tow show that Transformers can also implement Gradient Descent algorithm to minimize the cross-entropy loss of predicting q_{t+1} from $\phi_t^{(n)}$ or $\psi_t^{(n)}$ via a linear map.

Theorem C.8 (Log-depth construction for CE-GD)**.** *Consider the same setup of Lemmas C.1, and C.3. Furthermore, consider a dataset*

$$\mathcal{D}_m = \{(\mathfrak{h}_t, y_t)\}_{t=n}^{n+m-1}, \qquad \mathfrak{h}_t = (u_t, u_{t-1}, \dots, u_{t-n+1}), \qquad y_i \in \{e_1, \dots, e_N\}. \tag{20}$$

Let $\mathbf{W} \in \mathbb{R}^{N \times N^n}$ be the parameter of multinomial logistic regression, $q_{\mathbf{W}}(\cdot \mid \phi) = \mathrm{softmax}(\mathbf{W}\phi)$, trained by full-batch gradient descent on the empirical cross-entropy loss

$$\mathcal{L}_m(\mathbf{W}) := \frac{1}{m} \sum_{t=n}^{n+m-1} \ell(\mathbf{W}\phi_t^{(n)}, y_t), \qquad \ell(z, y) := -y^\top \log \mathrm{softmax}(z),$$

If the number of GD iterations satisfies $T = \mathcal{O}(\log(n))$, then there exists a Transformer of depth $\mathcal{O}(\log(n))$ that emulates T steps of full-batch GD on the CE loss of the logistic-regression model $q_{\mathbf{W}}(\cdot \mid \phi) = \mathrm{softmax}(\mathbf{W}\phi)$.

Proof. First, from Lemma C.6, we know that Transformers can realize both $\phi_t^{(n)}$ and ψ_t^n. Next, we consider logistic regression on these features. Let $\phi_t^{(n)} \in \mathbb{R}^{N^n}$ is indexed by $\alpha_t := (k_0, \ldots, k_{n-1}) \in [N]^n$. Then, using the fact that $\phi_t^{(n)}$ is a one-hot vector, we have

$$\mathbf{W}\phi_t^{(n)} = \mathbf{W}[:, (k_0, \ldots, k_{n-1})] = \mathbf{W}[:, \alpha_t]. \tag{21}$$

Now group the samples according to their active history index α_t, that is, collecting together all samples whose nonlinear feature equals the same basis vector e_{α_t}, we have

$$\mathcal{L}_m(\mathbf{W}) = \frac{1}{m} \sum_{t=n}^{n+m-1} \ell(\mathbf{W}[:, \alpha_t], y_t) = \frac{1}{m} \sum_{\alpha \in [M]^L} \sum_{t=n}^{n+m-1} \mathbf{1}\{\alpha_t = \alpha\} \ell(\mathbf{W}[:, \alpha], y_t). \tag{22}$$

Thus the GD dynamics are column-wise decoupled. For a single sample,

$$\nabla_{\mathbf{W}} \ell(\mathbf{W}\phi_t^{(n)}, y_t) = \left(\mathrm{softmax}(\mathbf{W}\phi_t^{(n)}) - y_t\right) \left(\phi_t^{(n)}\right)^\top. \tag{23}$$

Using $\phi_t^{(n)} = e_{\alpha_t}$, this becomes

$$\nabla_{\mathbf{W}} \ell(\mathbf{W}\phi_t^{(n)}, y_t) = \left(\mathrm{softmax}(\mathbf{W}[:, \alpha_t]) - y_t\right) e_{\alpha_t}^\top. \tag{24}$$

Hence, the sample t updates only the column $\mathbf{W}[:, \alpha_t]$. For the query-relevant column θ_{α_q} (where $q = n + m$), define the matching-history count and label-count vector

$$N_{\alpha_q} := \sum_{t=n}^{n+m-1} \mathbf{1}\{\alpha_t = \alpha_q\}, \qquad c_{\alpha_q} := \sum_{t=n}^{n+m-1} \mathbf{1}\{\alpha_t = \alpha_q\} y_t \in \mathbb{R}^N. \tag{25}$$

Then the full-batch GD recursion for the query-relevant column is

$$\mathbf{W}[:, \alpha_q]^{(s+1)} = \mathbf{W}[:, \alpha_q]^{(s)} - \frac{\eta}{m} \left(N_{\alpha_q} \mathrm{softmax}(\mathbf{W}[:, \alpha_q]^{(s)}) - c_{\alpha_q}\right), \quad s = 0, 1, \ldots, T-1. \tag{26}$$

Moreover, $\mathrm{softmax}\left(\mathbf{W}^{(T)}\phi_q^{(n)}\right) = \mathrm{softmax}\left(\mathbf{W}^{(T)}e_{\alpha_q}\right) = \mathrm{softmax}\left(\mathbf{W}[:, \alpha_q]^{(T)}\right)$. Thus to emulate GD for the query, the Transformer does not need to update all N^n columns of $\mathbf{W}$. It only needs to compute the sufficient statistics

$$(N_{\alpha_q}, c_{\alpha_q}) \tag{27}$$

for the history class matching the query, and then simulate the N-dimensional recursion above. In the following, we will show that a Transformer can compute these sufficient statistics in $\mathcal{O}(\log(n))$ depth. For each training example t, the Transformer compares its history $\mathfrak{h}_t = (u_t, \ldots, u_{t-n+1})$ with the query history $\mathfrak{h}_q = (u_q, \ldots, u_{q-n+1})$. For every lag $j \in [n-1]$,

$$u_{t-j}^\top u_{q-j} = \begin{cases} 1, & \text{iff} \quad u_{t-j} = u_{q-j} \\ 0, & \text{otherwise} \end{cases} \tag{28}$$

Therefore, we have

$$r_t := \sum_{j=0}^{n-1} u_{t-j}^\top u_{q-j} \quad \begin{cases} = n, & \text{iff} \quad h_t = h_q \\ < n, & \text{otherwise} \end{cases} \tag{29}$$

The sum defining r_t can be computed by a parallel binary reduction over the n lags, using $\mathcal{O}(\log(n))$ layers. A thresholding feed-forward map then computes

$$\mu_t := \mathbf{1}\{\mathfrak{h}_t = \mathfrak{h}_q\} = \mathrm{ReLU}(r_t - n + 1), \tag{30}$$

because r_t is an integer in $\{0, 1, \ldots, n-1\}$. Once the selectors μ_t are available, attention aggregation computes

$$c_{\alpha_q} = \sum_{t=n}^{n+m-1} \mu_t y_t, \qquad N_{\alpha_q} = \sum_{t=n}^{n+m-1} \mu_t. \tag{31}$$

This aggregation can be performed by a summary token attending to the context examples with values $(y_t, 1)$. In the idealized hard-attention model this is exact; in the standard softmax model it can be approximated arbitrarily well on the finite input domain. Finally, define the one-step GD update map

$$F(\mathbf{W}[:, \alpha], c, N) := \mathbf{W}[:, \alpha] - \frac{\eta}{m} \left(N \operatorname{softmax}(\mathbf{W}[:, \alpha]) - c \right). \tag{32}$$

The map F is smooth in $\mathbf{W}[:, \alpha]$ and continuous in $(\mathbf{W}[:, \alpha], c, N)$. Hence a feed-forward MLP layer can approximate it uniformly. In the idealized hard-attention model, the block computes it exactly. Applying this block recurrently for T Transformer layers, and applying softmax to the final output layer gives

$$\operatorname{softmax}(\mathbf{W}[:, \alpha_q]^{(T)}) = \operatorname{softmax}\Big(\mathbf{W}^{(T)} \phi_q^{(n)} \Big). \tag{33}$$

The history matching and nonlinear feature construction require $\mathcal{O}(\log(n))$ depth, while the unrolled GD simulation requires $\mathcal{O}(T)$ depth. Therefore the total depth is $\mathcal{O}(\log(n) + T)$. In particular, when $T = \mathcal{O}(\log(n))$, the total depth is $\mathcal{O}(\log(n))$. This completes the proof. □

D Spectral Learning for Prediction

Recall from Appendix B.2 that the spectral learning algorithm estimates low-order moment statistics from the observation prefix and forms observation operators to recursively update a belief state. Following Hsu et al. (2012), define the uni-, bi-, and trigram moment matrices

$$[P_1]_i := \Pr[O_1 = i] \ \in \mathbb{R}^N, \tag{34}$$

$$[P_{2,1}]_{ij} := \Pr[O_2 = i,\, O_1 = j] \ \in \mathbb{R}^{N\times N}, \tag{35}$$

$$[P_{3,o,1}]_{ij} := \Pr[O_3 = i,\, O_2 = o,\, O_1 = j] \ \in \mathbb{R}^{N\times N}, \quad o \in \mathcal{O}, \tag{36}$$

where N denotes the number of distinct observations and M the number of hidden states. We use the one-hot encodings $u_s := e_{o_s} \in \mathbb{R}^N$ as defined in Appendix C.

Let $U \in \mathbb{R}^{N\times M}$ be the matrix of left singular vectors of $P_{2,1}$ corresponding to its M largest singular values. By Lemma 2 of Hsu et al. (2012), $\mathrm{range}(U) = \mathrm{range}(\mathbf{B}^\top)$, so U spans the same subspace as the emission matrix and $U^\top \mathbf{B}^\top$ is invertible. The observable representation is then defined by

$$\widetilde{b}_1 := U^\top P_1 \ \in \mathbb{R}^M, \tag{37}$$

$$\widetilde{b}_\infty := (P_{2,1}^\top U)^+ P_1 \ \in \mathbb{R}^M, \tag{38}$$

$$B_o := (U^\top P_{3,o,1})(U^\top P_{2,1})^+ \ \in \mathbb{R}^{M\times M}, \quad o \in \mathcal{O}, \tag{39}$$

where $(\cdot)^+$ denotes the Moore–Penrose pseudoinverse. By Lemma 3 of Hsu et al. (2012), the belief state $\widetilde{b}_t \in \mathbb{R}^M$ evolves as $\widetilde{b}_{t+1} \propto B_{o_t}\widetilde{b}_t$, and the conditional predictive satisfies

$$\Pr(O_{t+1} = o \mid O_{1:t} = o_1, \ldots, o_t) \ \propto \ \widetilde{b}_\infty^\top B_{o_t} \cdots B_{o_1}\, \widetilde{b}_1. \tag{40}$$

In-context, P_1, $P_{2,1}$, and $P_{3,o,1}$ are replaced by their empirical estimates from the observation prefix, and U is estimated as the top-M left singular vectors of $\widehat{P}_{2,1}$.

The core computation in equation 40 decomposes into three primitive operations on the observation sequence $o_{1:t}$:

- $f_{\mathrm{uni}}(o_{1:t}) := U^\top P_1 \in \mathbb{R}^M$, a linear function of the empirical unigram frequencies.
- $f_{\mathrm{bi}}(o_{1:t},\, v) := (U^\top P_{2,1})^{-1} v \ \in \mathbb{R}^M$, applying the projected inverse bigram matrix to $v \in \mathbb{R}^M$.
- $f_{\mathrm{tri}}(o_{1:t},\, v,\, o) := U^\top P_{3,o,1}\, v \in \mathbb{R}^M$, applying the projected trigram operator for symbol o to $v \in \mathbb{R}^M$.

Note that $B_o = f_{\mathrm{bi}}(o_{1:t},\, f_{\mathrm{tri}}(o_{1:t},\, \cdot,\, o))$, so each belief-state update $\widetilde{b}_{t+1} \propto B_{o_t}\widetilde{b}_t$ is one application of f_{tri} followed by f_{bi}. The full prediction equation 40 is thus a sequential composition of $\mathcal{O}(t)$ such pairs, followed by an ℓ_1 normalization. Explicitly:

1. Initialize $\widetilde{b}_1 = f_{\mathrm{uni}}(o_{1:t})$.
2. For $k = 1, \ldots, t-1$:
   - Set $\widetilde{b}'_k = f_{\mathrm{tri}}(o_{1:t},\, \widetilde{b}_k,\, o_k)$.
   - Set $\widetilde{b}_{k+1} = f_{\mathrm{bi}}(o_{1:t},\, \widetilde{b}'_k)$.

   After this loop, $\widetilde{b}_t$ is the updated belief after processing $o_1, \ldots, o_{t-1}$.
3. For each $o \in \mathcal{O}$:
   - Set $\widetilde{b}^o_{t+1} = f_{\mathrm{tri}}(o_{1:t},\, \widetilde{b}_t,\, o)$.
   - Set $c^o = f_{\mathrm{bi}}(o_{1:t},\, \widetilde{b}^o_{t+1})$.
4. Output for symbol o: $\widetilde{b}_\infty^\top c^o$, where $\widetilde{b}_\infty = (P_{2,1}^\top U)^+ P_1$ is the fixed terminal vector from the observable representation defined above. Normalize over $o \in \mathcal{O}$.

We now argue that each primitive and their composition can be approximated by a Transformer.

Approximating f_{uni}. Since $P_1 = \mathbb{E}[u_t]$ and U is a fixed matrix (estimated once from the prefix via SVD of $\widehat{P}_{2,1}$), the map $o_{1:t} \mapsto U^\top \widehat{P}_1$ is a linear function of the empirical unigram counts, implementable by a single attention layer that averages the token embeddings.

Approximating f_{bi}. We need to implement $v \mapsto (U^\top P_{2,1})^{-1} v$ in-context. By the Neumann series identity, $(U^\top P_{2,1})^{-1} v \approx (I - U^\top P_{2,1})^k v$ for appropriately normalized $P_{2,1}$ and sufficiently large k, so it suffices to implement k repeated multiplications by $U^\top P_{2,1}$ Akyürek et al. (2023). The map $v \mapsto U^\top P_{2,1} v$ amounts to an empirical bigram-weighted sum projected by U, which can be implemented by a Transformer attention layer following Ekbote et al. (2025).

Approximating f_{tri}. We need to implement $v \mapsto U^\top P_{3,o,1}\, v$ in-context for a general belief vector $v \in \mathbb{R}^M$. We proceed in two steps.

First, consider the case $v = u_s$ for some one-hot observation vector $u_s \in \mathbb{R}^N$. Following the construction of Varre et al. (2025) (Section 4.1), a trigram estimator for this case can be realized by a two-layer Transformer with $n-1$ attention heads and no residual connection (disentangled architecture). Concretely, their construction computes

$$\frac{\sum_{\ell=3}^{t} u_{\ell-1}^\top u_t \cdot u_{\ell-2}^\top u_{t-1} \cdot u_\ell}{\sum_{\ell=3}^{t} u_{\ell-1}^\top u_t \cdot u_{\ell-2}^\top u_{t-1}},$$

where $u_{\ell-2}^\top u_{t-1}$ acts as a $0/1$ indicator that selects context positions whose second lag matches the query's second lag. This implements $U^\top P_{3,o,1}\, v$ when $v = u_{t-1}$ is itself a one-hot vector from the observed sequence.

Second, for a general belief vector $v \in \mathbb{R}^M$ already present in the residual stream, we observe that $U^\top P_{3,o,1}\, v$ is *linear* in v. Writing $v = \sum_i v_i e_i$, we have $U^\top P_{3,o,1}\, v = \sum_i v_i\, U^\top P_{3,o,1}\, e_i$. Since each e_i is a one-hot vector, each term $U^\top P_{3,o,1}\, e_i$ is computable by the construction above. The linear combination against the coefficients v_i can then be formed by a feed-forward layer using the belief vector already in the residual stream. Thus f_{tri} is approximable by a Transformer for any $v \in \mathbb{R}^M$.

Approximating the composition. The full algorithm requires composing $\mathcal{O}(t)$ alternating applications of f_{tri} and f_{bi}, each acting on the current M-dimensional belief vector $\widetilde{b}_k \in \mathbb{R}^M$. Crucially, the operators $B_{o_k} = f_{\text{bi}} \circ f_{\text{tri}}(\,\cdot\,, o_k)$ are never explicitly materialized as $M \times M$ matrices; instead, the binary tree construction of Hu et al. (2024) (Appendix D, eq. D.12 onwards) composes the M-dimensional vector maps $\widetilde{b} \mapsto f_{\text{bi}}(f_{\text{tri}}(\widetilde{b}, o_k))$ directly, with the in-context bigram and trigram statistics reused at each node of the tree. This yields $\mathcal{O}(\log t)$ depth for a length-t composition. With a context of m in-context sequences each of history length n, the total prefix length satisfies $t = \mathcal{O}(n+m)$, giving an overall Transformer depth of $\mathcal{O}(\log(n+m))$, as stated in Remark 3.3.

E SMALL TRAINED TRANSFORMER

Given a HMM configuration (A, B, π), we generate 8k sequences with length 2,049, train on 4k sequences, and validate on the rest 4k sequences.

E.1 ABLATION ON MODEL CONFIG

We first fix an HMM configuration with 3 states, 12 observations, 0.25 A entropy, and 0.75 B entropy. We report the training loss, validation loss, and accuracy on location $\{2, 4, 8, ..., 1024, 2048\}$.

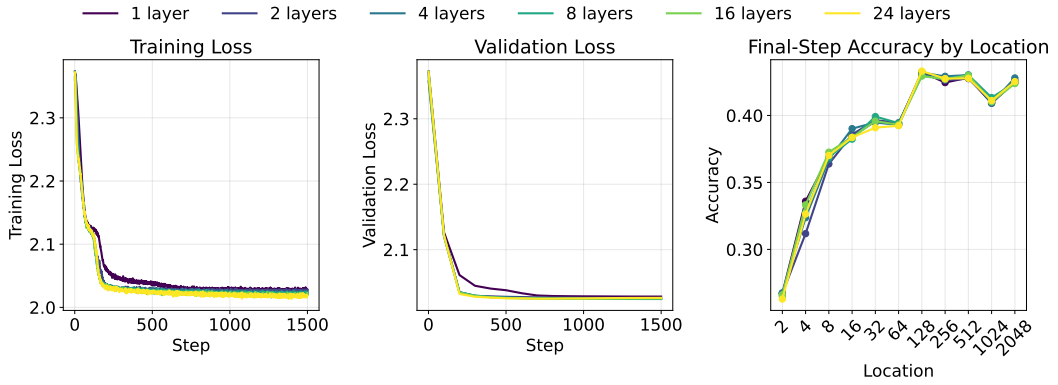


Figure 13: Training loss, validation loss, and accuracies with different number of layers.

More number of layers does not lead to better performance. We fix the model dimension to 64, feedforward dimension to 256, and number of heads to 4, and ablate the number of layers from 1 to 24. The results are shown in Figure 13. Across all runs, the training loss, validation loss, and accuracies converge to a similar point.

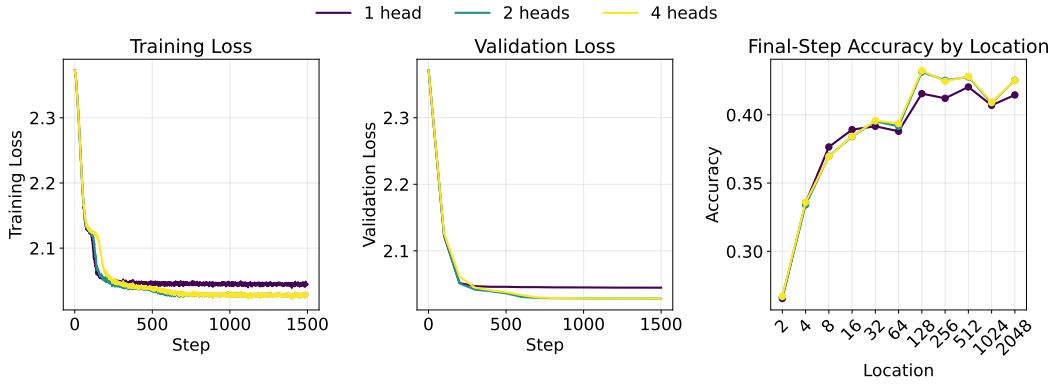


Figure 14: Training loss, validation loss, and accuracies with different number of heads.

More number of heads give slight performance improvement. We fix the model dimension to 64, feedforward dimension to 256, and number of layers to 1, and ablate the number of heads from 1 to 4. The results are shown in Figure 14. Increasing number of heads above 1 would have slightly improvement in training / validation losses and accuracies.

Smaller dimension leads to underfitting, while larger dimension does not lead to better performance. We fix the number of layers to 1 and number of heads to 1, and ablate the (model dimension, feedforward dimension) from (8, 32) to (128, 512). The results are shown in Figure 15. For size (8, 32) and (16, 64), we can see underfitting based on training and validation loss. We need at least (32, 128), and increasing beyond (64, 256) does not lead to better performance.

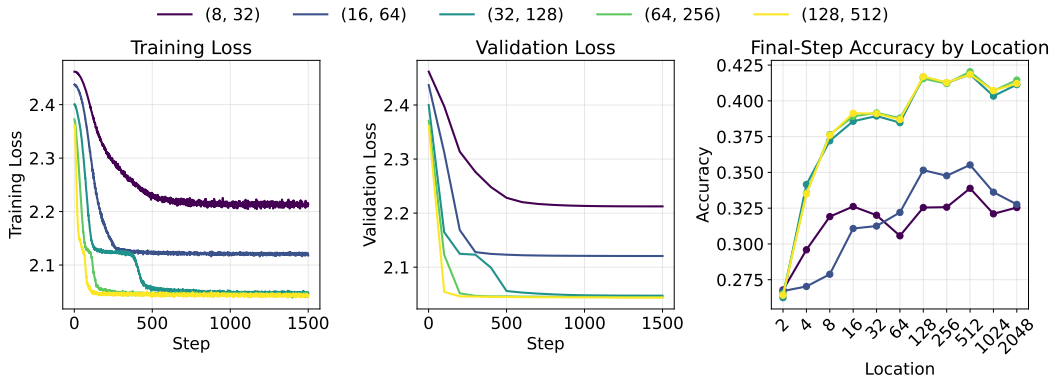


Figure 15: Training loss, validation loss, and accuracies with different model dimensions.

E.2 Ablation on HMM Config

Based on the previous section, we fix the model config to 1 head, 1 layer, 64 model dimension, and 256 feedforward dimension.

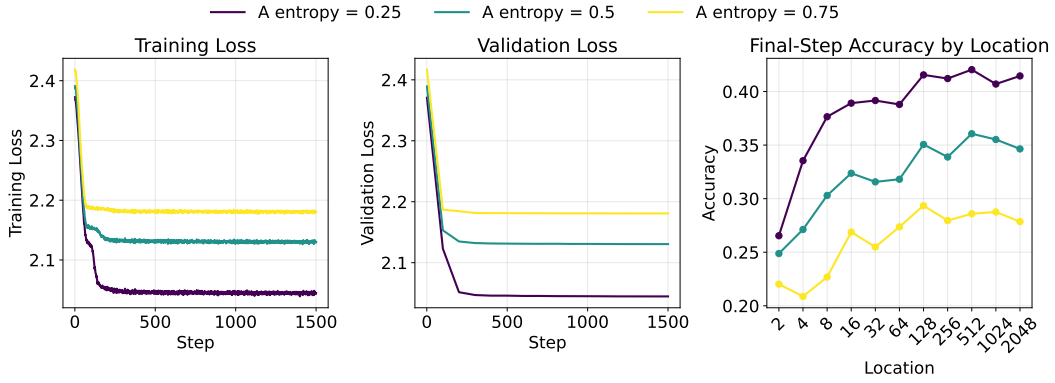


Figure 16: Training loss, validation loss, and accuracies with different A entropies.

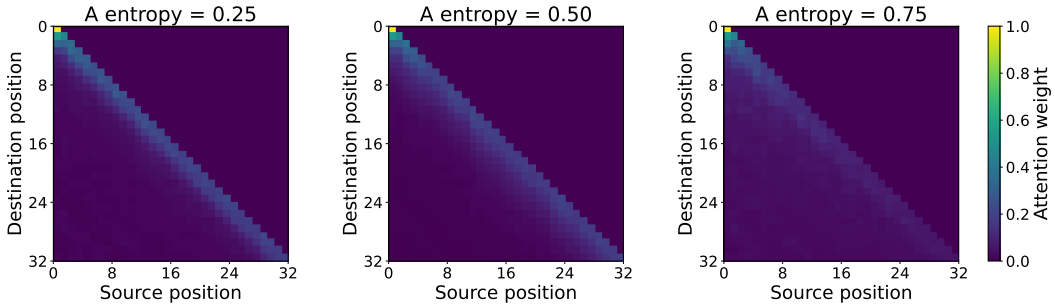


Figure 17: Attention pattern for different A entropies.

Slower mixing leads to longer attention window. We fix the HMM config to 3 states and 12 observations with B entropy 0.75. We ablate the A entropy, $\{0.25, 0.5, 0.75\}$. The results are shown in Figure 16. As we increase A entropy, the training and validation losses become worse which is expected as the current task becomes harder. Therefore, the accuracy also decreases with larger A entropy. The attention visualization is provided in Figure 17. From the attention visualization, we can see that, as A entropy increases (i.e. slower mixing), the attention window also expands. When the

HMM has slower mixing, the model needs to incorporate a longer context in order to make accurate predictions.

Larger B entropy leads to longer attention window. We fix the HMM config to 3 states and 12 observations with A entropy 0.25. We ablate the B entropy, $\{0.25, 0.5, 0.75\}$. The results are shown in Figure 18. As we increase B entropy, again the training and validation losses become worse as the current task becomes harder. The accuracy also decreases with larger B entropy. The attention visualization is provided in Figure 19. From the attention visualization, we can see that, as B entropy increases, the attention window also expands.

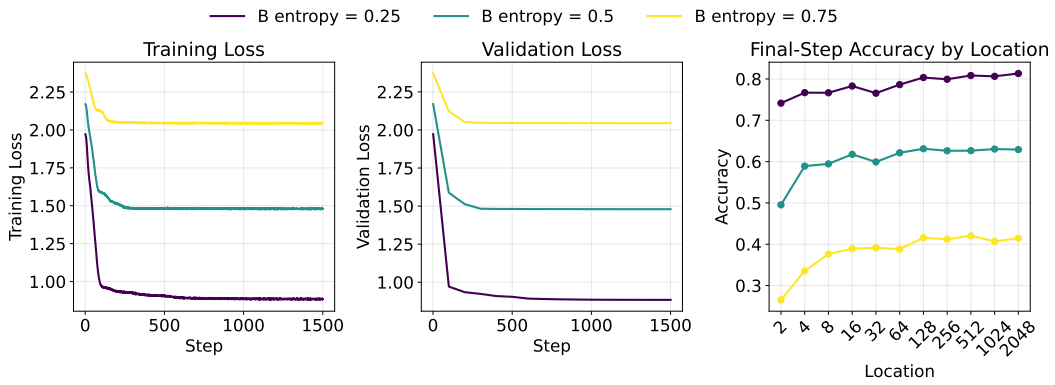


Figure 18: Training loss, validation loss, and accuracies with different B entropies.

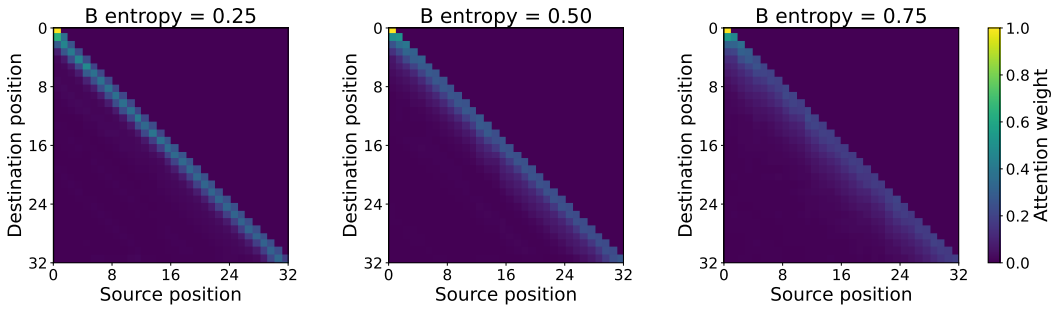


Figure 19: Attention pattern for different B entropies.

Number of observations does not affect attention window. We fix the HMM config to 3 states with A entropy 0.25 and B entropy 0.75. We ablate the number of observations, $\{3, 6, 12\}$. The results are shown in Figure 20. As we increase the number of observations, again the training and validation losses become worse as the current task becomes harder. The accuracy also decreases with more number of observations. However, the attention visualizations show in Figure 21 for different number of observations remain consistent. Attention pattern could be highly correlated with the entropies but not the number of observations.

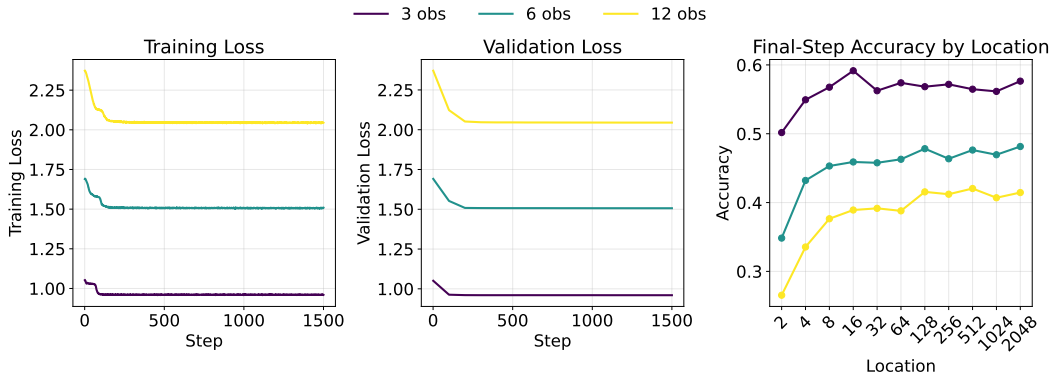


Figure 20: Training loss, validation loss, and accuracies with different number of observations.

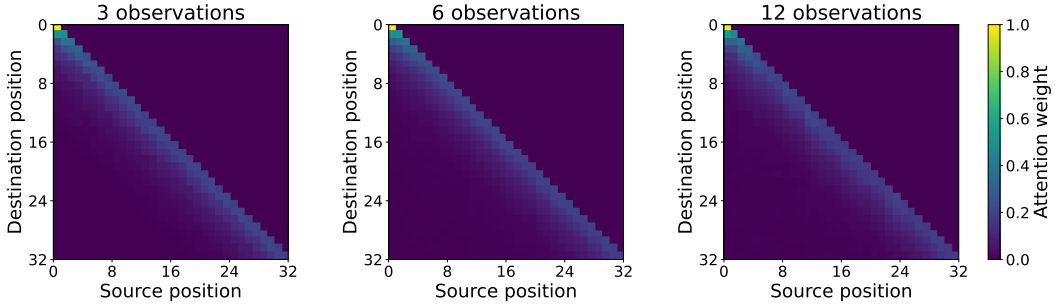


Figure 21: Attention pattern for different number of observations.

E.3 Ablation on Window Size n

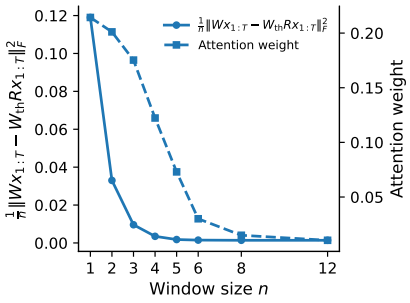
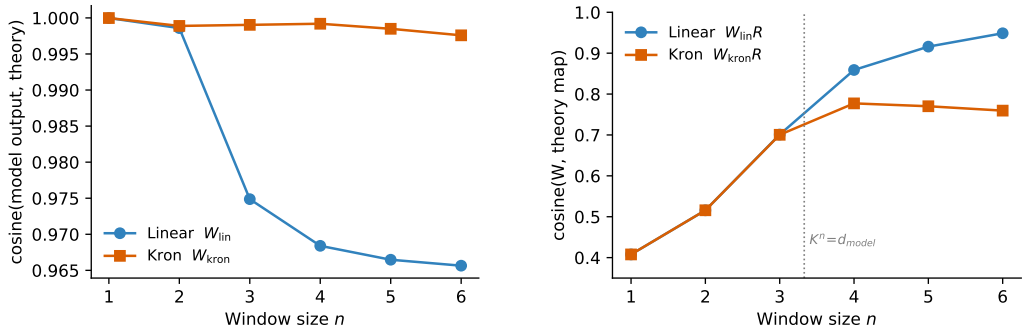


Figure 22: Discrepancy between the two mappings and attention weights.

The effective attention window matches the finite-window predictor. We further vary the window size n used by the theoretical finite-window predictor and measure the discrepancy between the direct hidden-to-logit map $Wx_{1:T}$ and the composed theoretical map $W_{\text{th}}Rx_{1:T}$. Figure 22 shows that this discrepancy decreases rapidly as n increases, and converges around $n = 6$. This is consistent with the attention pattern, where the attention mass concentrates mostly on the previous 6 tokens.

E.4 Linear vs. Kronecker n-gram Feature Mappings



(a) Output alignment

(b) Representation alignment

Figure 23: **Output vs. representation alignment for linear (additive, dimension Kn) and Kronecker (tabular, dimension K^n) n-gram features.** (a) Cosine similarity between the model's next-token logits and each theoretical predictor, aggregated by length-n context. (b) Cosine similarity between the learned hidden-to-logit readout W and the theory-induced map $W_{\text{th}}R$ at the post-attention residual stream.

The transformer predicts like the tabular model, but its post-attention representation is additive. We contrast two ways of representing the length-n context: the *linear* features, which stack the one-hot encodings of the past n observations (dimension Kn), and the *Kronecker* features, which form their full Cartesian product (dimension K^n, one coordinate per distinct context tuple). We probe these features at two levels. For *output alignment* (Figure 23a), we aggregate the model's next-token logits by context and measure the cosine similarity to each theory's per-context prediction (W_{kron}, and the additive W_{lin} projected onto contexts). For *representation alignment* (Figure 23b), we fit a linear map R from the post-attention residual stream $x_{1:T}$ to the features and measure the cosine similarity between the directly learned readout W and the composed map $W_{\text{th}}R$. The probe R is fit by ridge regression on every context position of the held-out validation split—4096 sequences of length 2049, i.e. $4096 \times (2049 - n) \approx 8.4 \times 10^6$ state/feature pairs—so the fit is heavily over-determined relative to its effective rank $d_{\text{model}} = 64$.

At the output level, the model's predictions match the Kronecker (tabular) predictor almost perfectly at every window (cosine ≈ 1.0), while the linear predictor's agreement decays with n (down to ≈ 0.97) as the additive restriction discards increasingly relevant context. At the representation level the ordering reverses: the linear mapping keeps improving (cosine $\to 0.95$), whereas the Kronecker mapping saturates at ≈ 0.76 once the number of tabular contexts K^n exceeds the model width d_{model}; because the probe is trained on millions of examples, this gap is not data-limited but reflects the capacity of the residual stream.

This double dissociation is exactly what our theory predicts. The output is tabular-optimal (aligned with the Kronecker predictor), yet the post-attention representation is additive (it linearly encodes only the linear features). By Lemma 3.1, the linear n-gram features cannot be mapped to the Kronecker features—and hence to the tabular-optimal next-token distribution—by a linear transformation alone; a non-linear layer (ReLU) is required. The attention block therefore assembles the additive features into the residual stream, and the subsequent non-linear MLP lifts them to the tabular prediction, which is why alignment is *linear* before the MLP but *Kronecker* at the output. Appendix C.3 characterizes the regimes in which the linear-feature approximation already yields accurate output predictions, consistent with the residual (≈ 0.97) agreement of the linear predictor in Figure 23a.

F LLMs Empirical Performance

We present detailed evaluation results for the experiment described in Section 2.3. Recall the setup: for each HMM configuration, we sample observation sequences $\mathbf{o}_{1:T}$ from $\boldsymbol{\lambda} = (\boldsymbol{\pi}, \mathbf{A}, \mathbf{B})$ and evaluate next-observation prediction o_{t+1} given $\mathbf{o}_{1:t}$. We fix number of hidden state $M = 4$ and vary three controls across 75 configurations: transition entropy $\mathcal{E}(\mathbf{A}) \in \{0,\ 0.25,\ 0.5,\ 0.75,\ 1\}$, inducing mixing rates $|\lambda_2(\mathbf{A})| \in \{1, 0.9, 0.75, 0.5, 0\}$; emission entropy $\mathcal{E}(\mathbf{B}) \in \{0,\ 0.25,\ 0.5,\ 0.75,\ 1\}$, ranging from injective (when $N \geq M$) to uniform; and alphabet size $N \in \{2, 4, 8\}$, where $N < M$ induces hidden-state aliasing. The initial distribution is set to the stationary distribution, $\boldsymbol{\pi} = \boldsymbol{\mu}$. For each configuration, we sample $4{,}096$ sequences of length $4{,}097$ and evaluate at context lengths $\{4, 8, \ldots, 4096\}$.

We report two metrics averaged over $4{,}096$ samples per HMM configuration. *Accuracy* is the fraction of positions at which the argmax prediction matches the ground truth. We additionally report Hellinger distance between predicted and reference next-token distributions, used in two ways: *Hellinger-to-oracle* measures deviation from the Bayes-optimal posterior; and *Hellinger-to-baseline* measures distributional similarity to each algorithmic baseline. In both cases, smaller values indicate better performance.

We evaluate 12 open-weight pretrained LLMs drawn from the Qwen, Llama, Gemma, and OLMo families, ranging from 0.6B to 8B parameters, without any fine-tuning. Each model receives raw observation tokens via a fixed alphabet-to-vocabulary mapping; at every position, we restrict the next-token logits to the observation vocabulary and renormalize using softmax.

The report is separated into the following subsections:

- Appendix F.1: Empirical Performance of Different LLM Families
- Appendix F.2: Cases When LLMs Fail to Converge to Oracle
- Appendix F.3: Baseline Comparison
- Appendix F.4: Soft n-gram with Varying n
- Appendix F.5: More Hidden States ($M = 12$)

F.1 Empirical Performance of Different LLM Families

For each LLM family, we present three plots corresponding to alphabet sizes $N \in \{2, 4, 8\}$. Each plot contains 25 subplots arranged over a grid of transition entropies $\mathcal{E}(\mathbf{A}) \in \{0,\ 0.25,\ 0.5,\ 0.75,\ 1\}$ and emission entropies $\mathcal{E}(\mathbf{B}) \in \{0,\ 0.25,\ 0.5,\ 0.75,\ 1\}$. Within each subplot, the upper panel shows top-1 prediction accuracy and the lower panel shows Hellinger-to-oracle distance, both as a function of context length. The `oracle` and `bigram` baselines are included in each subplot for reference.

Among the families, the Qwen models (Qwen2.5-1.5B, Qwen3-0.6B, Qwen3-1.7B, Qwen3-4B, Qwen3-8B) exhibit the most consistent convergence across model sizes, with larger models achieving higher accuracy at longer context lengths and converging closer to the oracle distribution. The Llama and Gemma families generally converge but show a larger gap from the oracle than Qwen under high-entropy conditions, with Llama3.1-8B suffering a sharp performance drop at context length 4096. The OLMo family (OLMo2-1B, OLMo2-7B) performs the worst overall: it degrades markedly at longer context lengths ($T \in \{1024, 2048, 4096\}$), exhibiting a pronounced U-shaped trend, and underperforms other families when either transition or emission entropies is high.

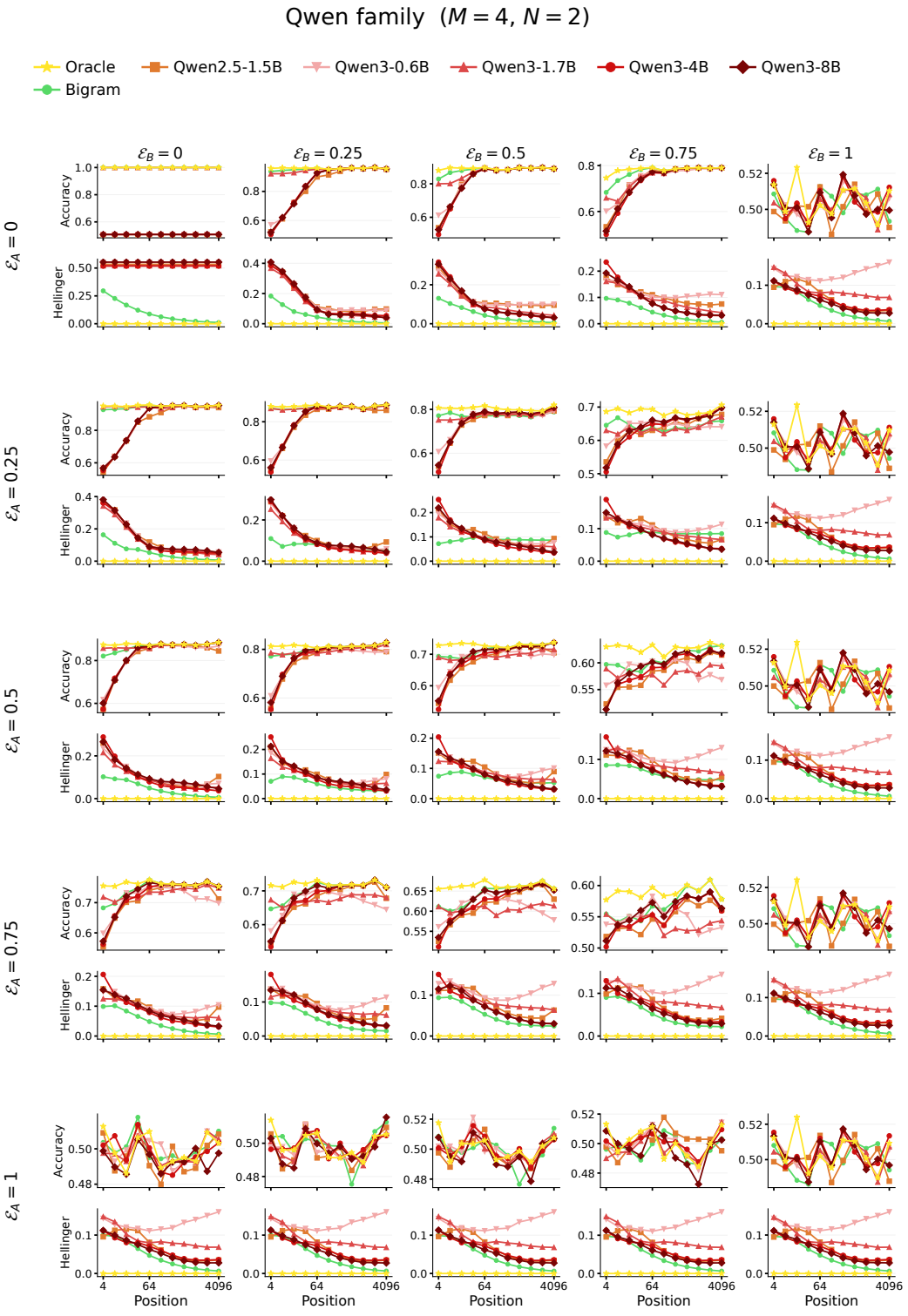


Figure 24: Qwen family LLMs evaluated on HMMs with $M = 4$, $N = 2$. Subplots vary transition entropy $\mathcal{E}(\mathbf{A})$ (rows) and emission entropy $\mathcal{E}(\mathbf{B})$ (columns); each shows top-1 accuracy (upper) and Hellinger-to-oracle distance (lower) versus context length, with `oracle` and `bigram` baselines.

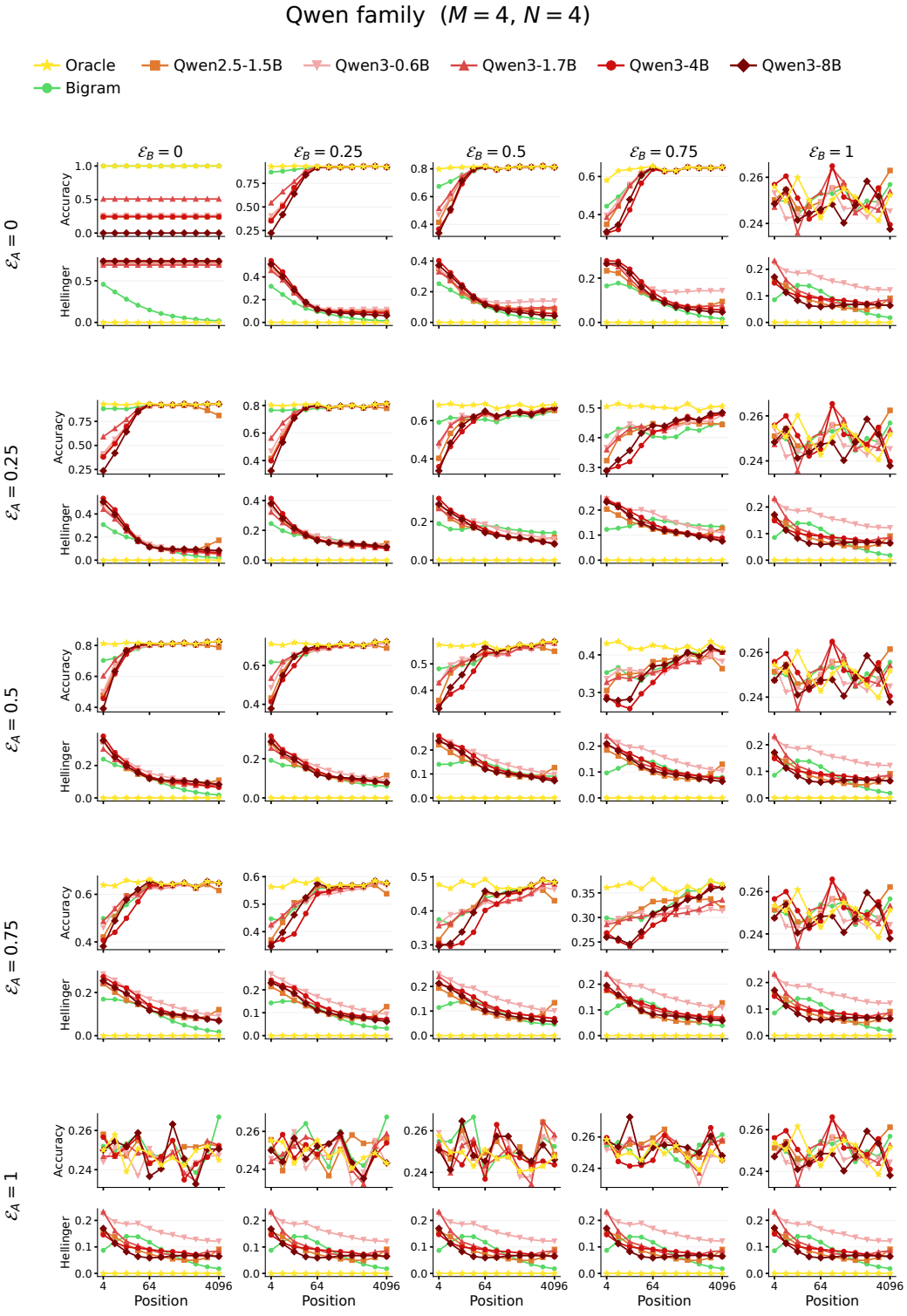


Figure 25: Qwen family LLMs evaluated on HMMs with $M = 4$, $N = 4$. Subplots vary transition entropy $\mathcal{E}(\mathbf{A})$ (rows) and emission entropy $\mathcal{E}(\mathbf{B})$ (columns); each shows top-1 accuracy (upper) and Hellinger-to-oracle distance (lower) versus context length, with `oracle` and `bigram` baselines.

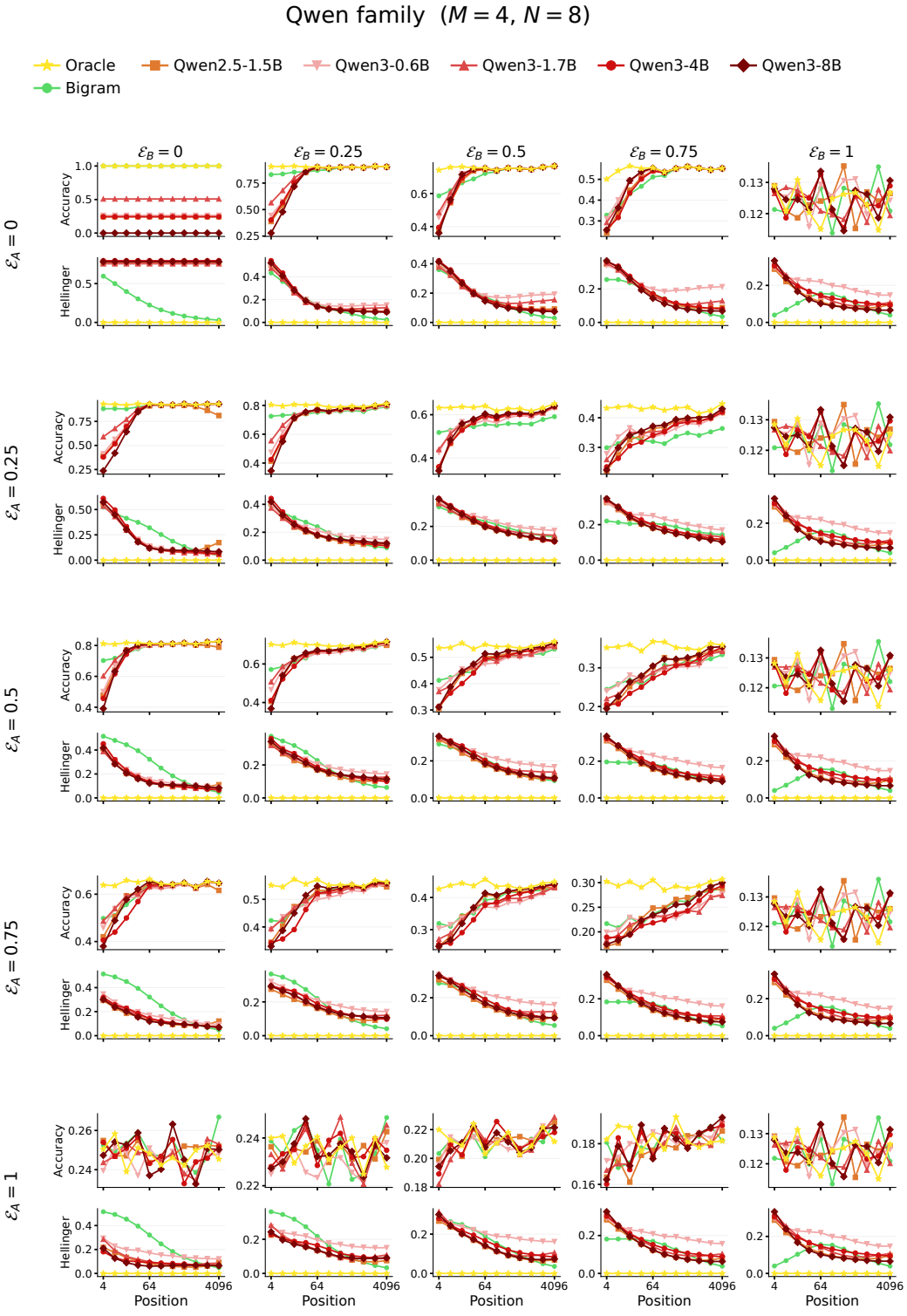


Figure 26: Qwen family LLMs evaluated on HMMs with $M = 4$, $N = 8$. Subplots vary transition entropy $\mathcal{E}(\mathbf{A})$ (rows) and emission entropy $\mathcal{E}(\mathbf{B})$ (columns); each shows top-1 accuracy (upper) and Hellinger-to-oracle distance (lower) versus context length, with `oracle` and `bigram` baselines.

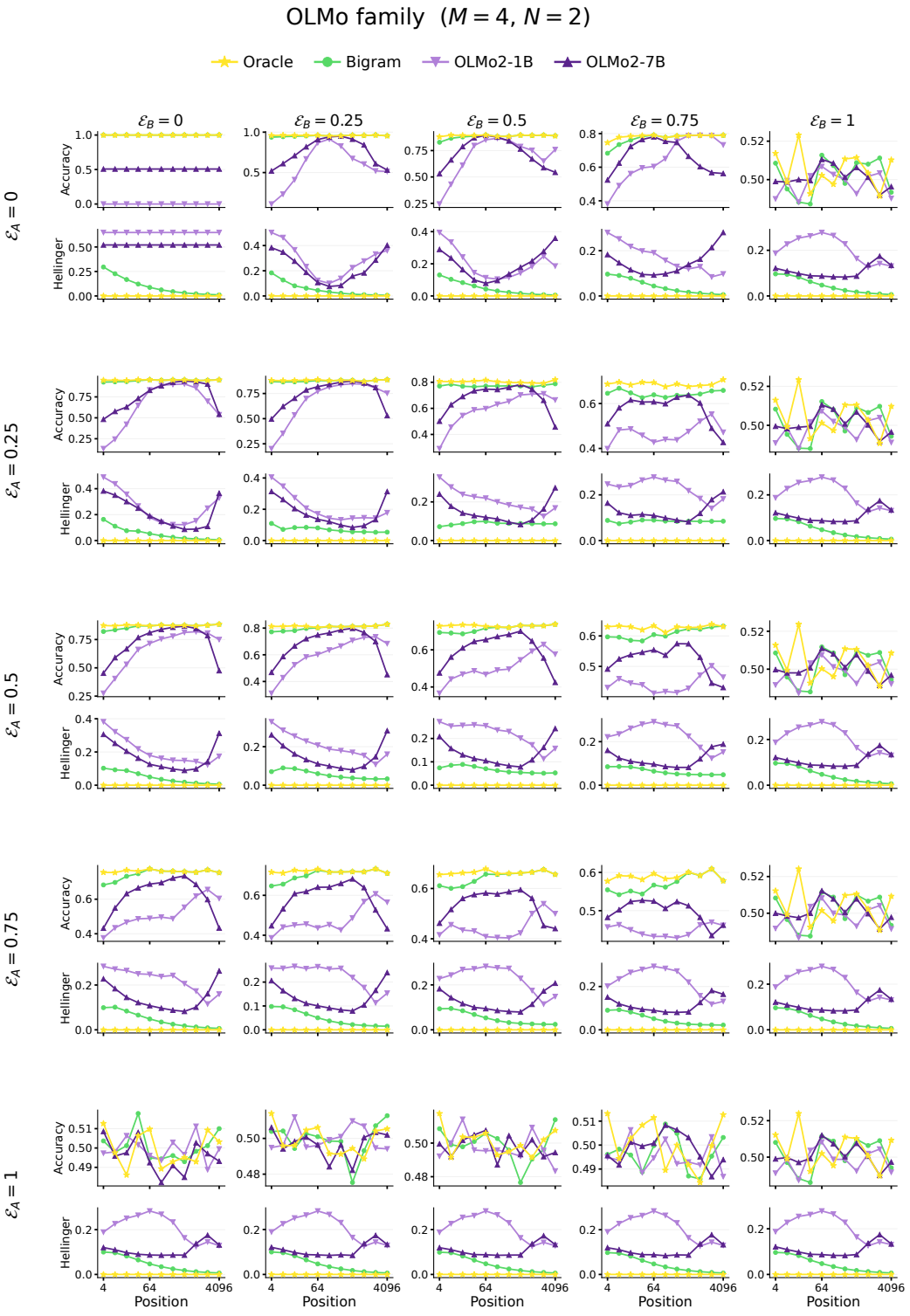


Figure 27: OLMo family LLMs evaluated on HMMs with $M = 4$, $N = 2$. Subplots vary transition entropy $\mathcal{E}(\mathbf{A})$ (rows) and emission entropy $\mathcal{E}(\mathbf{B})$ (columns); each shows top-1 accuracy (upper) and Hellinger-to-oracle distance (lower) versus context length, with `oracle` and `bigram` baselines.

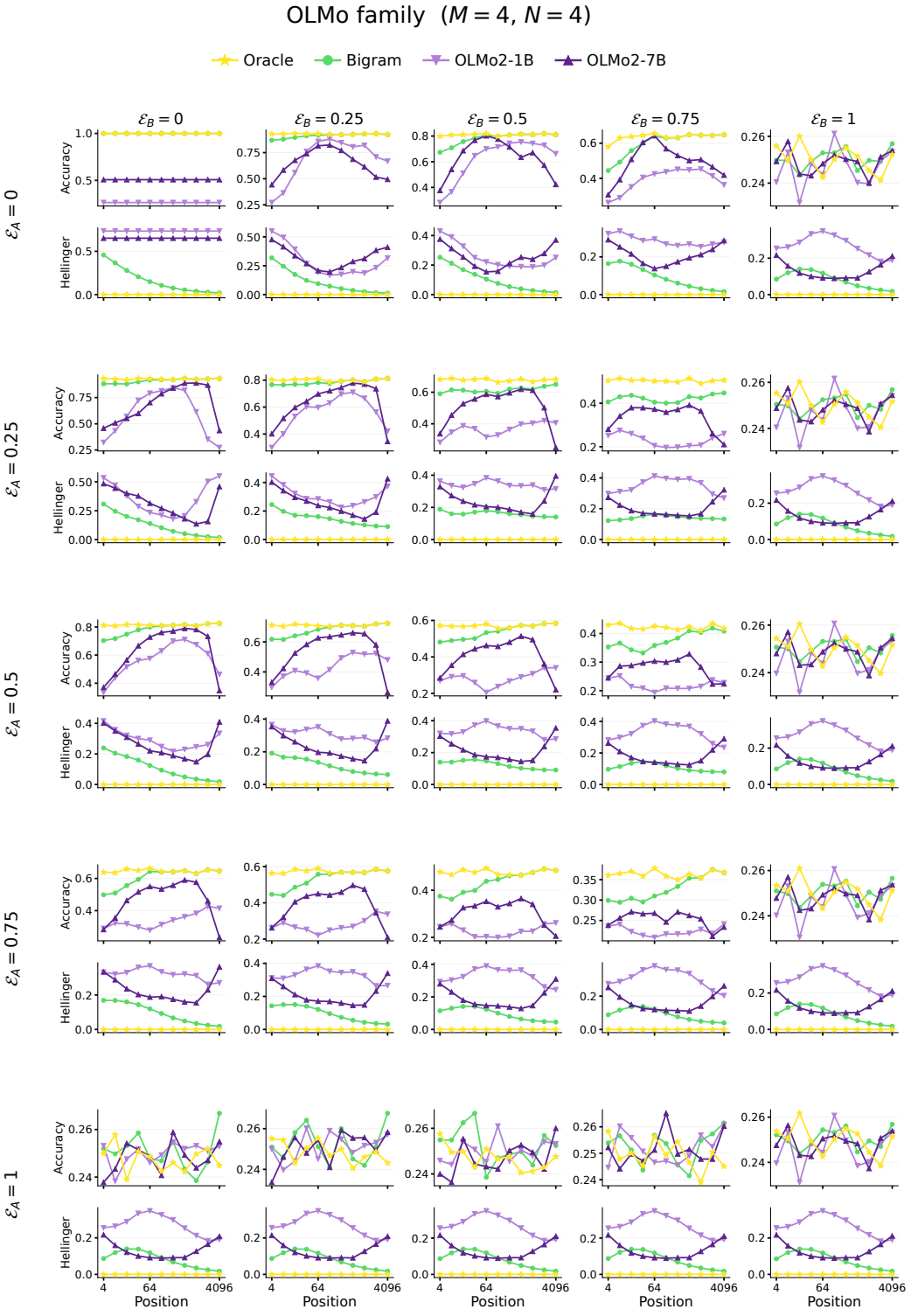


Figure 28: OLMo family LLMs evaluated on HMMs with $M = 4$, $N = 4$. Subplots vary transition entropy $\mathcal{E}(\mathbf{A})$ (rows) and emission entropy $\mathcal{E}(\mathbf{B})$ (columns); each shows top-1 accuracy (upper) and Hellinger-to-oracle distance (lower) versus context length, with `oracle` and `bigram` baselines.

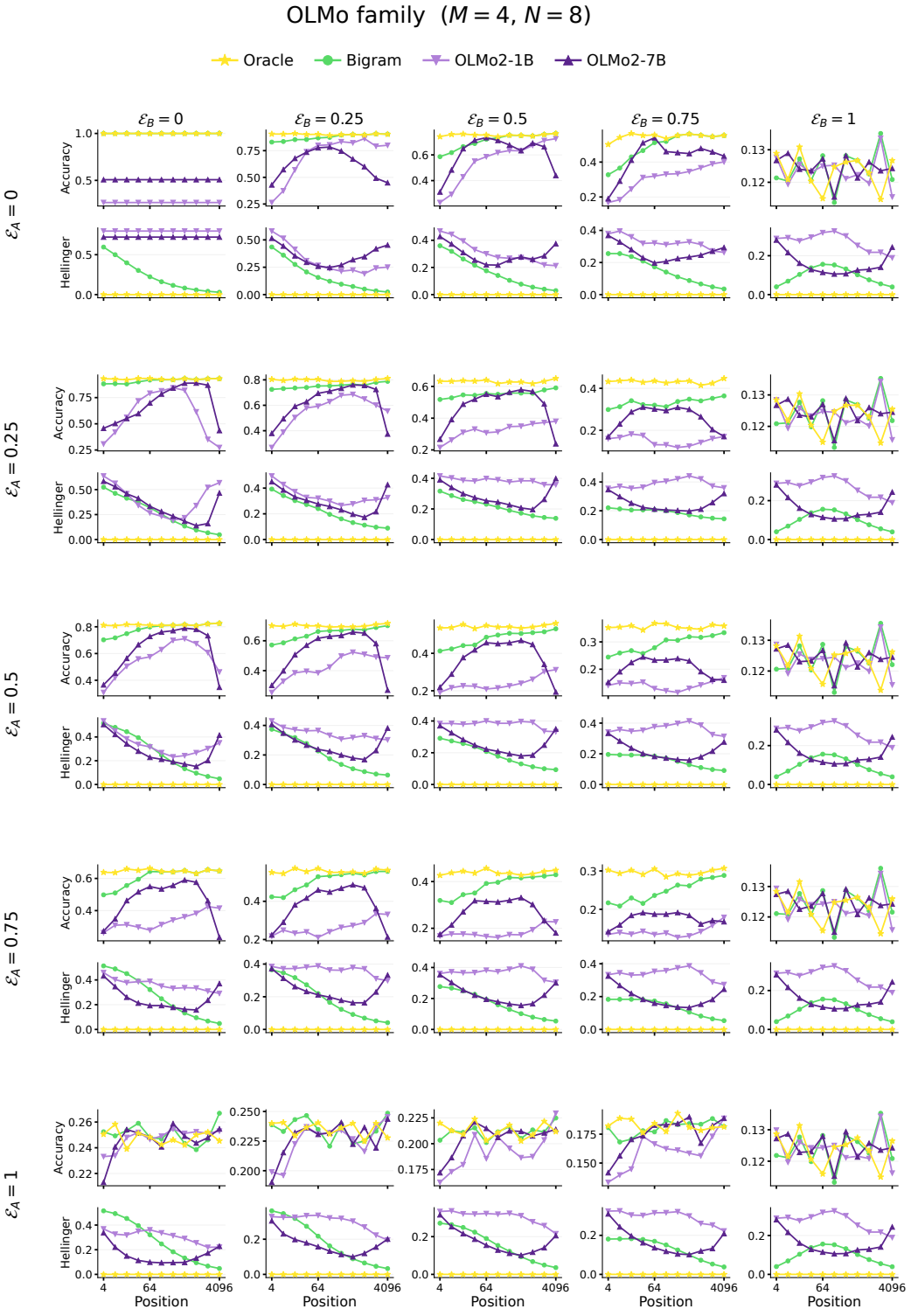


Figure 29: OLMo family LLMs evaluated on HMMs with $M = 4$, $N = 8$. Subplots vary transition entropy $\mathcal{E}(\mathbf{A})$ (rows) and emission entropy $\mathcal{E}(\mathbf{B})$ (columns); each shows top-1 accuracy (upper) and Hellinger-to-oracle distance (lower) versus context length, with `oracle` and `bigram` baselines.

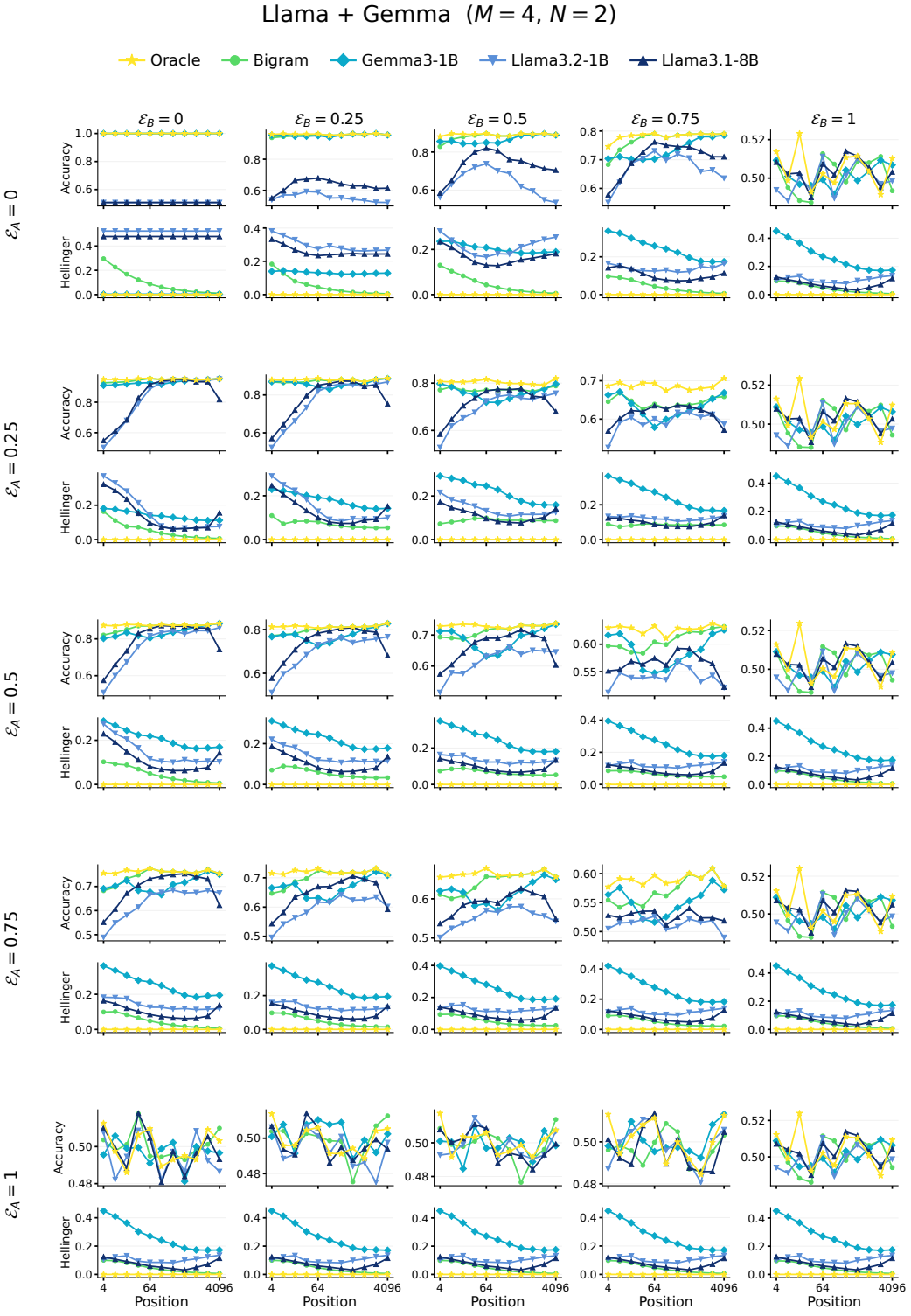


Figure 30: Llama and Gemma family LLMs evaluated on HMMs with $M = 4$, $N = 2$. Subplots vary transition entropy $\mathcal{E}(\mathbf{A})$ (rows) and emission entropy $\mathcal{E}(\mathbf{B})$ (columns); each shows top-1 accuracy (upper) and Hellinger-to-oracle distance (lower) versus context length, with `oracle` and `bigram` baselines.

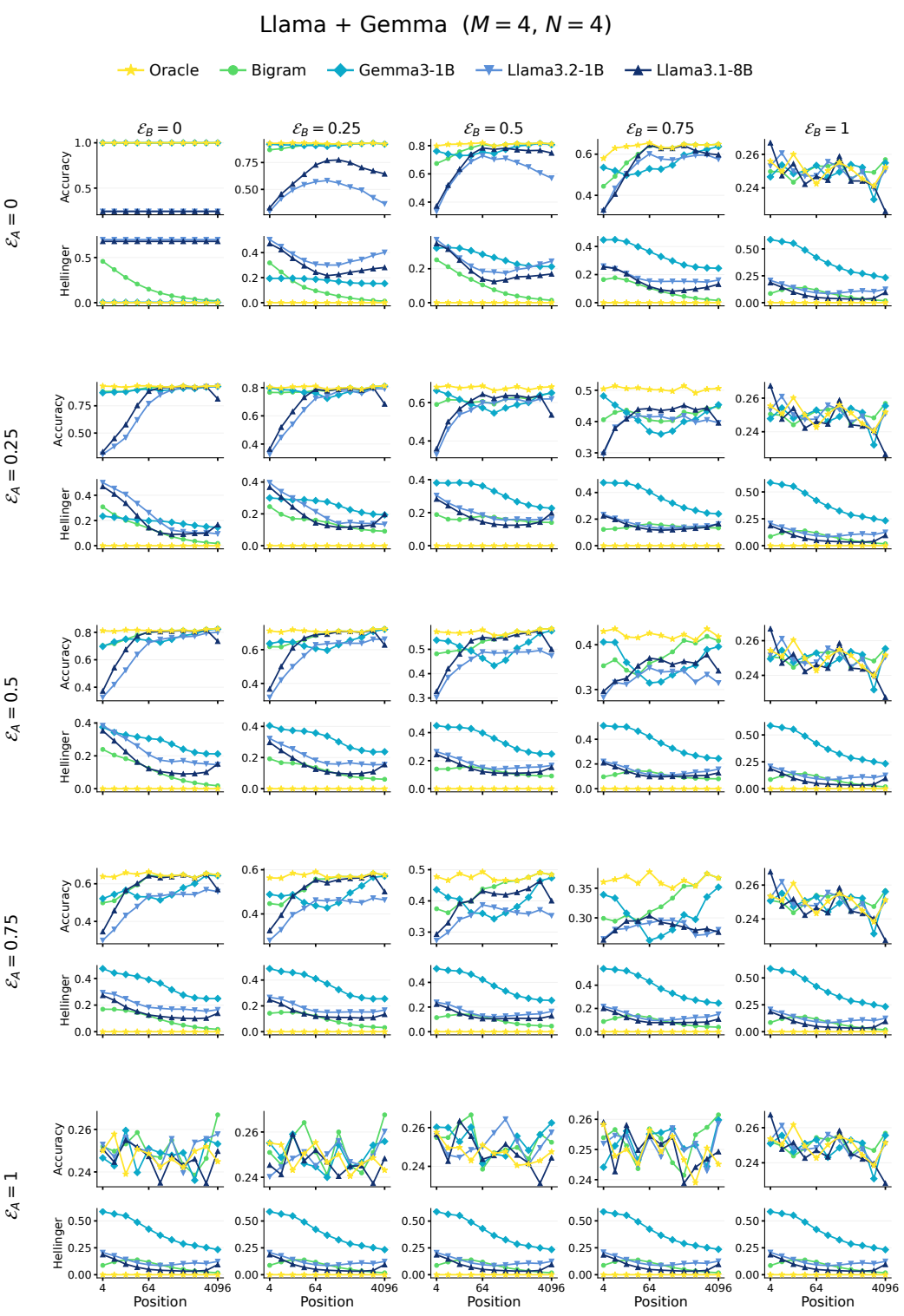


Figure 31: Llama and Gemma family LLMs evaluated on HMMs with $M = 4$, $N = 4$. Subplots vary transition entropy $\mathcal{E}(\mathbf{A})$ (rows) and emission entropy $\mathcal{E}(\mathbf{B})$ (columns); each shows top-1 accuracy (upper) and Hellinger-to-oracle distance (lower) versus context length, with `oracle` and `bigram` baselines.

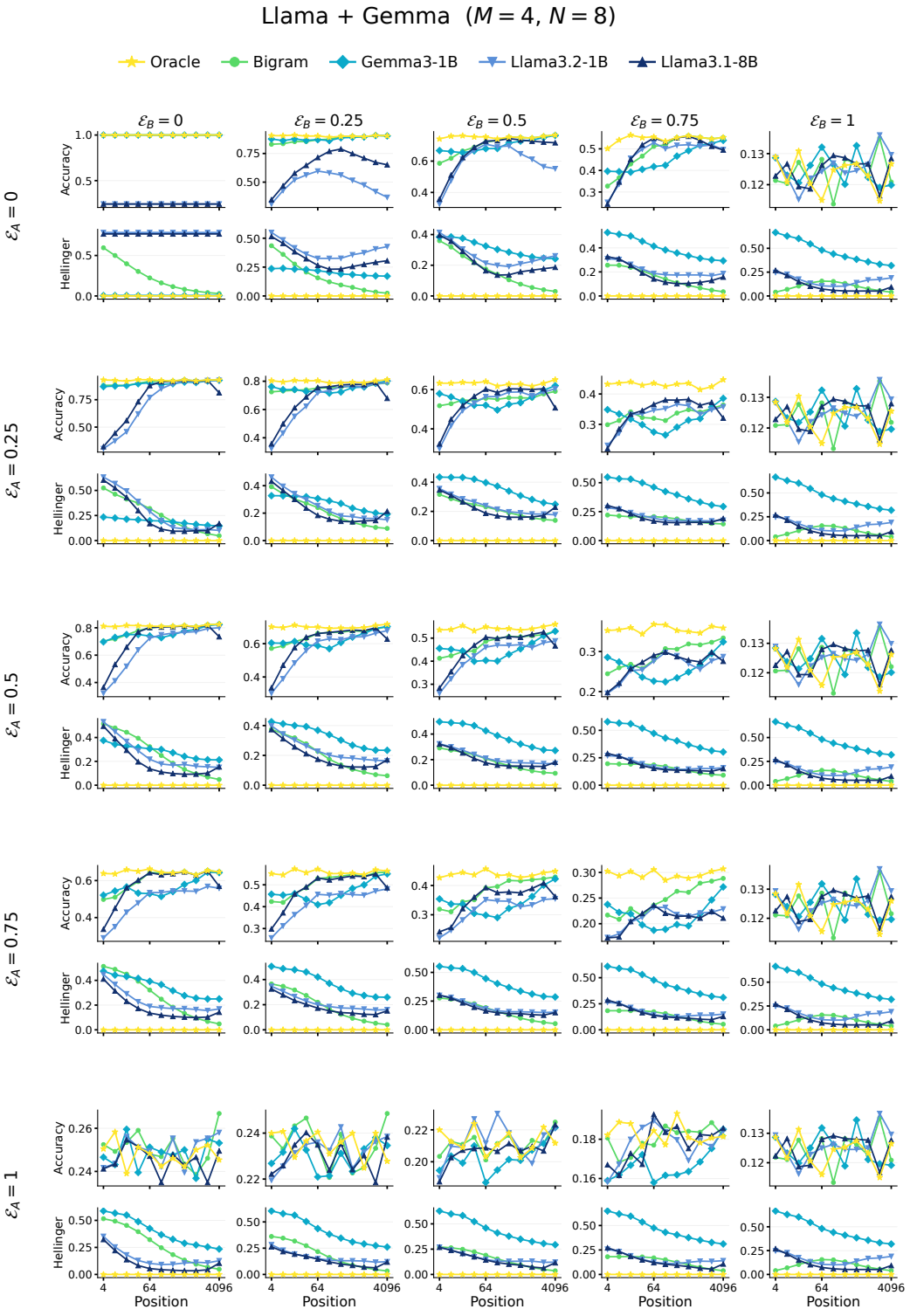


Figure 32: Llama and Gemma family LLMs evaluated on HMMs with $M = 4$, $N = 8$. Subplots vary transition entropy $\mathcal{E}(\mathbf{A})$ (rows) and emission entropy $\mathcal{E}(\mathbf{B})$ (columns); each shows top-1 accuracy (upper) and Hellinger-to-oracle distance (lower) versus context length, with `oracle` and `bigram` baselines.

F.2 Cases When LLMs Fail to Converge to Oracle

We first note an edge case when both transition entropy $\mathcal{E}(\mathbf{A})$ and emission entropy $\mathcal{E}(\mathbf{B})$ are zero. In this setting, the observation sequence reduces to a constant string (e.g., "AAAA…"), which should be trivially easy to predict. Nevertheless, most models fail to continue predicting the repeated token, likely as an artifact of natural language pretraining Yona et al. (2025). We exclude this configuration from subsequent analysis.

More broadly, convergence to the `oracle` can be assessed jointly via top-1 accuracy and Hellinger-to-oracle distance. We report both metrics across the remaining 74 configurations in Figure 33. As discussed earlier, certain model families exhibit systematic suboptimality. Focusing on a strong performer such as Qwen3-8B, we find no clear correlation between suboptimality and entropy, though the Hellinger gap shows a mild increase as the number of observations grows.

F.3 Baseline Comparison

We compare LLM performance against the baseline algorithms described in Section 2.2 and Appendix B, grouping them into the same four classes as in Figure 4. All baselines are tuned via extensive hyperparameter search to ensure fair comparison. Specifically, Baum-Welch is run with 10 random restarts and up to 200 iterations each; the linear and non-linear n-gram models are tuned over the order n and L2 regularization strength; and the spectral algorithms apply smoothing to the estimated bigram and trigram statistics.

The classical Baum-Welch algorithm (Figures 34, 35, 36) performs strongly on this task, but its behavior differs markedly from LLMs. Most notably, it achieves high accuracy even at very short context lengths and maintains stable performance as context grows, whereas LLMs typically require longer context to converge. Under high emission entropy, its accuracy becomes less stable across context lengths. Both observations reflect the iterative nature of Baum-Welch: given the number of hidden states M, it fits HMM parameters directly to the observed sequence, so its performance is governed by parameter estimation quality rather than in-context sequence modeling.

The linear and non-linear n-gram variants (Figures 37, 38, 39, 40, 41, 42) exhibit in-context learning behavior qualitatively similar to LLMs, with accuracy steadily improving over longer contexts. There are, however, notable differences among variants: the linear n-gram model trained with MSE loss fails to converge to the LLM or oracle performance, whereas the cross-entropy and ridge regression variants do.

The spectral algorithms (Figures 43, 44, 45) show a somewhat different profile from the n-gram variants, tending to exhibit a U-shaped accuracy curve with more pronounced improvement at longer context lengths. This pattern is consistent with the fact that reliable estimation of the underlying bigram and trigram statistics requires a larger number of observations, suggesting that the spectral methods may become more competitive as context grows.

Taken together, none of the baseline algorithms can be confidently identified as matching LLM behavior across all HMM configurations. The n-gram variants and spectral methods each resemble LLMs in certain regimes but diverge in others. Baum-Welch, however, can be confidently ruled out: its ability to achieve strong performance from very short contexts reflects a fundamentally different learning mechanism rather than the gradual in-context generalization characteristic of LLMs.

F.4 Soft n-gram Baselines with Varying n

We examine how the order n of the n-gram models affects predictive performance across HMM configurations. Recall that n controls the length of the conditioning history: larger n allows the model to capture longer-range dependencies in the observation sequence, at the cost of requiring more data to reliably estimate the associated statistics. We report results for both linear and non-linear variants across $N \in \{2, 4, 8\}$ to assess how the optimal order interacts with alphabet size and entropy (Figure 46, 47, 47).

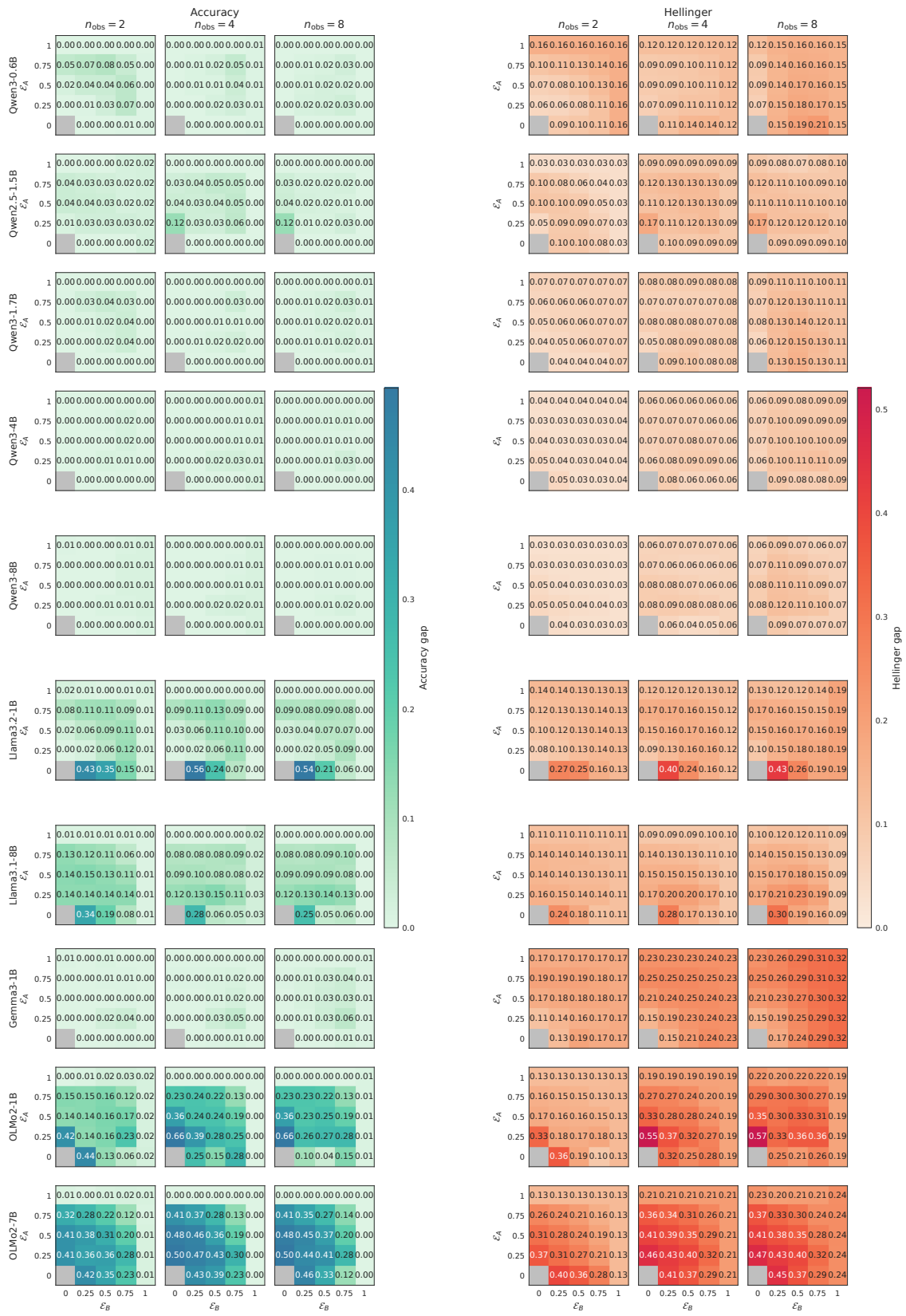


Figure 33: LLM suboptimality vs. the Bayes-optimal `oracle` at position t=4096 (M=4). Per model (rows): accuracy gap (left) and Hellinger distance to the oracle posterior (right); lower is better, 0 = matches oracle. Columns are $n_{\text{obs}} \in \{2, 4, 8\}$; each cell is a 5×5 sweep of transition entropy $\mathcal{E}_A$ (vertical) vs. emission entropy $\mathcal{E}_B$ (horizontal). Darker color means more performance gap.

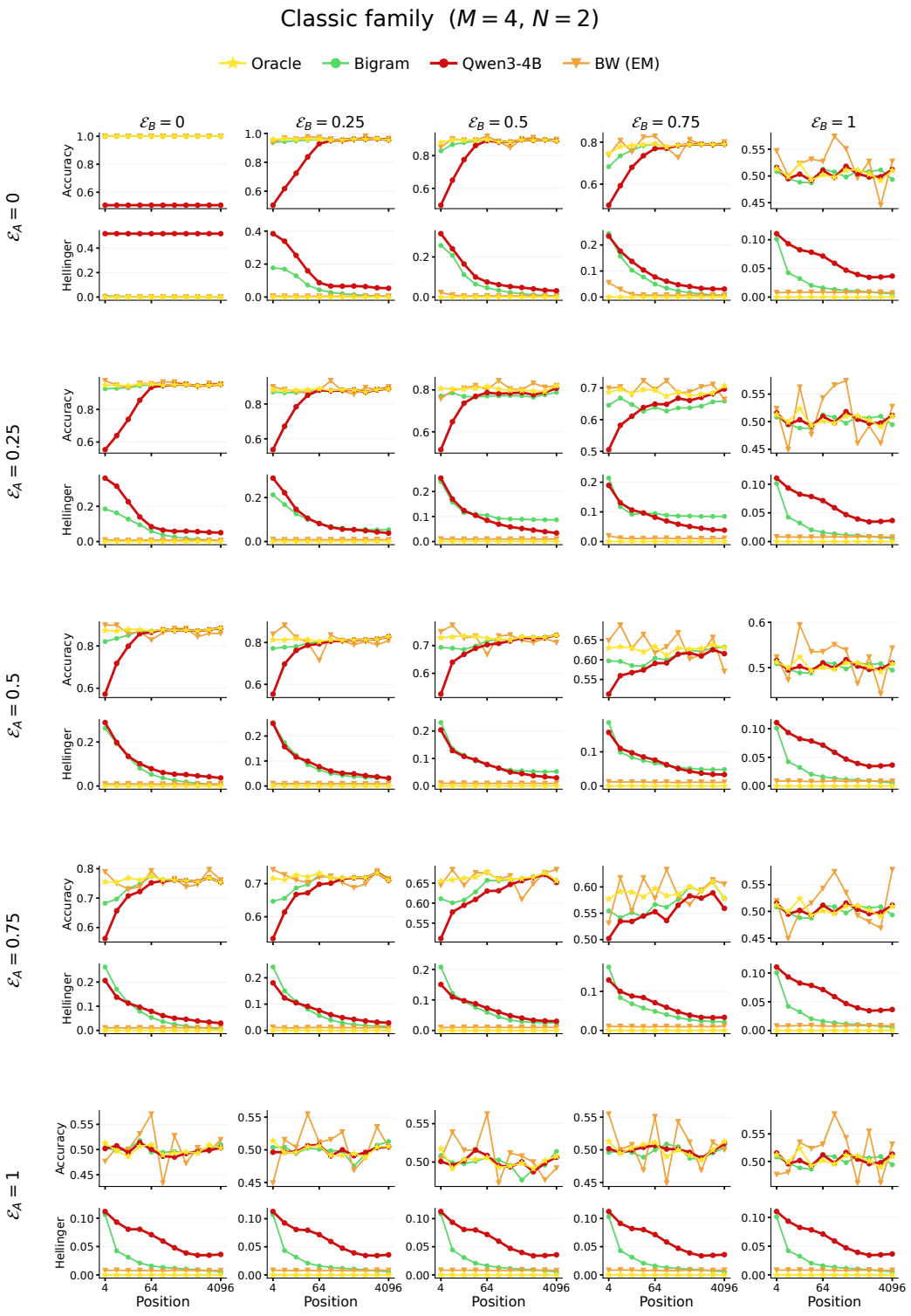


Figure 34: Grid over transition entropy $\mathcal{E}_A$ (rows) and emission entropy $\mathcal{E}_B$ (columns); each cell shows top-1 accuracy (top) and Hellinger distance to the oracle posterior (bottom) as a function of sequence position. Higher accuracy / lower Hellinger is better.

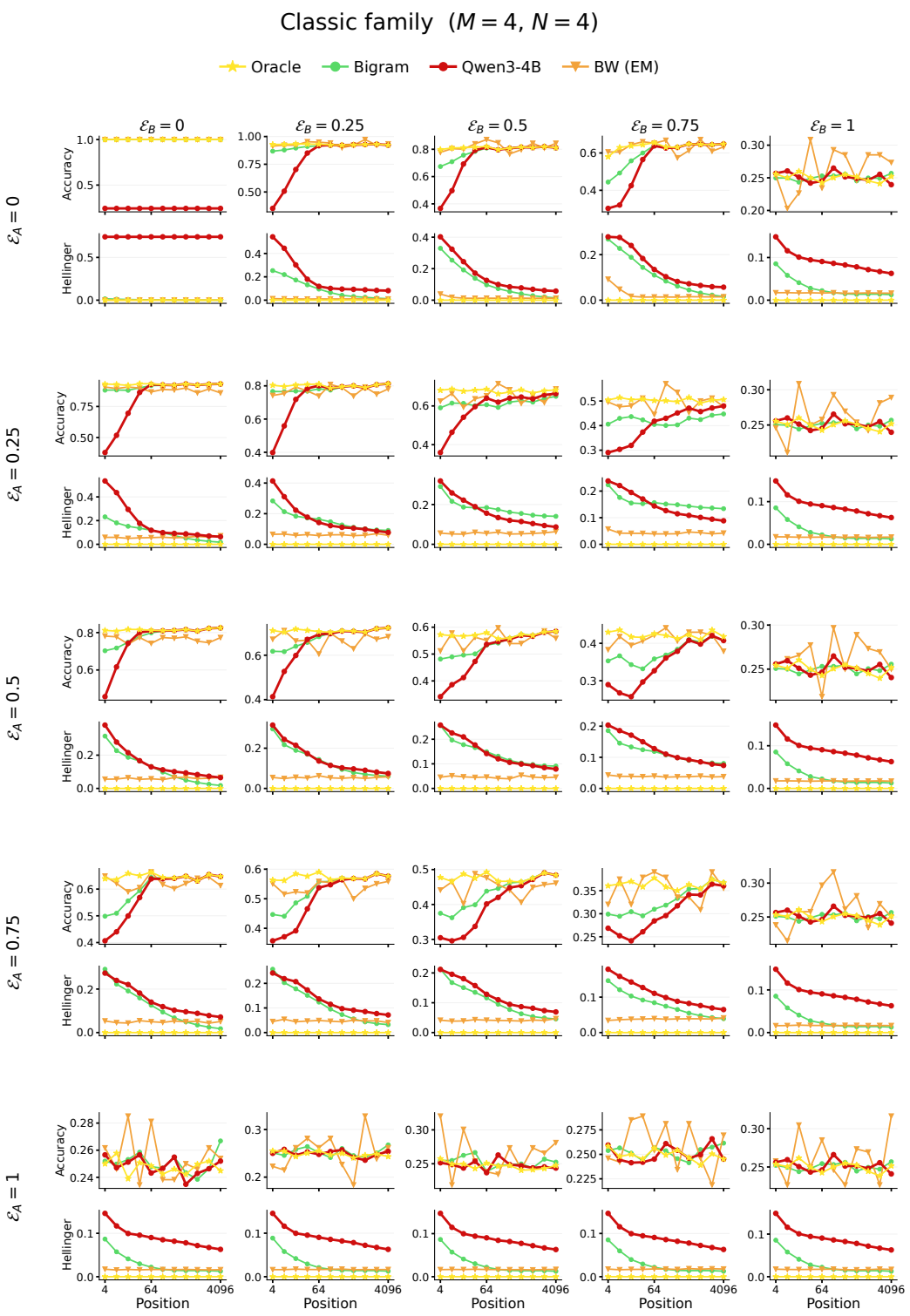


Figure 35: Grid over transition entropy $\mathcal{E}_A$ (rows) and emission entropy $\mathcal{E}_B$ (columns); each cell shows top-1 accuracy (top) and Hellinger distance to the oracle posterior (bottom) as a function of sequence position. Higher accuracy / lower Hellinger is better.

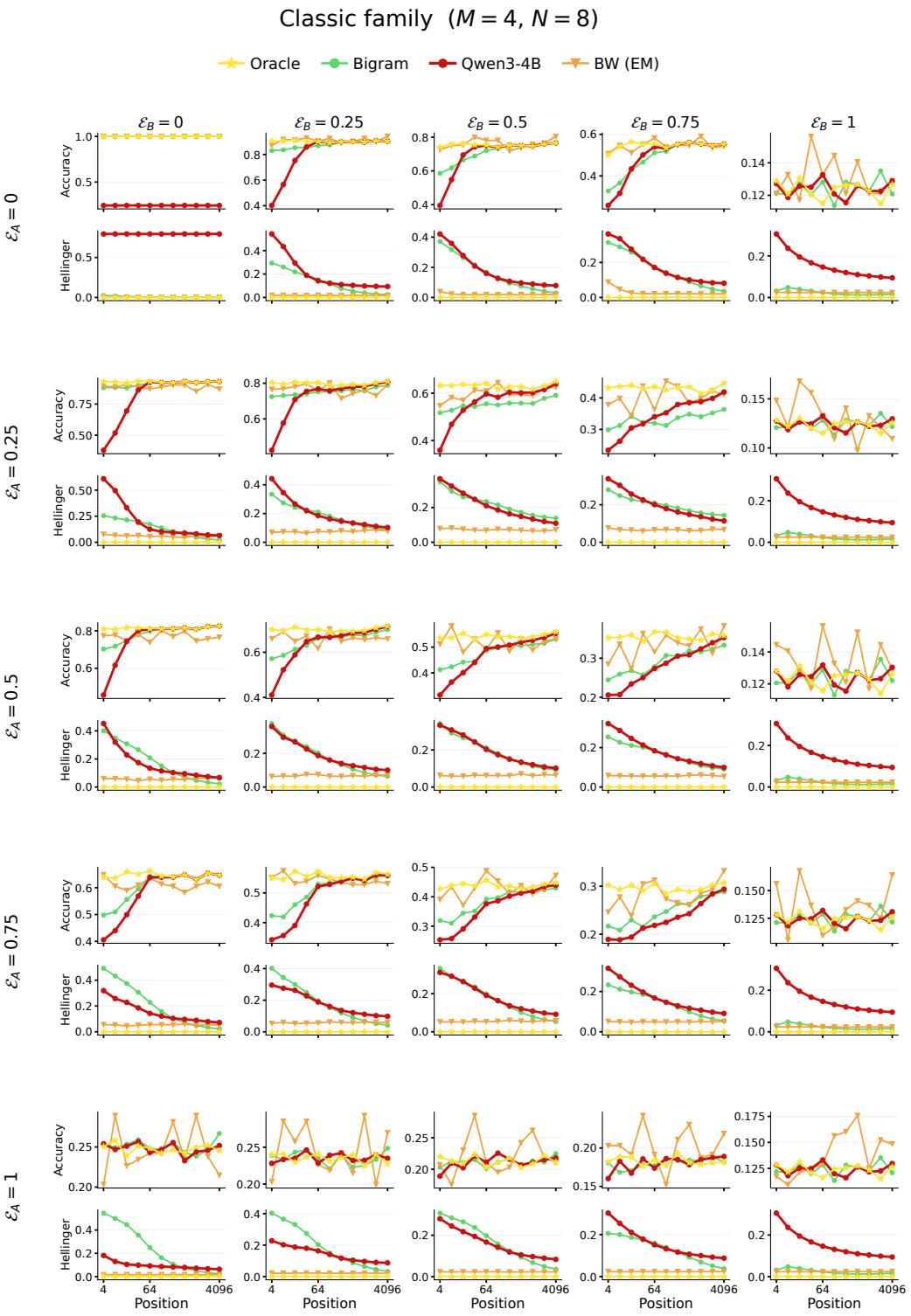


Figure 36: Grid over transition entropy $\mathcal{E}_A$ (rows) and emission entropy $\mathcal{E}_B$ (columns); each cell shows top-1 accuracy (top) and Hellinger distance to the oracle posterior (bottom) as a function of sequence position. Higher accuracy / lower Hellinger is better.

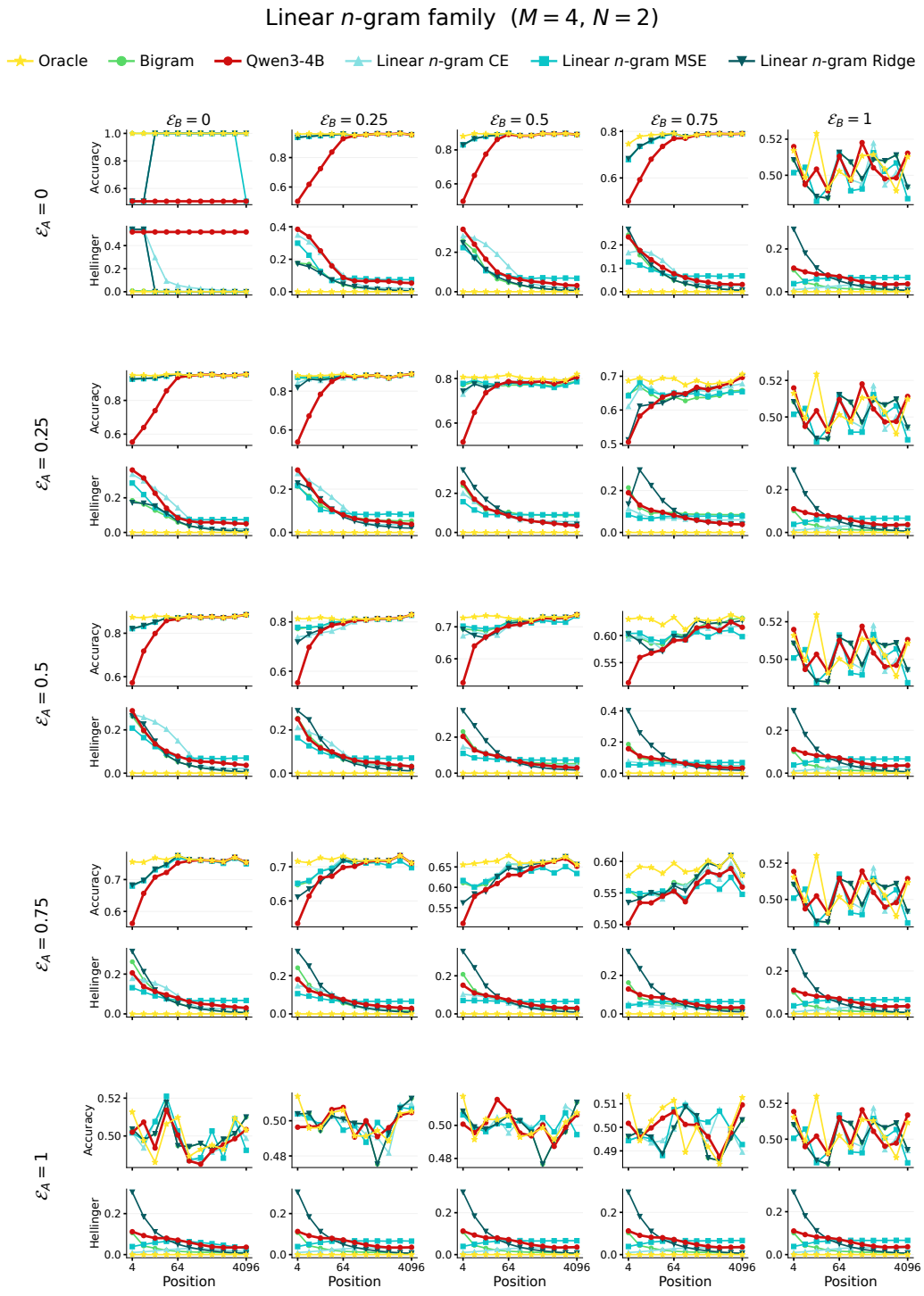


Figure 37: Grid over transition entropy $\mathcal{E}_A$ (rows) and emission entropy $\mathcal{E}_B$ (columns); each cell shows top-1 accuracy (top) and Hellinger distance to the oracle posterior (bottom) as a function of sequence position. Higher accuracy / lower Hellinger is better.

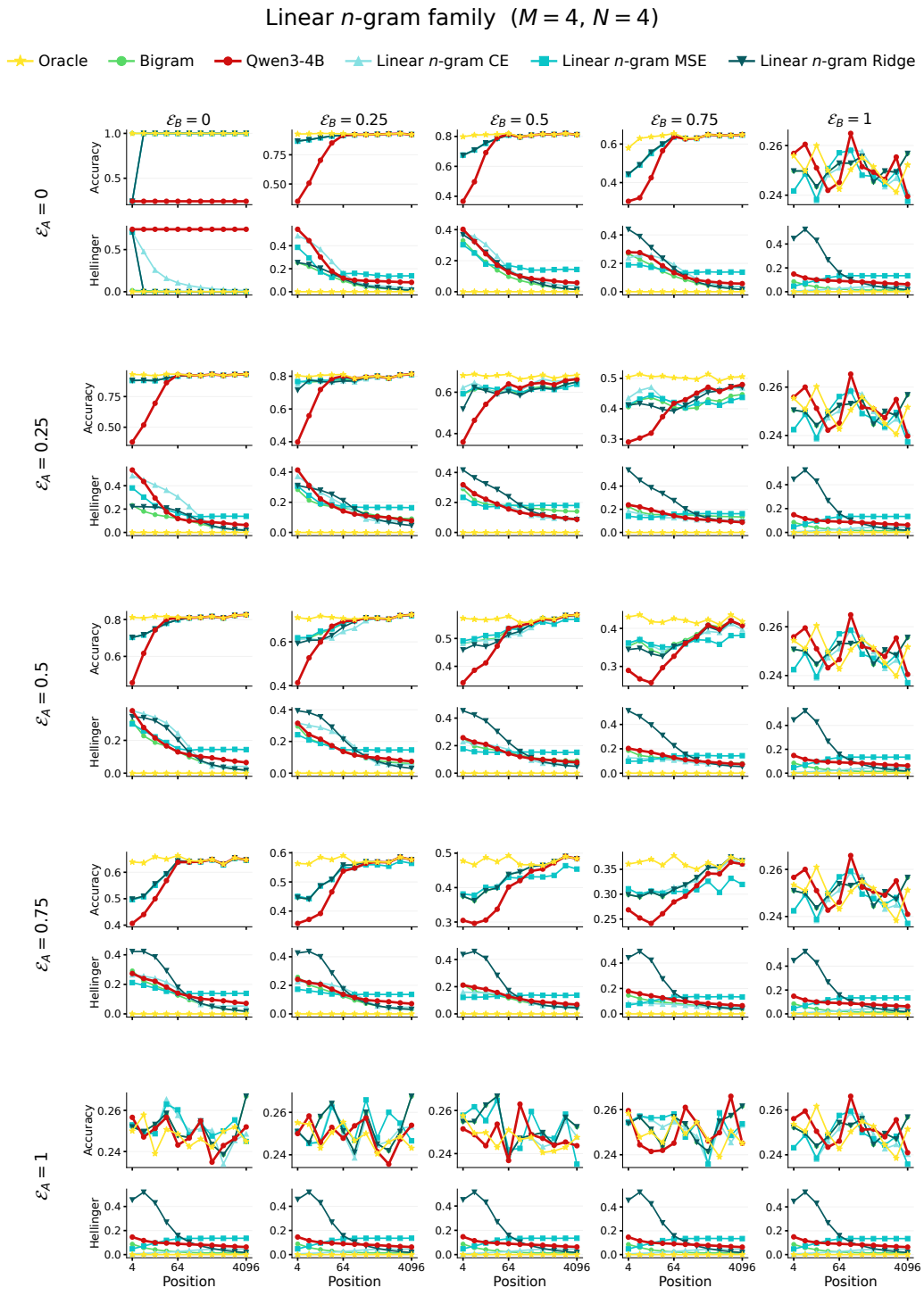


Figure 38: Grid over transition entropy $\mathcal{E}_A$ (rows) and emission entropy $\mathcal{E}_B$ (columns); each cell shows top-1 accuracy (top) and Hellinger distance to the oracle posterior (bottom) as a function of sequence position. Higher accuracy / lower Hellinger is better.

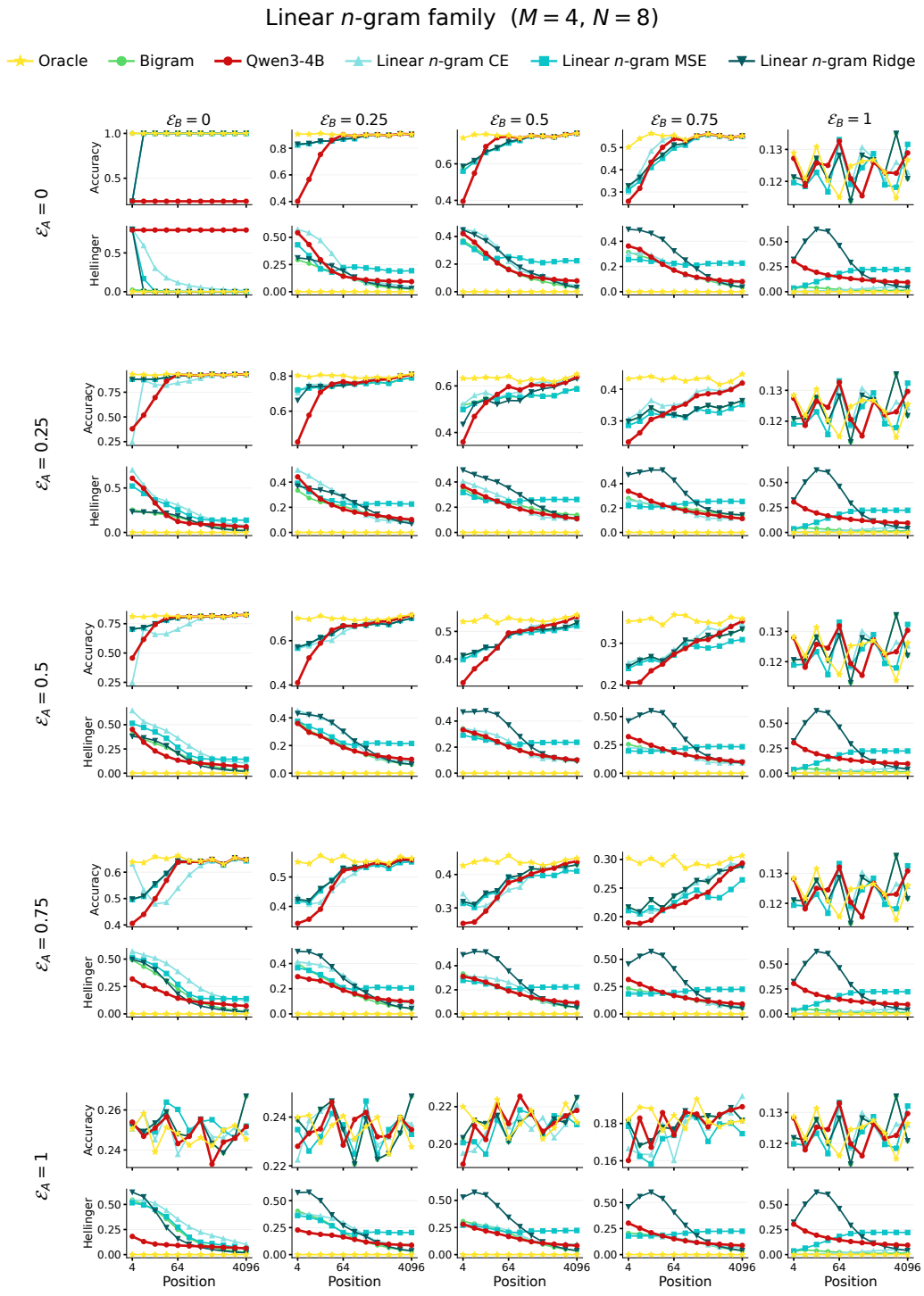


Figure 39: Grid over transition entropy $\mathcal{E}_A$ (rows) and emission entropy $\mathcal{E}_B$ (columns); each cell shows top-1 accuracy (top) and Hellinger distance to the oracle posterior (bottom) as a function of sequence position. Higher accuracy / lower Hellinger is better.

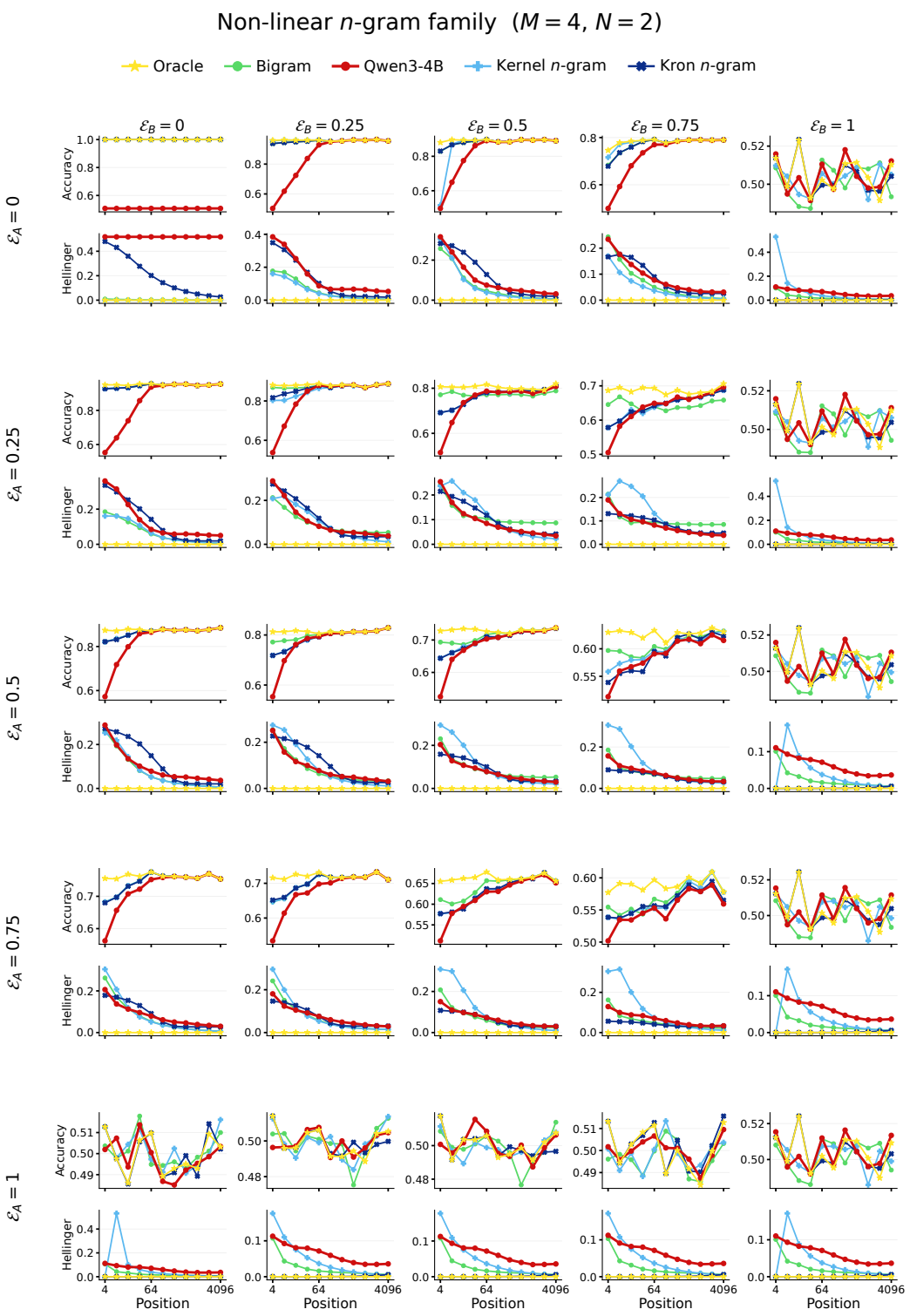


Figure 40: Grid over transition entropy $\mathcal{E}_A$ (rows) and emission entropy $\mathcal{E}_B$ (columns); each cell shows top-1 accuracy (top) and Hellinger distance to the oracle posterior (bottom) as a function of sequence position. Higher accuracy / lower Hellinger is better.

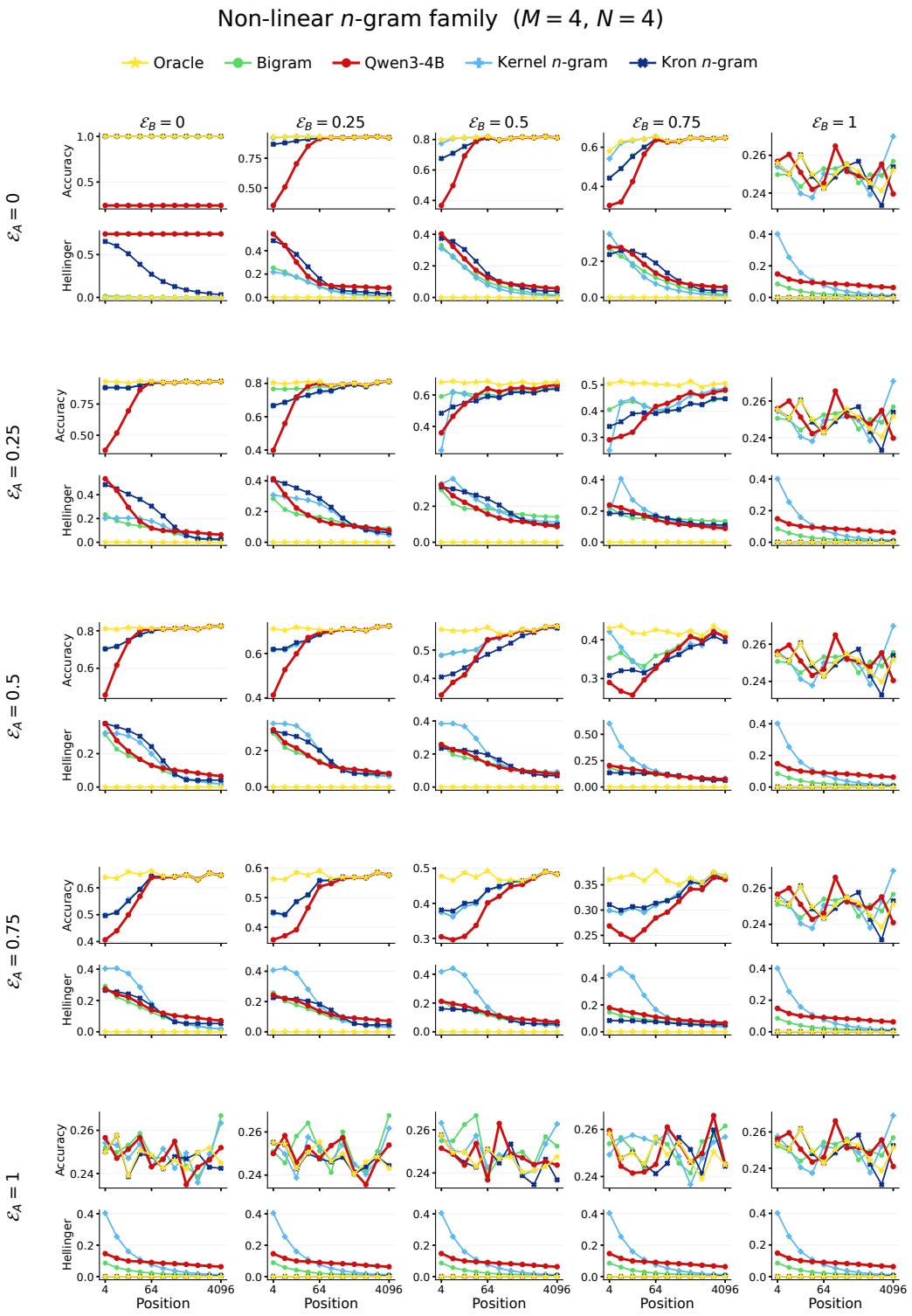


Figure 41: Grid over transition entropy $\mathcal{E}_A$ (rows) and emission entropy $\mathcal{E}_B$ (columns); each cell shows top-1 accuracy (top) and Hellinger distance to the oracle posterior (bottom) as a function of sequence position. Higher accuracy / lower Hellinger is better.

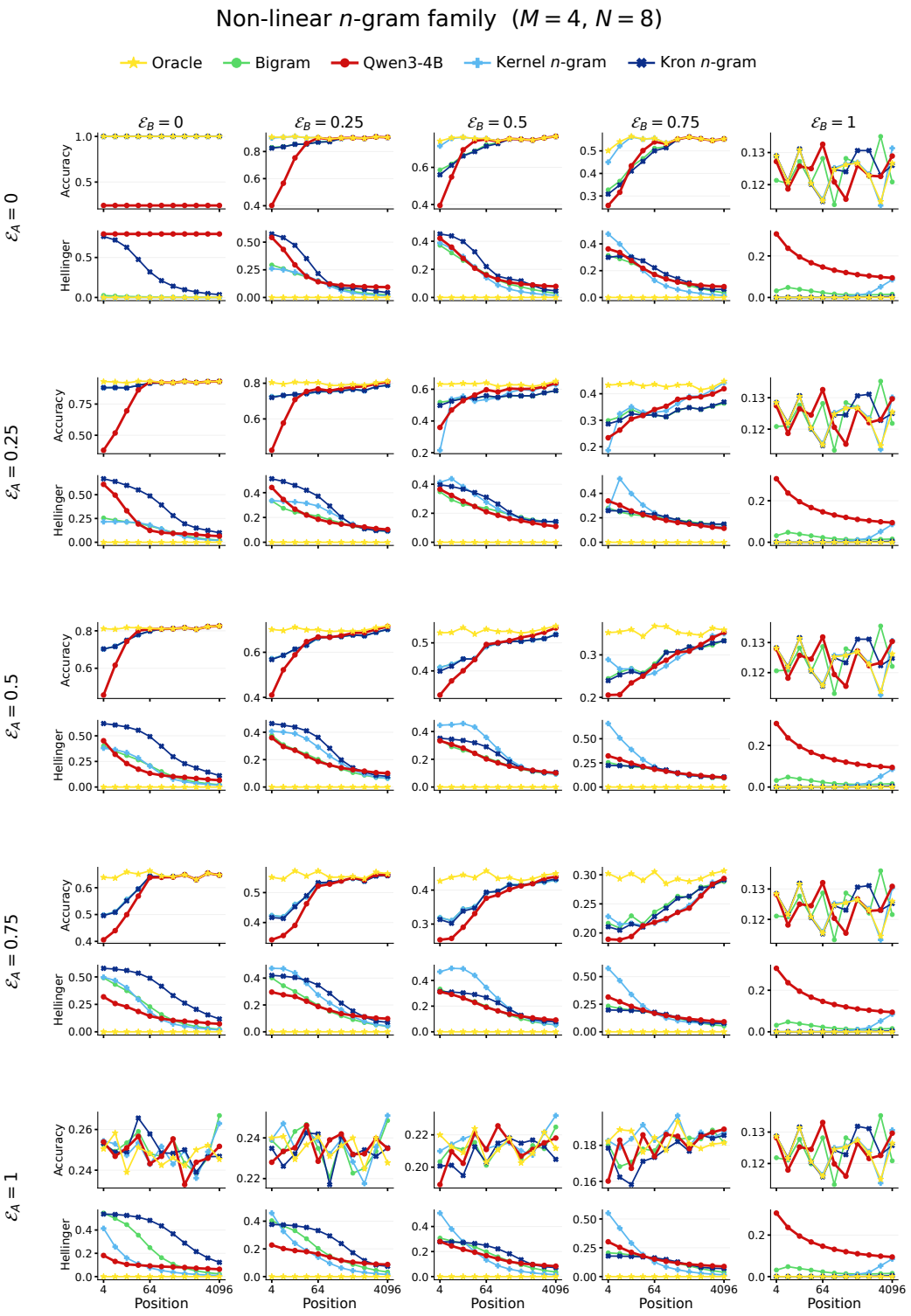


Figure 42: Grid over transition entropy $\mathcal{E}_A$ (rows) and emission entropy $\mathcal{E}_B$ (columns); each cell shows top-1 accuracy (top) and Hellinger distance to the oracle posterior (bottom) as a function of sequence position. Higher accuracy / lower Hellinger is better.

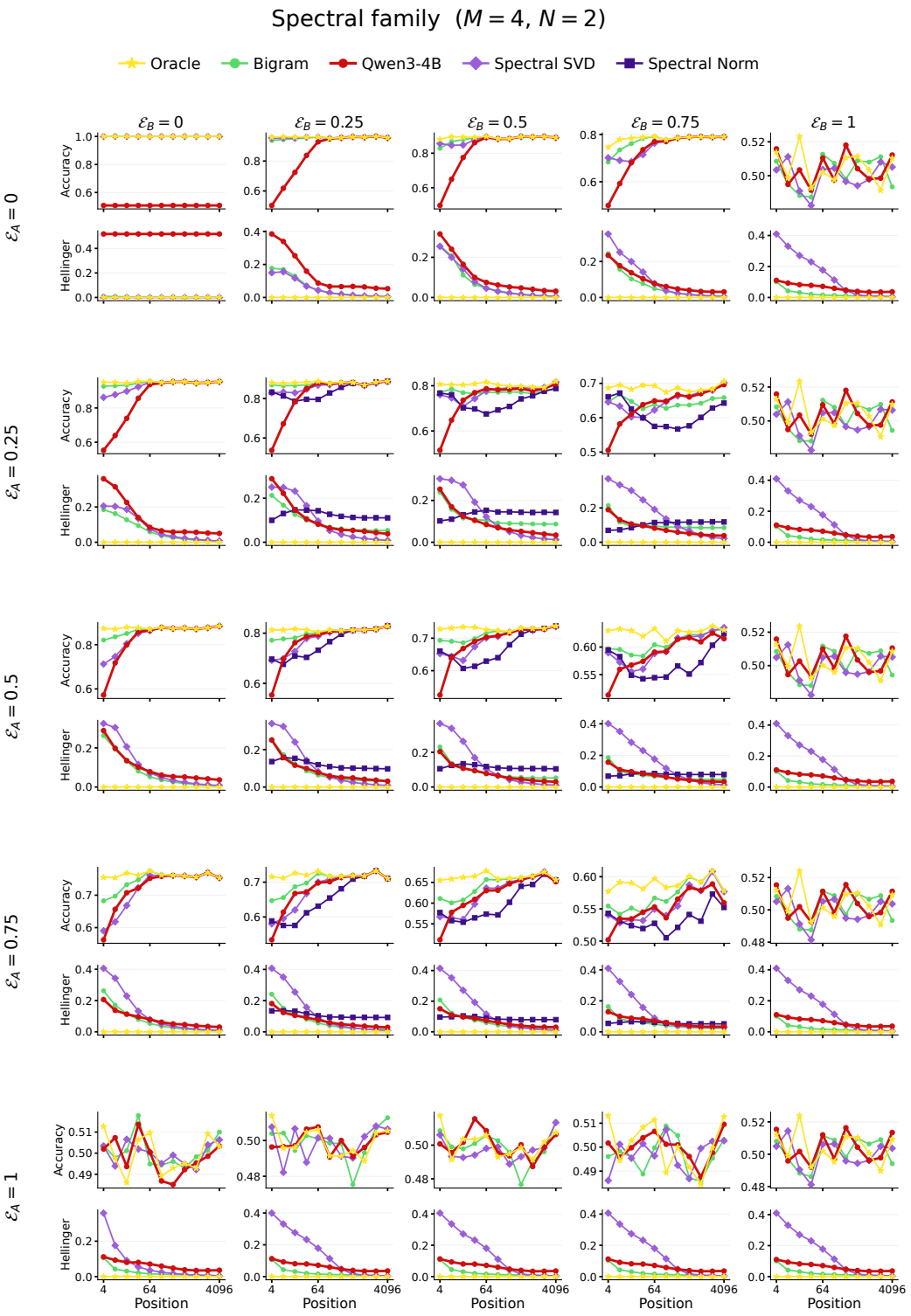


Figure 43: Grid over transition entropy $\mathcal{E}_A$ (rows) and emission entropy $\mathcal{E}_B$ (columns); each cell shows top-1 accuracy (top) and Hellinger distance to the oracle posterior (bottom) as a function of sequence position. Higher accuracy / lower Hellinger is better.

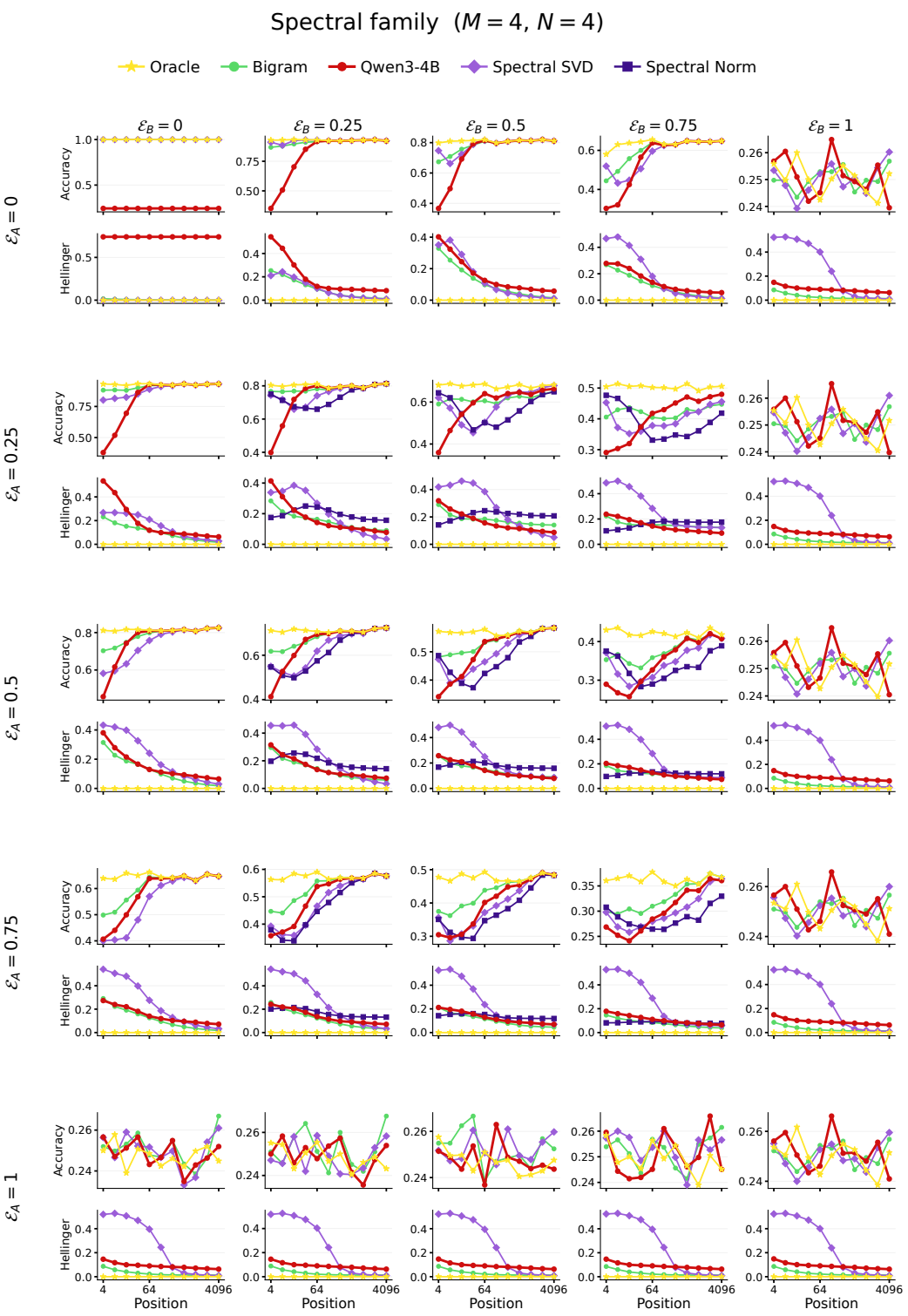


Figure 44: Grid over transition entropy $\mathcal{E}_A$ (rows) and emission entropy $\mathcal{E}_B$ (columns); each cell shows top-1 accuracy (top) and Hellinger distance to the oracle posterior (bottom) as a function of sequence position. Higher accuracy / lower Hellinger is better.

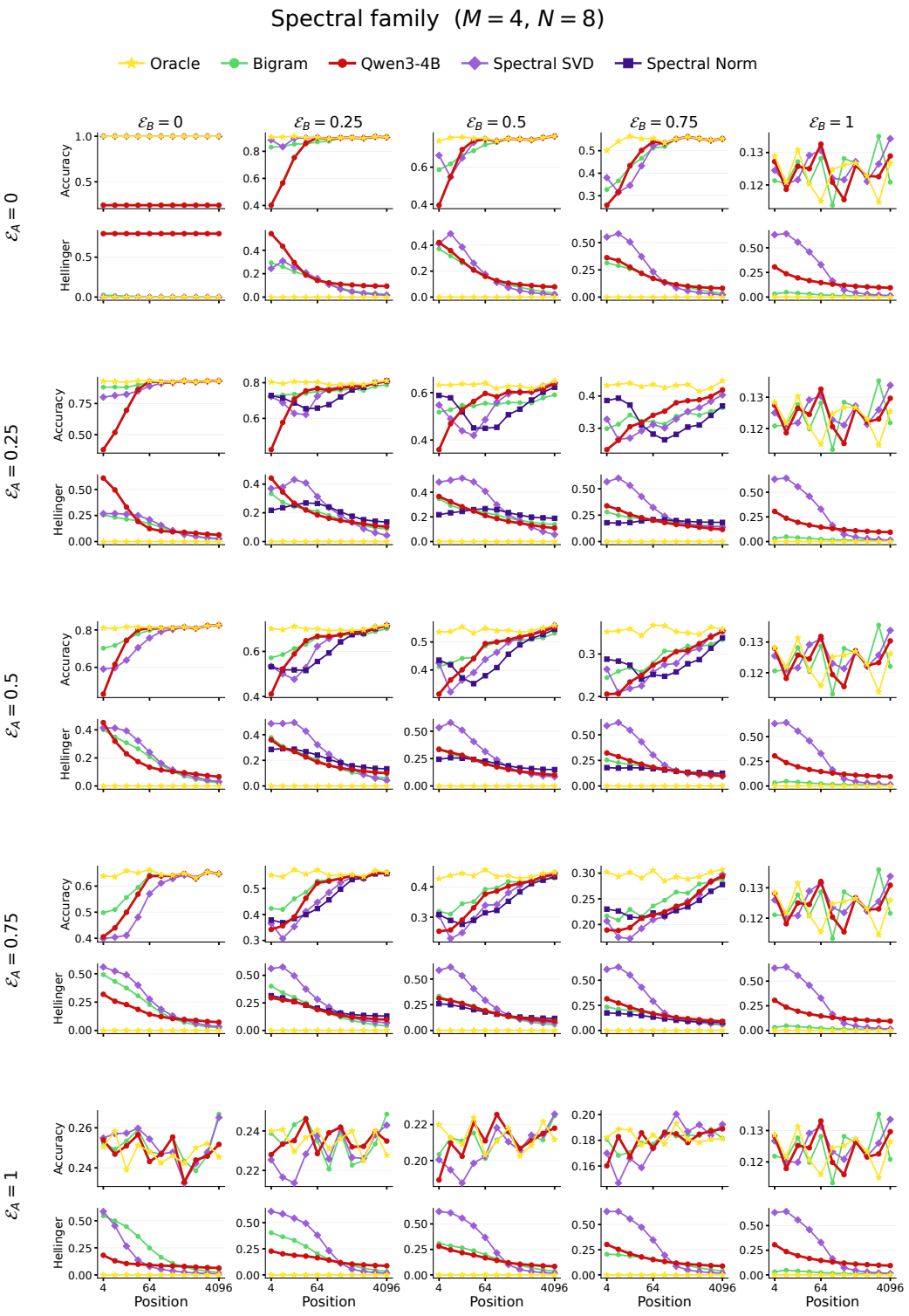


Figure 45: Grid over transition entropy $\mathcal{E}_A$ (rows) and emission entropy $\mathcal{E}_B$ (columns); each cell shows top-1 accuracy (top) and Hellinger distance to the oracle posterior (bottom) as a function of sequence position. Higher accuracy / lower Hellinger is better.

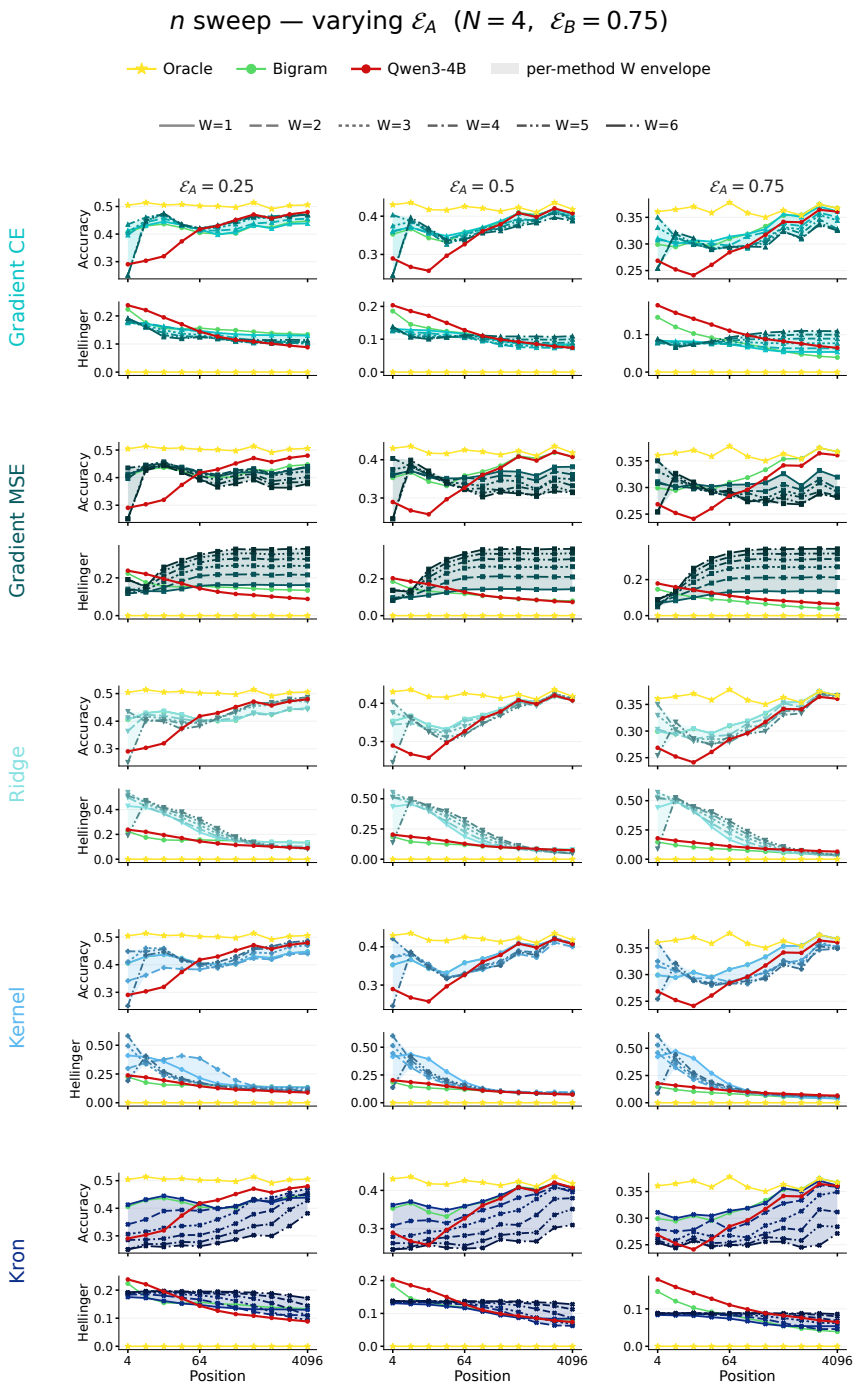


Figure 46: Rows are the windowed baselines (Gradient CE / MSE / Ridge / Kernel / Kron); columns vary the transition entropy $\mathcal{E}_A \in \{0.25, 0.5, 0.75\}$. Each cell shows top-1 accuracy (top) and Hellinger distance to the oracle posterior (bottom) vs. sequence position (log axis). Within a cell, the colored lines are the per-window fits ($n = 1 \ldots 6$, distinguished by linestyle and a light→dark shade) and the shaded band is their min–max envelope; Oracle, Bigram, and Qwen3-4B are drawn for reference.

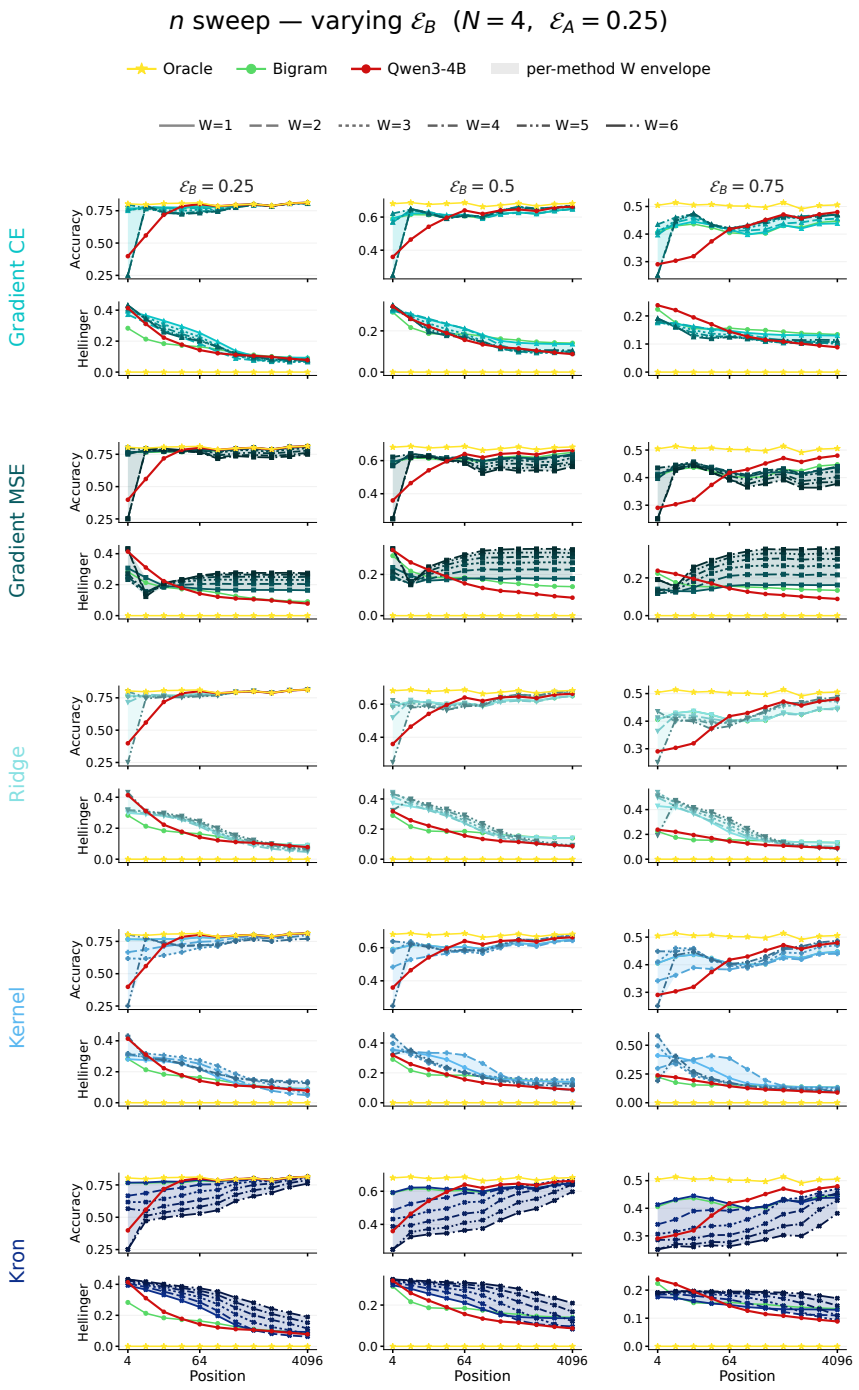


Figure 47: Rows are the windowed baselines (Gradient CE / MSE / Ridge / Kernel / Kron); columns vary the emission entropy $\mathcal{E}_B \in \{0.25, 0.5, 0.75\}$. Each cell shows top-1 accuracy (top) and Hellinger distance to the oracle posterior (bottom) vs. sequence position (log axis). Within a cell, the colored lines are the per-window fits ($n = 1 \ldots 6$, distinguished by linestyle and a light→dark shade) and the shaded band is their min–max envelope; Oracle, Bigram, and Qwen3-4B are drawn for reference.

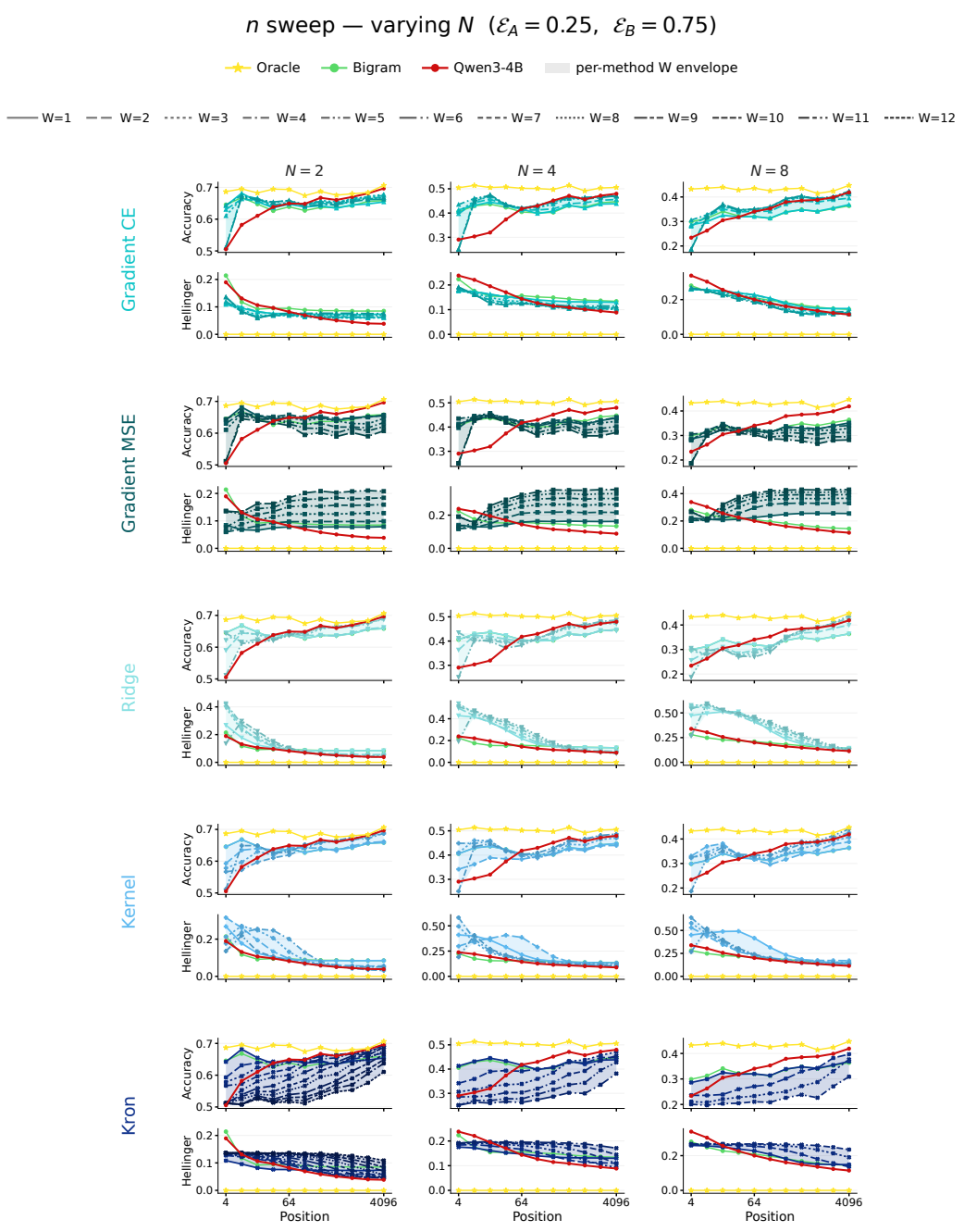


Figure 48: Rows are the windowed baselines (Gradient CE / MSE / Ridge / Kernel / Kron); columns vary the number of observations $N \in \{2, 4, 8\}$. Each cell shows top-1 accuracy (top) and Hellinger distance to the oracle posterior (bottom) vs. sequence position (log axis). Within a cell, the colored lines are the per-window fits ($n = 1 \dots 6$, distinguished by linestyle and a light→dark shade) and the shaded band is their min–max envelope; Oracle, Bigram, and Qwen3-4B are drawn for reference.

F.5 More Hidden States ($M = 12$)

For all the results we show above, they are done with a fixed number of hidden states $M = 4$. We also compare LLM performance on a more complex HMM with 12 hidden states, to see if the insights we gain above is extensible. We find that the convergence trend is generally consistent across these two M's (Figure 49, 50, 51, 52).

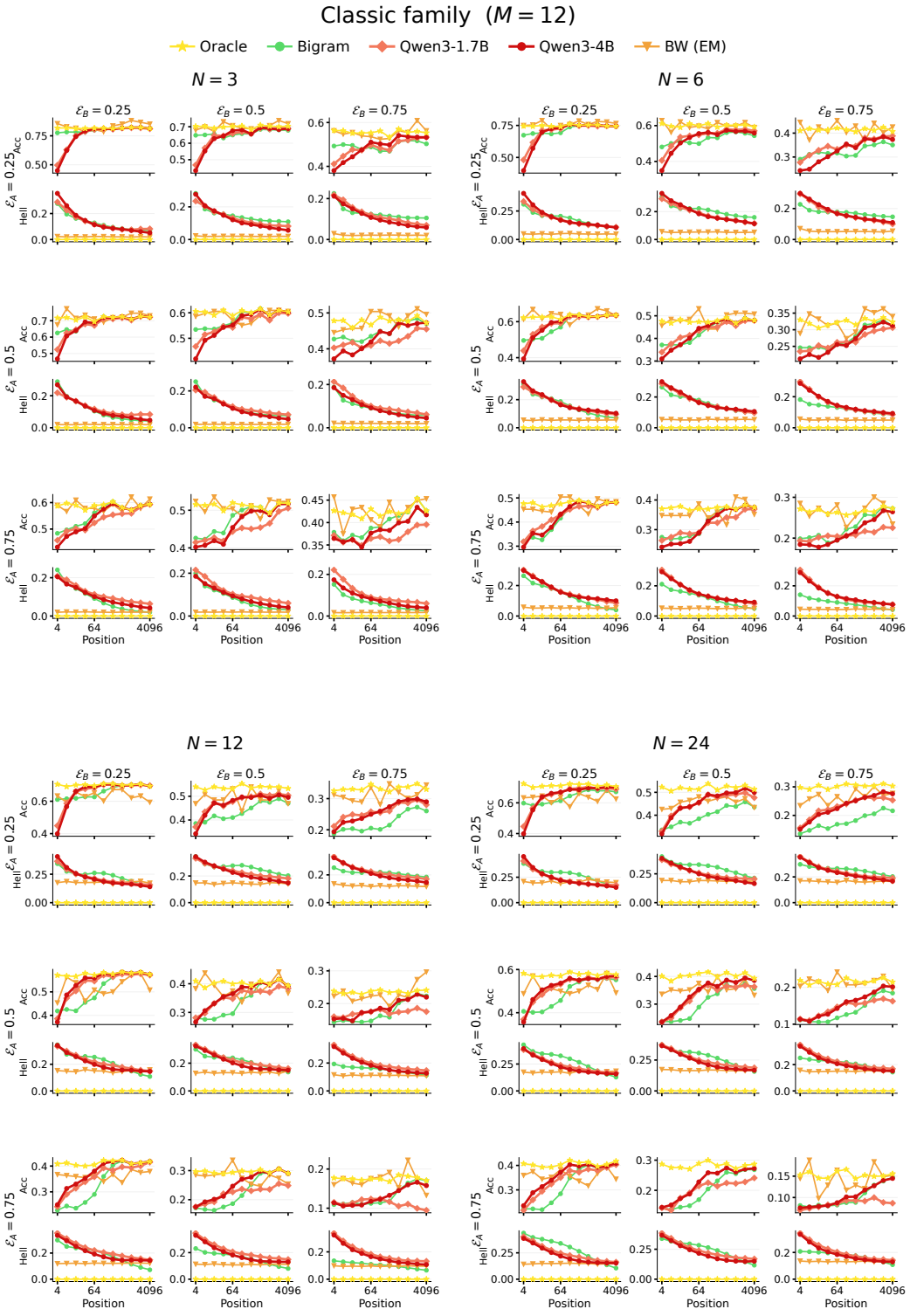


Figure 49: Classic baselines (Bigram, Baum–Welch) vs. position on 12-state HMMs ($M = 12$). Each of the four blocks is a fixed observation-alphabet size $N \in \{3, 6, 12, 24\}$; within a block, rows vary the transition entropy $\mathcal{E}_A$ and columns the emission entropy $\mathcal{E}_B$ (both $\in \{0.25, 0.5, 0.75\}$). Every cell shows top-1 accuracy (top) and Hellinger distance to the oracle posterior (bottom) versus sequence position (log axis). Oracle, Bigram, Qwen3-1.7B and Qwen3-4B are drawn for reference alongside Baum–Welch (BW/EM). Higher accuracy / lower Hellinger is better.

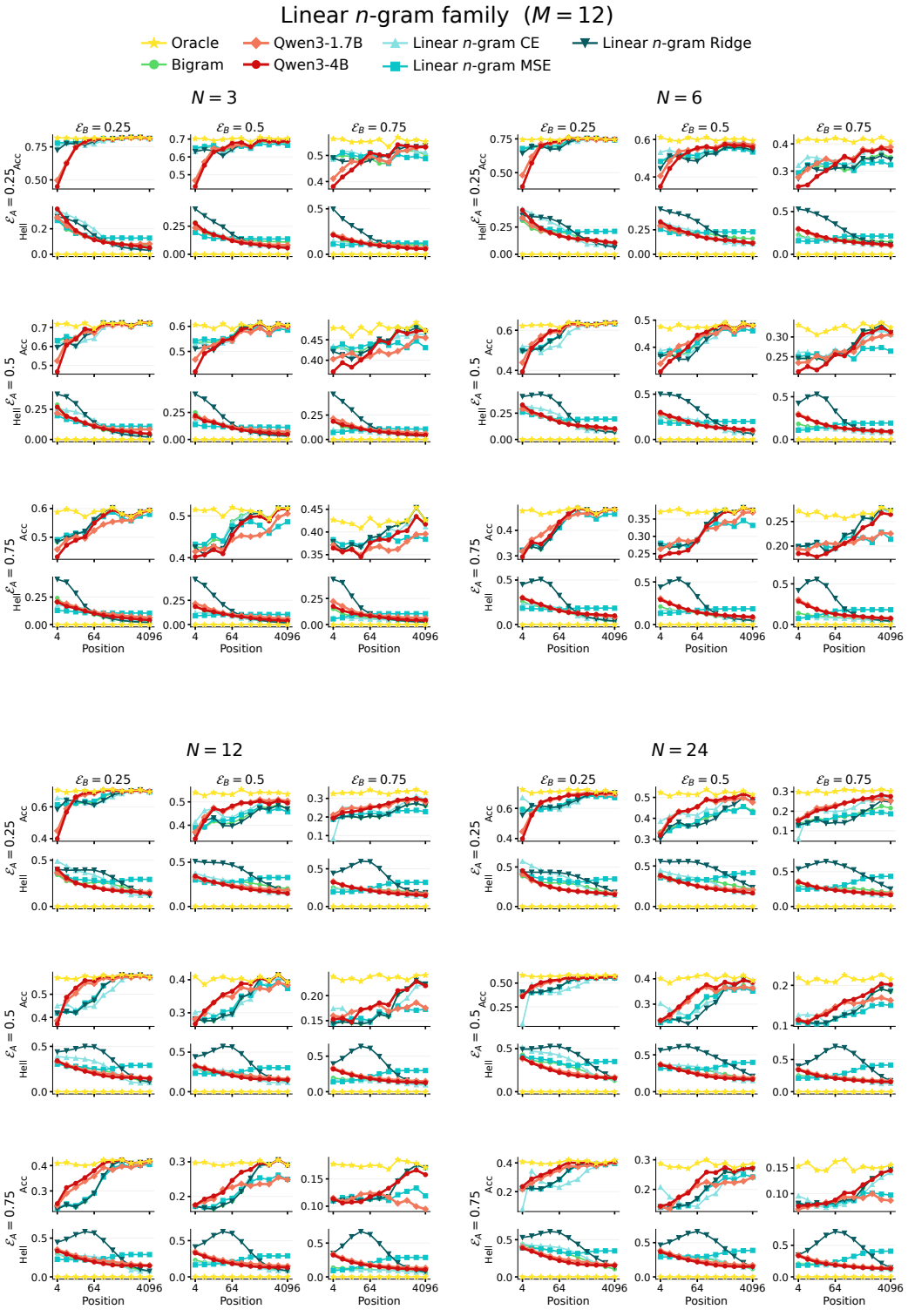


Figure 50: Linear n-gram family vs. position on 12-state HMMs ($M = 12$). Each of the four blocks is a fixed observation-alphabet size $N \in \{3, 6, 12, 24\}$; within a block, rows vary the transition entropy $\mathcal{E}_A$ and columns the emission entropy $\mathcal{E}_B$ (both $\in \{0.25, 0.5, 0.75\}$). Every cell shows top-1 accuracy (top) and Hellinger distance to the oracle posterior (bottom) versus sequence position (log axis). Oracle, Bigram, Qwen3-1.7B and Qwen3-4B are drawn for reference alongside Linear n-gram CE / MSE / Ridge; the n-gram window is selected per position by validation. Higher accuracy / lower Hellinger is better.

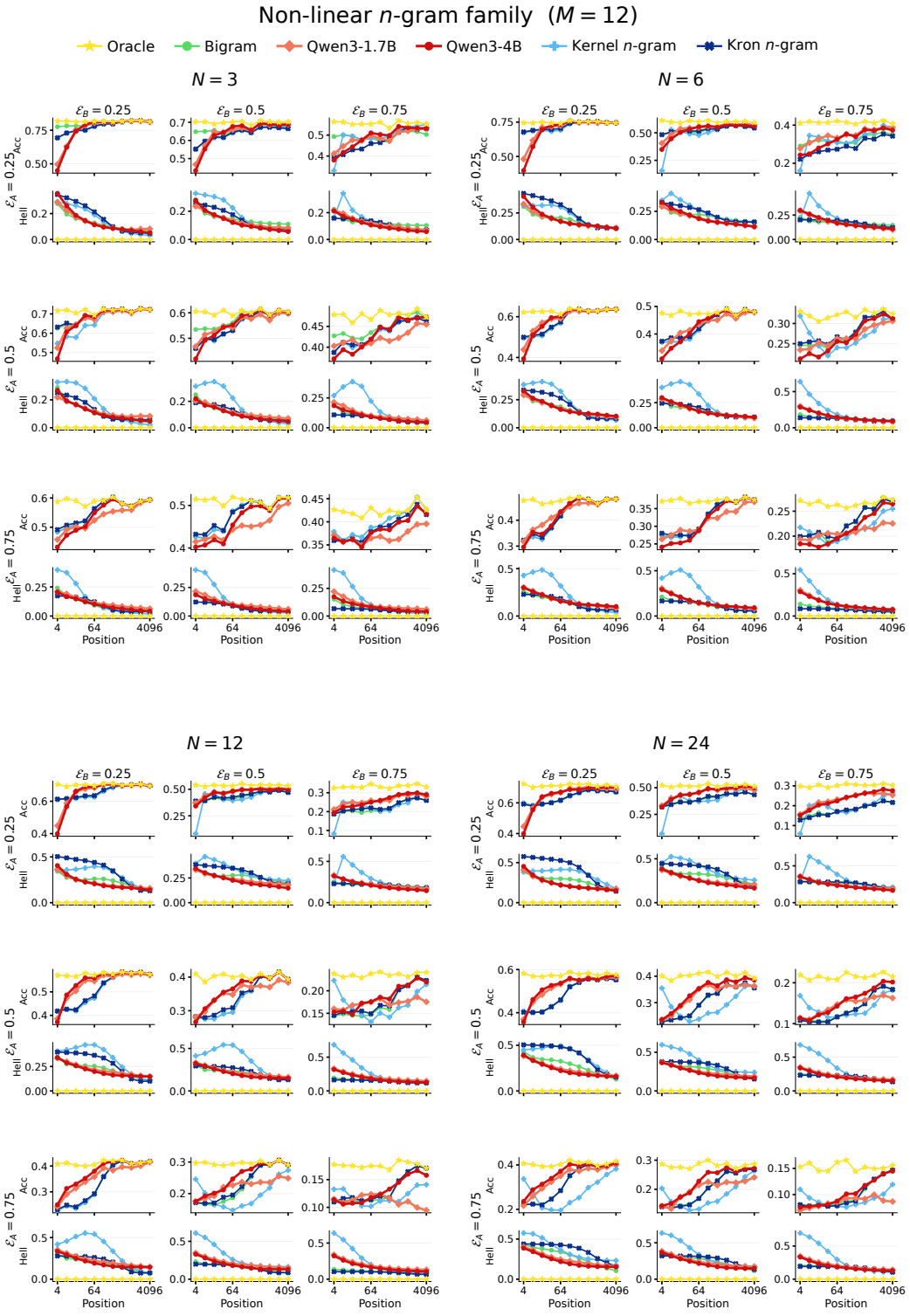


Figure 51: Non-linear n-gram family vs. position on 12-state HMMs ($M = 12$). Each of the four blocks is a fixed observation-alphabet size $N \in \{3, 6, 12, 24\}$; within a block, rows vary the transition entropy $\mathcal{E}_A$ and columns the emission entropy $\mathcal{E}_B$ (both $\in \{0.25, 0.5, 0.75\}$). Every cell shows top-1 accuracy (top) and Hellinger distance to the oracle posterior (bottom) versus sequence position (log axis). Oracle, Bigram, Qwen3-1.7B and Qwen3-4B are drawn for reference alongside Kernel n-gram and Kron n-gram; the n-gram window is selected per position by validation. Higher accuracy / lower Hellinger is better.

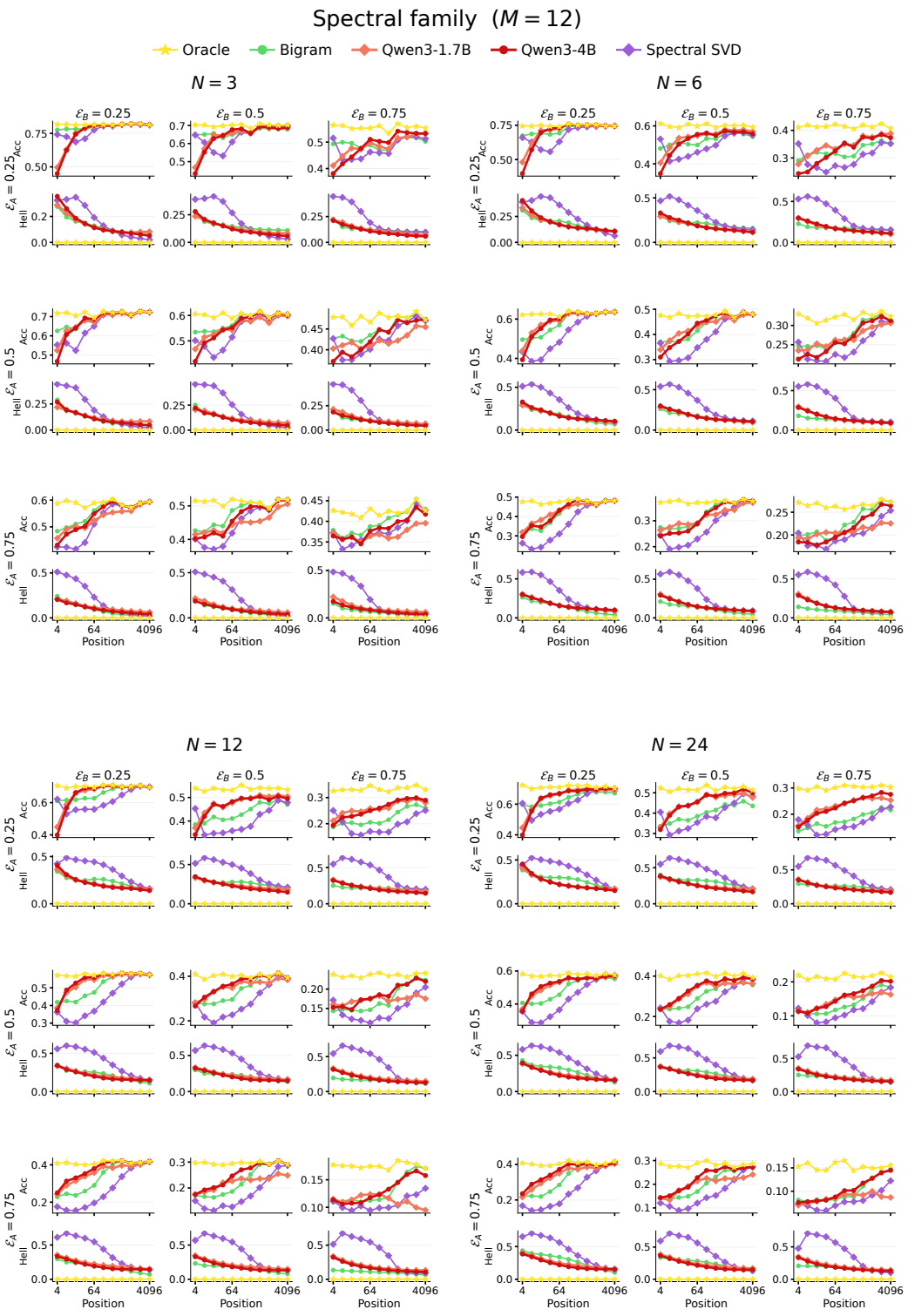


Figure 52: Spectral family vs. position on 12-state HMMs ($M = 12$). Each of the four blocks is a fixed observation-alphabet size $N \in \{3, 6, 12, 24\}$; within a block, rows vary the transition entropy $\mathcal{E}_A$ and columns the emission entropy $\mathcal{E}_B$ (both $\in \{0.25, 0.5, 0.75\}$). Every cell shows top-1 accuracy (top) and Hellinger distance to the oracle posterior (bottom) versus sequence position (log axis). Oracle, Bigram, Qwen3-1.7B and Qwen3-4B are drawn for reference alongside Spectral SVD (Spectral Norm was not run at $M = 12$). Higher accuracy / lower Hellinger is better.

G Principal Activations Probe (PAP) and Results

G.1 Null test on linear probing the belief simplex

A high R^2 on a held-out split shows that the probe generalizes, but it does not by itself rule out that a sufficiently expressive linear map could fit any target from the residual stream. To check that the recovered geometry reflects alignment between the activations and the belief, we run a label-shuffling null test. For each activation $x_{\ell,t}$ in the training split we permute the corresponding belief label b_t across examples, breaking the activation–belief correspondence while preserving the marginal distribution of beliefs. We then refit the probe on the shuffled pairs using the same PCA rank and ridge regularization as in the main experiments, and evaluate it on an unshuffled test split.

G.2 Belief probing at very short sequence length

When sequence length is 4, the R^2 scores trend behaves differently than longer sequence. The reason is that the belief vectors has less diversity at such short sequence length (this HMM's mixing rate is 4.18). One can see the simplex in Figure 53, right.

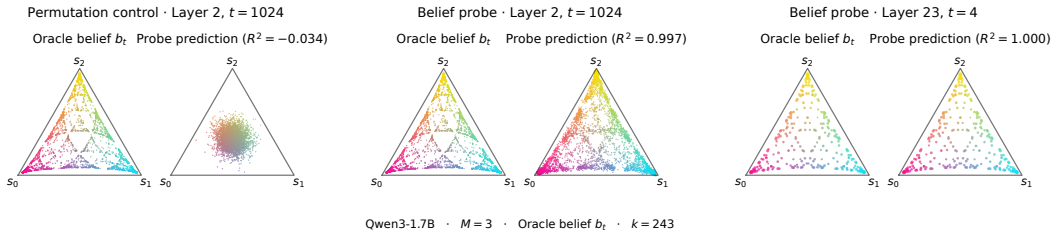


Figure 53: (Left) Null test of linear belief representation probing. (Middle) Comparison setting. (Right) Geometry when sequence length is 4.

G.3 Causal intervention is ineffective when LLMs don't perform well

Empirically, we see models like OLMo-2-1B struggle on the HMM prediction task F.1. How do their internals look like? OLMo-2-1B admits a high-R^2 probe on the Oracle belief, yet causal intervention fails entirely even under a full residual patch (Figure 54; contrast with Figure 8).

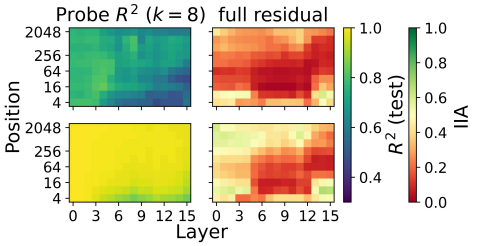


Figure 54: OLMo-2-1B probing and causal patching on the two HMM regimes: high R^2 but failed intervention, contrasting with Figure 8.

G.4 Causal representation quality aligns with LLM empirical performance.

In main paper Section 4.3, we introduced two regimes. In the *belief-dominated* regime, it require integrating observations over long contexts to maintain an accurate belief over hidden states. In the *observation-dominated* regime, it make the most recent observation highly informative, so a bigram predictor is near-optimal. Recall that the PAP results (Figure 11): in the belief-dominated regime, Linear n-gram has a distinctly stronger causal intervention effect than Bigram or Spectral; in the observation-dominated regime, all three have strong effects. This difference tracks empirical performance, as shown in Figure 55. In the belief-dominated regime, the LLM converges to Linear

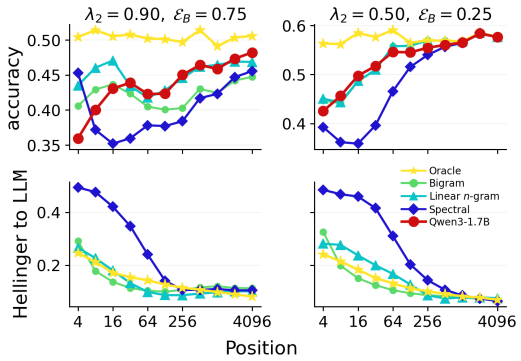


Figure 55: LLM empirical performance in the two HMM regimes.

n-gram by position 64 while the other two trail through position 4096; in the observation-dominated regime, all three converge to the Oracle by position 1024. The association between causal intervention effect and LLM predictive accuracy holds consistently across settings.

G.5 Layers as Stages of Computation

Both PAP and causal intervention results show strong indication that early layers and late layers have distinct computation roles. We further compute the centered Kernel alignment scores between each layers, as shown in Figure 56, left. We see that early-to-mid layers are aligned closely, and late layers form another group. We further investigate the projection from each layer residual to the output token space using the `lm_head`, we calculate the mass of the valid observation tokens from the projected outputs. As shown in 56 right, the later layers live in the output space. This partially explains the phenomenon we see in Section 4 on the belief-dominated regime — patching (only) the late layers causally affect the LLM outputs.

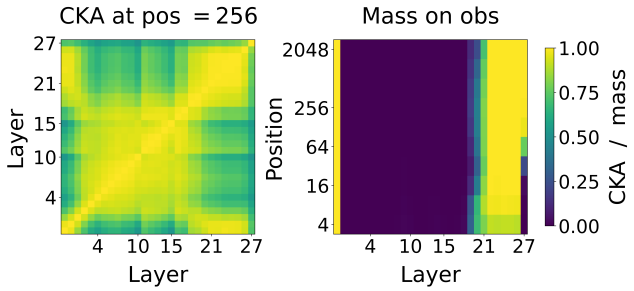


Figure 56: Left: CKA between layers. Right: Probability mass on valid observations.

G.6 Quality at Varying top-k Principal Components

To see how much of the belief geometry lives in the leading directions of the residual stream, we project each layer's activations onto their top-k principal components and fit a linear probe to recover the oracle (Bayesian) belief b_t from that k-dimensional subspace. Figure 57 reports the held-out probe R^2 as a function of transformer layer (x-axis) and token position (y-axis), across the nine HMM configurations with $\mathcal{E}(\mathbf{A}), \mathcal{E}(\mathbf{B}) \in \{0.25, 0.5, 0.75\}$ and PCA dimensions $k \in \{4, 8, 16, 32, 64, 128, 2048\}$, where $k = 2048$ effectively corresponds to the full residual stream. Three trends emerge: (i) probe performance saturates quickly with k, with most belief information captured by roughly 64–128 principal components and little improvement beyond that; (ii) belief information is distributed non-uniformly across depth, as small-k probes perform worst in the middle layers, suggesting that belief representations are spread across many directions there but become concentrated into leading components in early and late layers; and (iii) the number of components required for saturation increases with inference difficulty, with low-entropy regimes saturating in very few PCs and higher emission entropy $\mathcal{E}(\mathbf{B})$ requiring more components. Together, these results suggest that the top-k PCA subspace provides a faithful low-dimensional summary of the belief representation, motivating the use of modest values of k in the PCA-based causal interventions below.

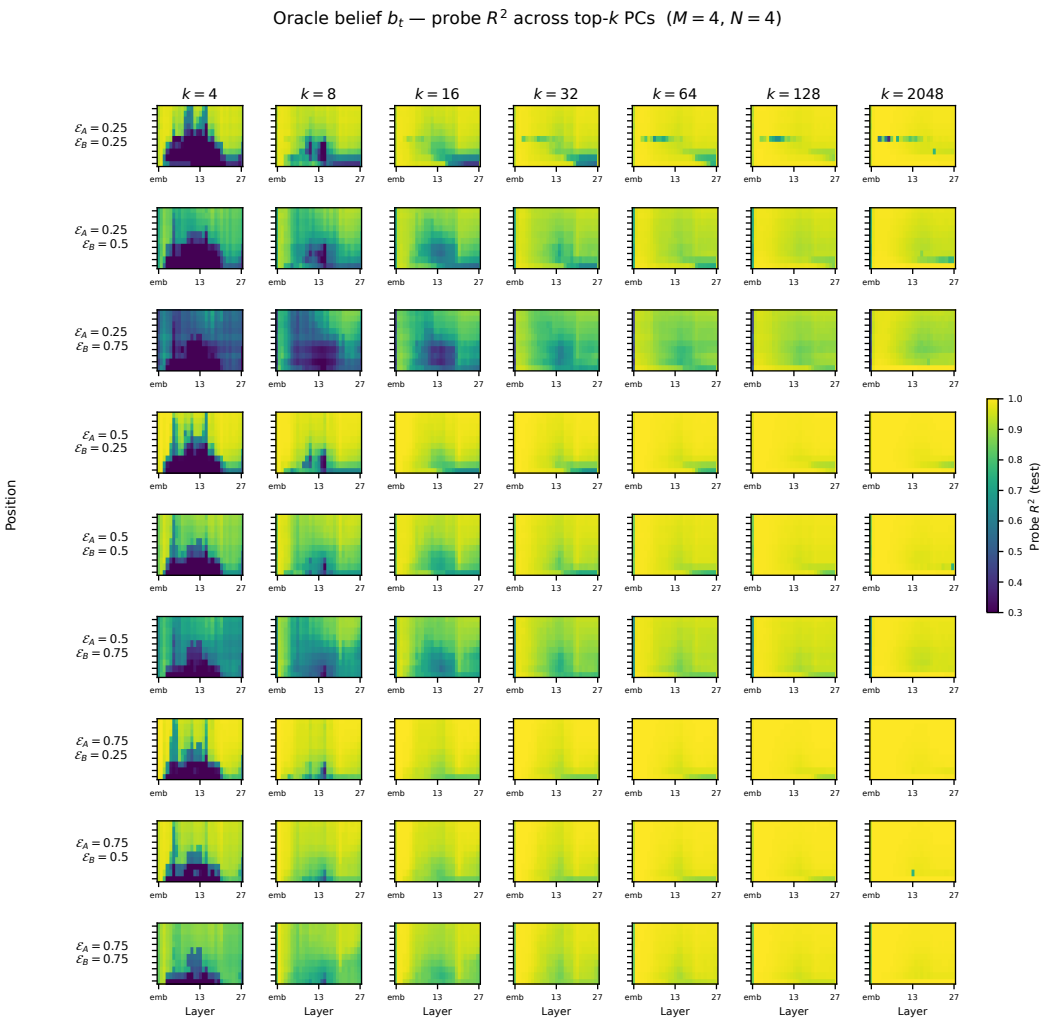


Figure 57: **Probe R^2 for recovering the oracle belief from top-k principal components.** Each panel shows the held-out linear-probe R^2 for predicting the oracle belief b_t from the top-k PCA subspace of Qwen3-1.7B residual activations, as a function of transformer layer (x: $\texttt{emb}, 0, \ldots, 27$) and token position ($y$: $4, \ldots, 2048$). Rows correspond to the nine HMM configurations with $\mathcal{E}(\mathbf{A}), \mathcal{E}(\mathbf{B}) \in \{0.25, 0.5, 0.75\}$, and columns vary the retained PCA dimension k, with $k = 2048$ corresponding to the full residual stream.

G.7 PAP Hyperparameter Robustness

PCA rank. Probe quality improves monotonically with PCA rank (Figure 57), but causal performance is non-monotonic. Mean IIA peaks at $k = 8$ for $N = 4$ and $k = 16$ for $N = 8$, then declines sharply at larger ranks (Table 3). We therefore use these peak ranks in the corresponding intervention experiments.

Table 3: **Mean IIA across PCA ranks.** Bold indicates the best rank for each observation alphabet size N.

N	$k = 4$	$k = 8$	$k = 16$	$k = 32$	$k = 64$	$k = 128$	$k = 256$
4	.938	**.965**	.930	.228	.006	.002	.001
8	.910	.919	**.931**	.402	.009	$<$.001	$<$.001

Ridge regularization. PAP is largely insensitive to the ridge coefficient over a broad range: mean IIA remains nearly unchanged for $\alpha \leq 10^7$, degrading only at substantially stronger regularization (Table 4).

Table 4: **Mean IIA across ridge coefficients α.**

α	1	10^4	10^6	10^7	10^8
IIA	.949	.949	.940	.948	.773

G.8 Alternative Intervention Metric

The main text reports interchange intervention accuracy (IIA), which measures agreement between the discrete predictions of the patched model and target algorithm. We additionally evaluate distributional agreement using

$$\text{KL-Eff} = 1 - \frac{D_{\text{KL}}!\left(p_{\text{target}}, |, p_{\text{patched}}\right)}{D_{\text{KL}}!\left(p_{\text{target}}, |, p_{\text{base}}\right)},$$

where p_{target} is the target algorithm distribution and p_{patched} and p_{base} are the patched and unpatched model distributions. A value of 1 denotes perfect recovery of the target distribution, 0 denotes no improvement over the unpatched model, and negative values indicate that patching moves the prediction farther from the target.

IIA and KL-Eff yield the same qualitative conclusion: Soft n-gram has the strongest causal effect across all nine HMM configurations (Table 5).

Table 5: **IIA / KL-Eff for Qwen3-1.7B ($N = 4$, $k = 8$).** Results are 3×3 max-pooled. B = Bigram, SN = Soft n-gram, Sp = Spectral. Bold indicates the best result within each configuration.

Rep.	.25/.25	.25/.50	.25/.75	.50/.25	.50/.50	.50/.75	.75/.25	.75/.50	.75/.75
B	.831/.900	.563/.770	.542/.550	.919/.904	.715/.731	.420/.627	.832/.804	.598/.677	.405/.403
SN	**1.000/.985**	**1.000/.989**	**.992/.971**	**.999/.976**	**1.000/.964**	**.967/.919**	**.985/.930**	**.989/.911**	**.705/.471**
Sp	.810/.866	.674/.758	.638/.599	.892/.909	.609/.782	.553/.115	.804/.860	.590/.457	.359/.356

G.9 Robustness Across Model Families

We evaluate Llama 3.2 3B, OLMo 2 1B, and Gemma 3 1B in addition to Qwen3-1.7B. Behaviorally, Llama approaches the Oracle similarly to Qwen, whereas OLMo and Gemma remain substantially farther from the Oracle (Table 6). PAP closely tracks this distinction: Llama exhibits the same regime-dependent causal signature as Qwen, while OLMo and Gemma show substantially weaker or less consistent intervention effects (Table 7). Thus, the mechanistic signature identified by PAP is strongest in models that successfully solve the prediction task.

Table 6: **Accuracy / Hellinger distance to the Oracle across model families.** Context length 1024, $N = 4$.

Model	.25/.25	.25/.50	.25/.75	.50/.25	.50/.50	.50/.75	.75/.25	.75/.50	.75/.75
Oracle	.789/.000	.667/.000	.491/.000	.704/.000	.570/.000	.410/.000	.568/.000	.476/.000	.354/.000
Qwen3-1.7B	.787/.092	.641/.106	.459/.098	.702/.085	.560/.090	.382/.084	.565/.082	.452/.081	.317/.075
Llama3.2-3B	.777/.115	.637/.115	.457/.105	.695/.097	.560/.097	.381/.087	.562/.088	.459/.084	.305/.074
OLMo2-1B	.666/.265	.404/.339	.208/.368	.518/.288	.308/.334	.215/.321	.301/.326	.226/.323	.228/.282
Gemma3-1B	.770/.209	.600/.246	.406/.265	.673/.246	.530/.262	.354/.268	.527/.263	.411/.270	.297/.272

Table 7: **PAP across model families (R^2 / IIA), $N = 4$, $k = 8$.** B = Bigram, SN = Soft n-gram, Sp = Spectral; bold indicates the best value within each model and configuration.

Model–rep.	.25/.25	.25/.50	.25/.75	.50/.25	.50/.50	.50/.75	.75/.25	.75/.50	.75/.75
Qwen–B	**.984**/.831	**.966**/.563	**.890**/.542	**.988**/.919	**.971**/.715	**.894**/.420	**.979**/.832	**.949**/.598	**.826**/.405
Qwen–SN	.977/**1.000**	.932/**1.000**	.804/**.992**	.970/**.999**	.912/**1.000**	.752/**.967**	.919/**.985**	.826/**.989**	.561/**.755**
Qwen–Sp	.927/.810	.769/.674	.385/.638	.932/.892	.799/.609	.395/.553	.863/.804	.674/.590	.239/.359
Llama–B	.977/.971	**.957**/.677	**.878**/.623	**.978**/.925	**.963**/.905	**.882**/.658	**.973**/.930	**.941**/.860	**.816/.736**
Llama–SN	**.979/.995**	.938/**.996**	.776/**.941**	.971/.949	.923/**.926**	.740/**.908**	.915/**.967**	.822/**.936**	.554/.605
Llama–Sp	.920/.921	.756/.787	.362/.648	.919/**.958**	.778/.828	.370/.589	.846/.949	.655/.692	.232/.392
OLMo–B	**.990**/.772	**.965**/.645	**.879**/.374	**.989**/.725	**.968**/.479	**.883**/.387	**.976**/.456	**.943/.406**	**.818**/.315
OLMo–SN	.980/.834	.941/**.749**	.775/**.429**	.969/**.743**	.920/**.613**	.732/**.405**	.915/**.493**	.820/.366	.551/.292
OLMo–Sp	.927/**.902**	.771/.572	.424/.328	.923/.696	.781/.467	.443/.285	.847/.426	.677/.334	.331/**.330**
Gemma–B	**.987**/.493	**.962**/.362	**.875**/.303	**.985**/.384	**.963**/.359	**.880**/.277	**.972**/.298	**.939**/.309	**.815**/.274
Gemma–SN	.972/**.630**	.927/**.567**	.766/**.613**	.966/**.457**	.915/**.482**	.719/**.559**	.913/.294	.817/.293	.538/**.413**
Gemma–Sp	.923/.513	.755/.426	.363/.388	.922/.410	.776/.352	.373/.317	.844/**.305**	.652/**.310**	.239/.287

G.10 Robustness Across Model Sizes

We further evaluate Qwen3 models at 0.6B, 1.7B, and 4B parameters. The qualitative PAP signature is stable across scale: Soft n-gram interventions are strongest in regimes requiring belief tracking, while Bigram and other representations become more causally effective as the observations themselves become increasingly predictive (Table 8).

Table 8: **PAP across Qwen3 sizes (R^2 / IIA), $N = 4$, $k = 8$.** Abbreviations follow Table 7.

Size–rep.	.25/.25	.25/.50	.25/.75	.50/.25	.50/.50	.50/.75	.75/.25	.75/.50	.75/.75
0.6B–B	.984/.777	.965/.530	.888/**.570**	.986/.850	.968/.618	.891/.482	.976/.757	.947/.582	**.826**/.528
0.6B–SN	**.977/1.000**	.927/**1.000**	.801/.987	.965/**1.000**	.908/.992	.733/.957	.916/.967	.817/.918	.547/.677
0.6B–Sp	.926/.790	.768/**.771**	.380/.636	.929/.892	.794/.612	.391/.510	.857/.801	.670/.546	**.241/.402**
1.7B–B	.984/.831	**.966**/.563	**.890**/.542	**.988**/.919	**.971**/.715	**.894**/.420	**.979/.832**	**.949**/.598	**.826**/.405
1.7B–SN	**.977/1.000**	**.932/1.000**	**.804/.992**	**.970**/.999	**.912/1.000**	**.752/.967**	**.919**/.985	**.826/.989**	**.561/.705**
1.7B–Sp	**.927**/.810	**.769**/.674	**.385/.638**	**.932**/.892	**.799**/.609	**.395/.553**	**.863**/.804	**.674/.590**	.239/.359
4B–B	**.989/.988**	.965/**.687**	.881/.439	**.988/.976**	.968/**.774**	.886/**.754**	.977/.828	.946/**.750**	.820/**.622**
4B–SN	.975/**1.000**	.921/**1.000**	.786/.987	.964/**1.000**	.902/.998	.731/.948	.910/**.997**	.821/.932	.553/.685
4B–Sp	.922/**.905**	.753/.720	.351/.558	.924/**.919**	.781/**.680**	.368/.432	.852/**.887**	.660/.548	.228/.379

Summary. Across HMM configurations, model families, and model sizes, PAP most clearly recovers the proposed mechanistic signature in models that perform the prediction task well. In belief-dominated regimes, successful models are most strongly influenced by Soft n-gram representations; in observation-dominated regimes, where recent observations are already highly informative, multiple representations can exert causal influence. This agreement between predictive behavior and causal intervention provides additional evidence that PAP identifies representations used by the model rather than merely decodable correlates.

G.11 Additional Results for Causal Interventions

In Section 4 of the main paper, we illustrated principal activations patching (PAP) using two representative HMM configurations, corresponding to $belief$-dominated and $observation$-dominated regimes. Here, we extend the analysis to the full family of HMMs spanning transition entropy $\mathcal{E}(\mathbf{A}) \in \{0.25, 0.5, 0.75\}$, emission entropy $\mathcal{E}(\mathbf{B}) \in \{0.25, 0.5, 0.75\}$, and observation alphabet size $N \in \{2, 4, 8\}$.

For the PCA-subspace and probe-inverse interventions, we choose the probe dimension based on the causal rank sweep in Appendix G.7: $k = 8$ for $N = 4$ and $k = 16$ for $N = 8$. Unless otherwise specified, the main-text experiments use $k = 8$. Results are shown for Qwen3-1.7B. Across all settings, we compare the oracle-belief probe R^2 with the interchange-intervention accuracy (IIA) obtained from full-residual, PCA-subspace, and probe-inverse patching. This allows us to assess how faithfully low-dimensional belief representations capture the causal variables used by the model during inference, as shown in Figures 58–60.

G.12 Additional Results on Algorithm Beliefs

We consider four families of algorithm beliefs: classical n-gram models (Bigram, Trigram), linear n-gram learners (Gradient CE, Gradient MSE, Ridge MSE), non-linear n-gram learners (Kernel, Kron), and spectral predictors (Norm and Norm (GT)). For each algorithm, we train a linear probe with dimension $k = 8$ to predict that algorithm's next-token distribution from the residual stream. We then evaluate both the held-out probe R^2 and the interchange-intervention accuracy (IIA) obtained from a probe-inverse intervention, which patches the corresponding $k = 8$ latent subspace from a source sequence into a target sequence and measures the resulting agreement with the algorithm's own prediction. Using the same subspace for both decoding and intervention enables a direct comparison between representational accessibility and causal influence.

Results are shown for Qwen3-1.7B across the nine HMM configurations defined by transition entropy $\mathcal{E}(\mathbf{A}) \in \{0.25, 0.5, 0.75\}$ and emission entropy $\mathcal{E}(\mathbf{B}) \in \{0.25, 0.5, 0.75\}$, with $M = N = 4$, as shown in Figures 61–64.

G.13 More Examples of Algorithm Operator Probing

Complementing the algorithm *beliefs*, we probe for the algorithm *operators*—the weight matrices that parameterize each predictor—to test whether the residual stream linearly encodes the operators themselves, not just their outputs. For each operator we fit a PCA(k) + ridge probe at every layer (token position 64) and report the held-out MSE of the recovered operator, one curve per probe dimension k. We consider three operators: the linear soft n-gram weight $W \in \mathbb{R}^{N \times Nn}$ (window n=4, lags $o_{t-3:t}$), the spectral observation operator B_k^{norm}, and the non-linear Kronecker n-gram weight (window n=4). A representative ground-truth vs. recovered operator matrix appears in the main paper (Fig. 12); here we summarize the recovery MSE across the nine non-trivial entropy configurations at N=4 (Qwen3-1.7B). MSE decreases monotonically with the probe dimension k and is lowest at the early and late layers (worst in the middle), mirroring the belief and prediction probes, as shown in Figures 65–67.

G.14 Different LLM Sizes

To assess whether our findings depend on model scale, we repeat the oracle-belief probing and full-residual interchange-intervention experiments using Qwen3-4B and compare the results to Qwen3-1.7B. Figure 68 shows that the qualitative patterns are highly consistent across model sizes. In particular, the oracle belief remains strongly decodable from the residual stream, and belief patching continues to exhibit high interchange-intervention accuracy in the same regions of layer-position space. While the larger model generally achieves slightly higher probe R^2 and IIA, the overall geometry and causal localization of the belief representation remain largely unchanged, suggesting that the mechanisms identified in this work are robust across model scale.

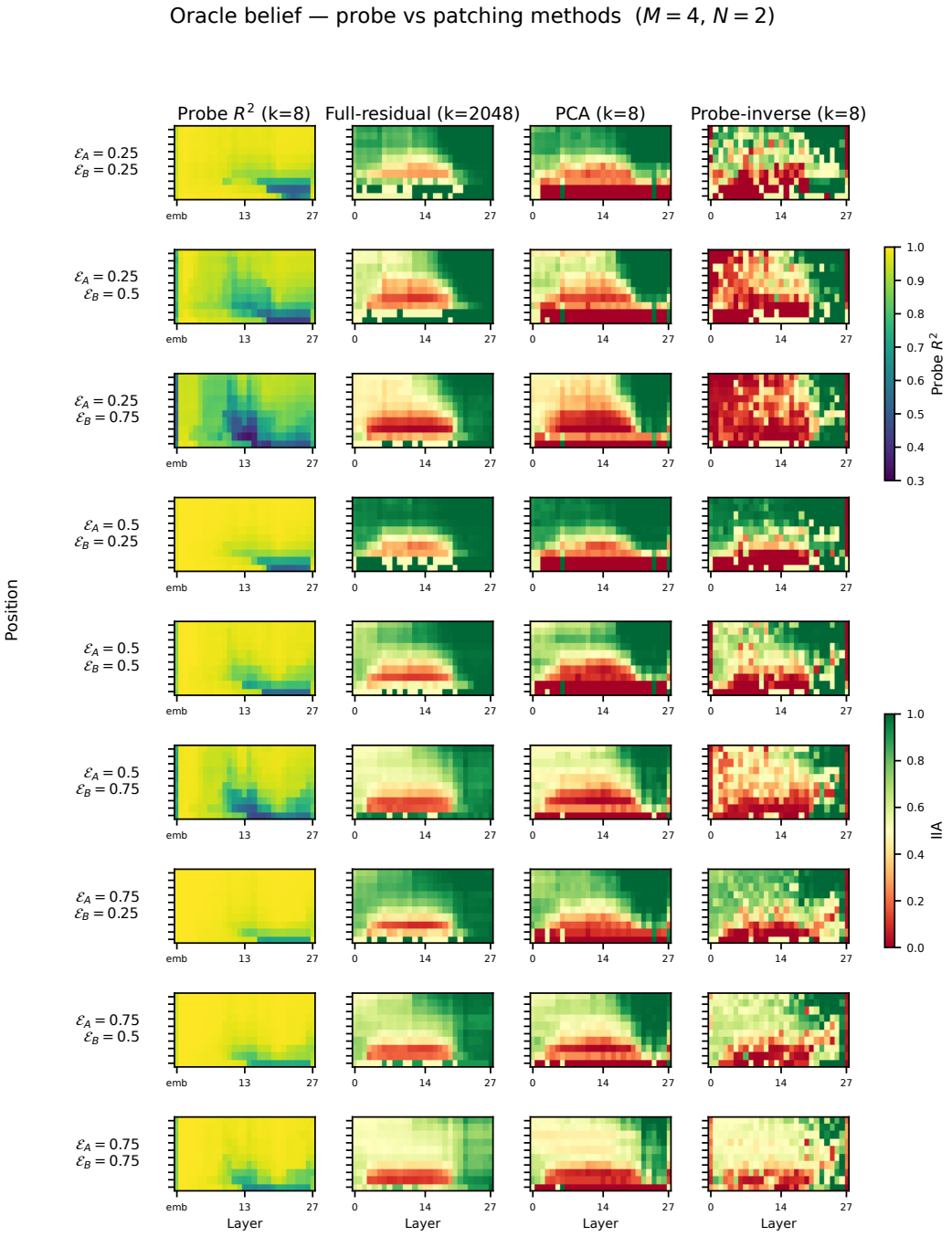


Figure 58: **Causal interventions for $N = 2$.** Rows are HMM configs (transition entropy $\mathcal{E}(\mathbf{A})$, emission entropy $\mathcal{E}(\mathbf{B})$). Columns: linear-probe R^2 for the oracle belief $\mathbb{P}(h_t|\mathbf{o}_{1:t}, \boldsymbol{\lambda})$ (k=8), then interchange-intervention accuracy (IIA) for full-residual (k=2048), PCA (k=8), and probe-inverse (k=8) patches. Each panel is a layer × position heatmap.

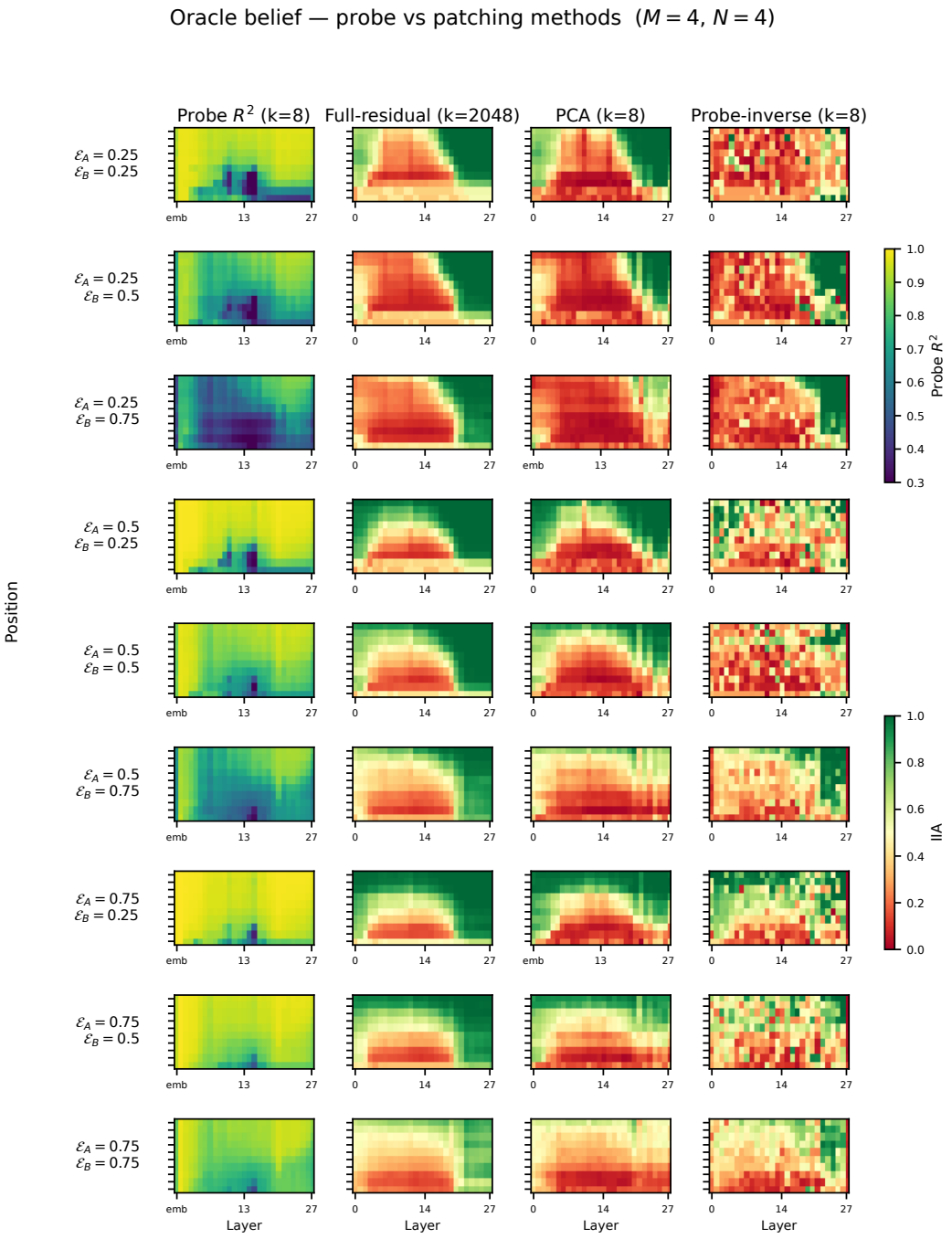


Figure 59: **Causal interventions for $N = 4$.** Rows are HMM configs (transition entropy $\mathcal{E}(\mathbf{A})$, emission entropy $\mathcal{E}(\mathbf{B})$). Columns: linear-probe R^2 for the oracle belief $\mathbb{P}(h_t|\mathbf{o}_{1:t}, \boldsymbol{\lambda})$ (k=8), then interchange-intervention accuracy (IIA) for full-residual (k=2048), PCA (k=8), and probe-inverse (k=8) patches. Each panel is a layer $\times$ position heatmap.

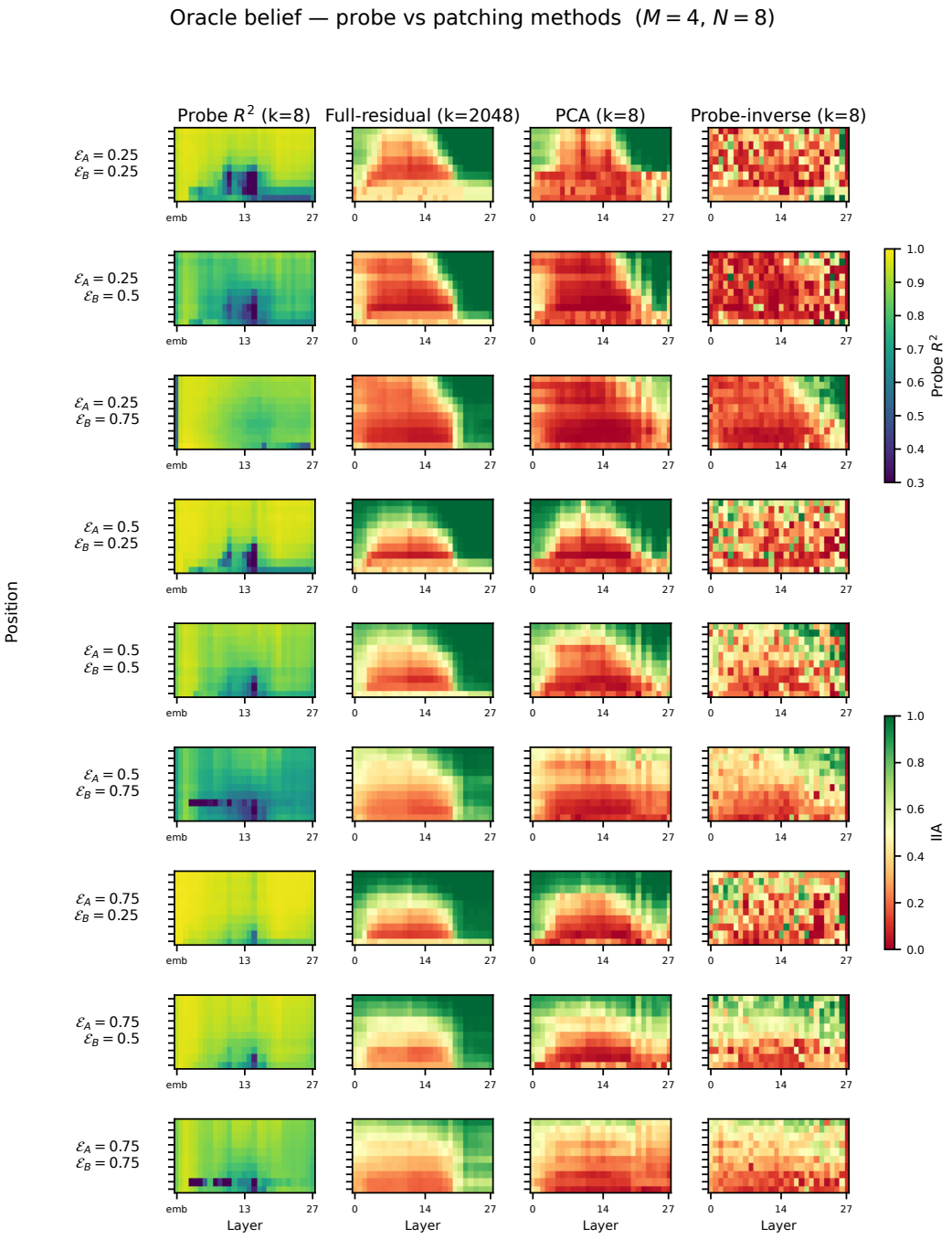


Figure 60: **Causal interventions for $N = 8$.** Rows are HMM configs (transition entropy $\mathcal{E}(\mathbf{A})$, emission entropy $\mathcal{E}(\mathbf{B})$). Columns: linear-probe R^2 for the oracle belief $\mathbb{P}(h_t|\mathbf{o}_{1:t}, \boldsymbol{\lambda})$ (k=8), then interchange-intervention accuracy (IIA) for full-residual (k=2048), PCA (k=8), and probe-inverse (k=8) patches. Each panel is a layer × position heatmap.

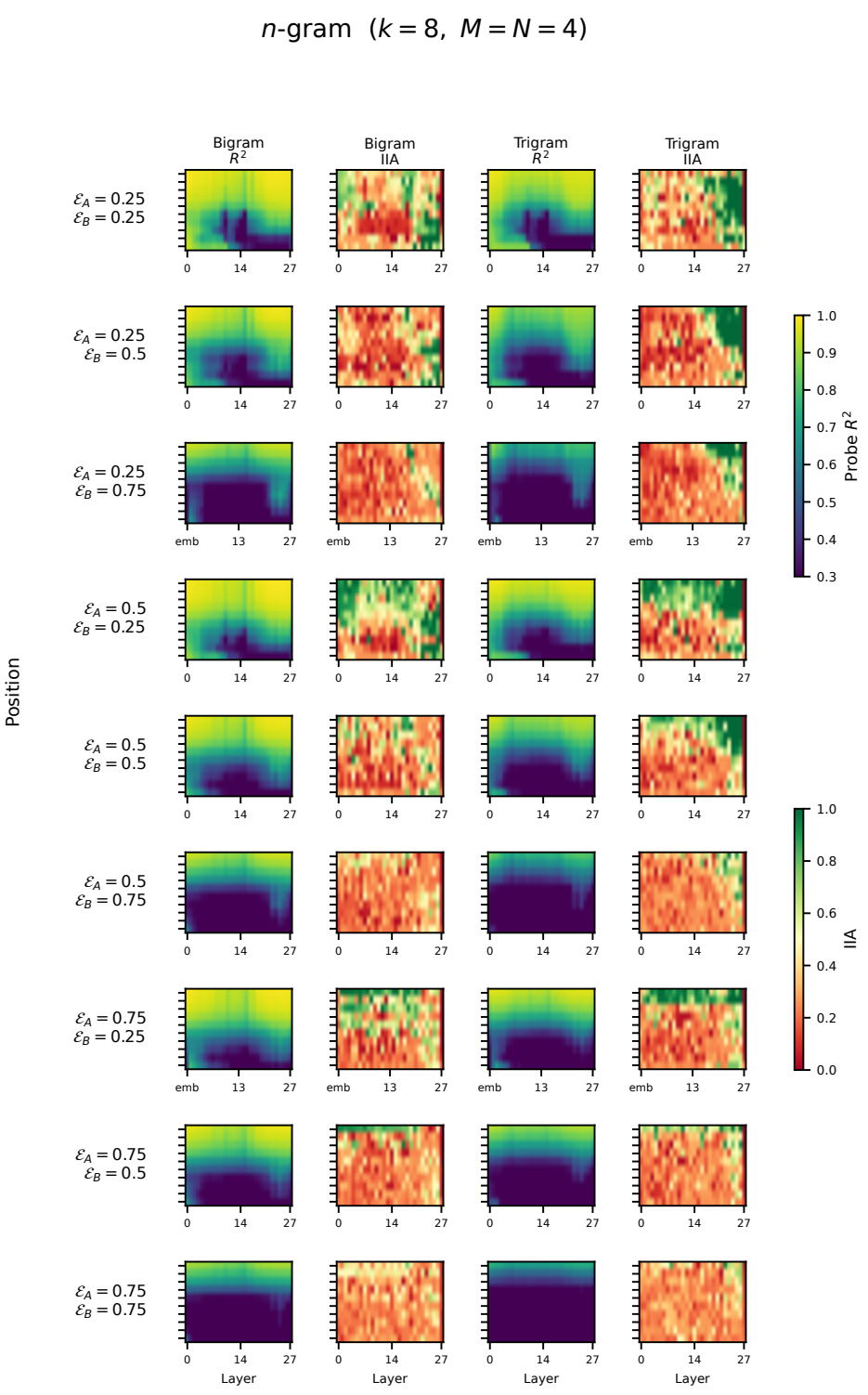


Figure 61: **Algorithm beliefs: classical n-gram predictors ($k = 8$, $M = N = 4$).** Rows correspond to HMM configurations indexed by transition entropy $\mathcal{E}(\mathbf{A})$ and emission entropy $\mathcal{E}(\mathbf{B})$. For each target predictor (Bigram, Trigram), the first column shows the held-out linear-probe R^2 for decoding the predictor's next-token distribution from the residual stream (`viridis`, $[0.3, 1]$), and the second column shows the probe-inverse interchange-intervention accuracy (IIA) obtained by patching the corresponding $k = 8$ latent subspace (`RdYlGn`, $[0, 1]$). Each panel is a layer×position heatmap. Results are shown for Qwen3-1.7B.

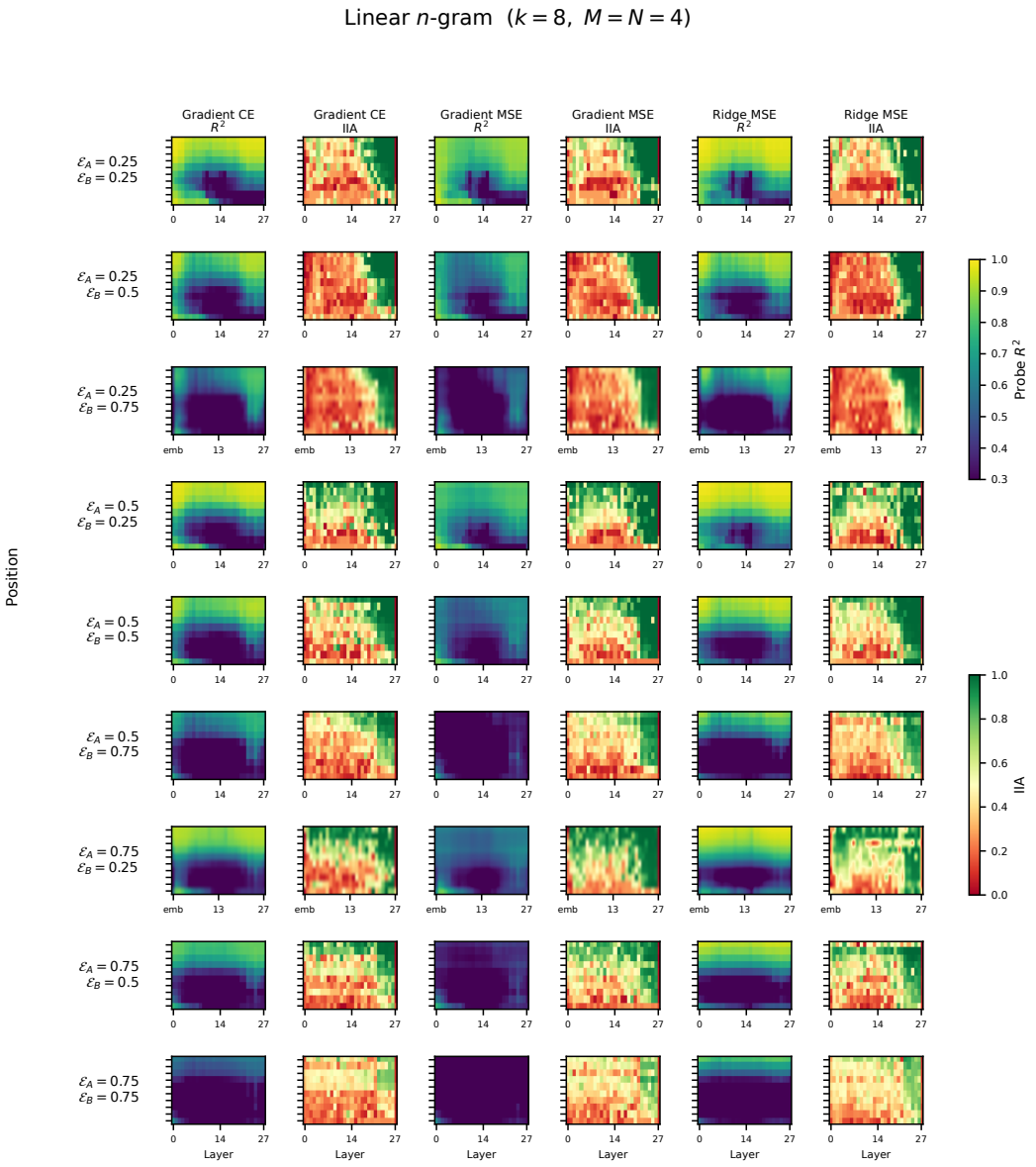


Figure 62: **Algorithm beliefs: linear n-gram learners ($k = 8$, $M = N = 4$).** Same format as Fig. 61, for the linear predictors Gradient CE, Gradient MSE, and Ridge MSE.

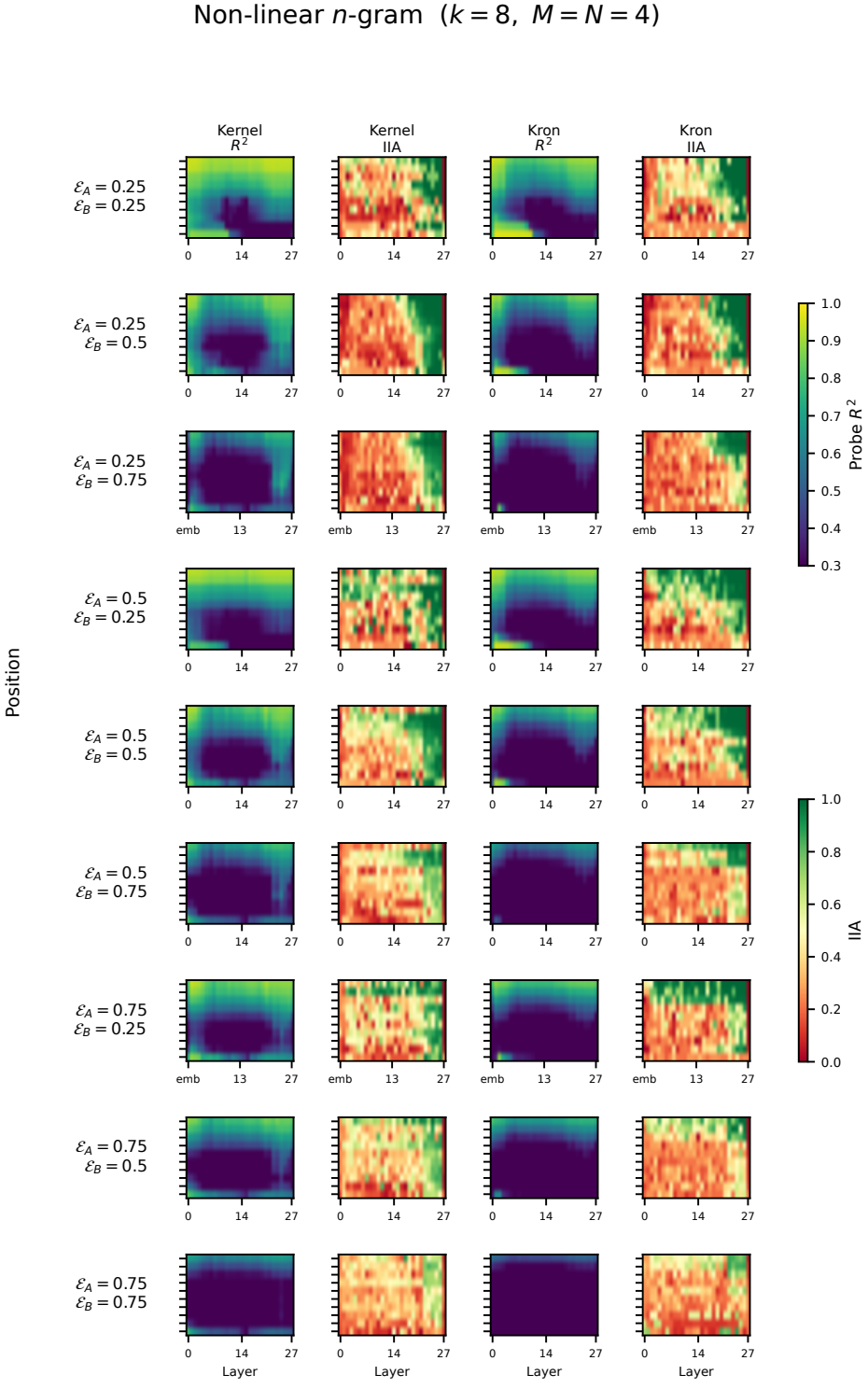


Figure 63: **Algorithm beliefs: non-linear n-gram learners ($k = 8$, $M = N = 4$).** Same format as Fig. 61, for the non-linear predictors Kernel and Kron.

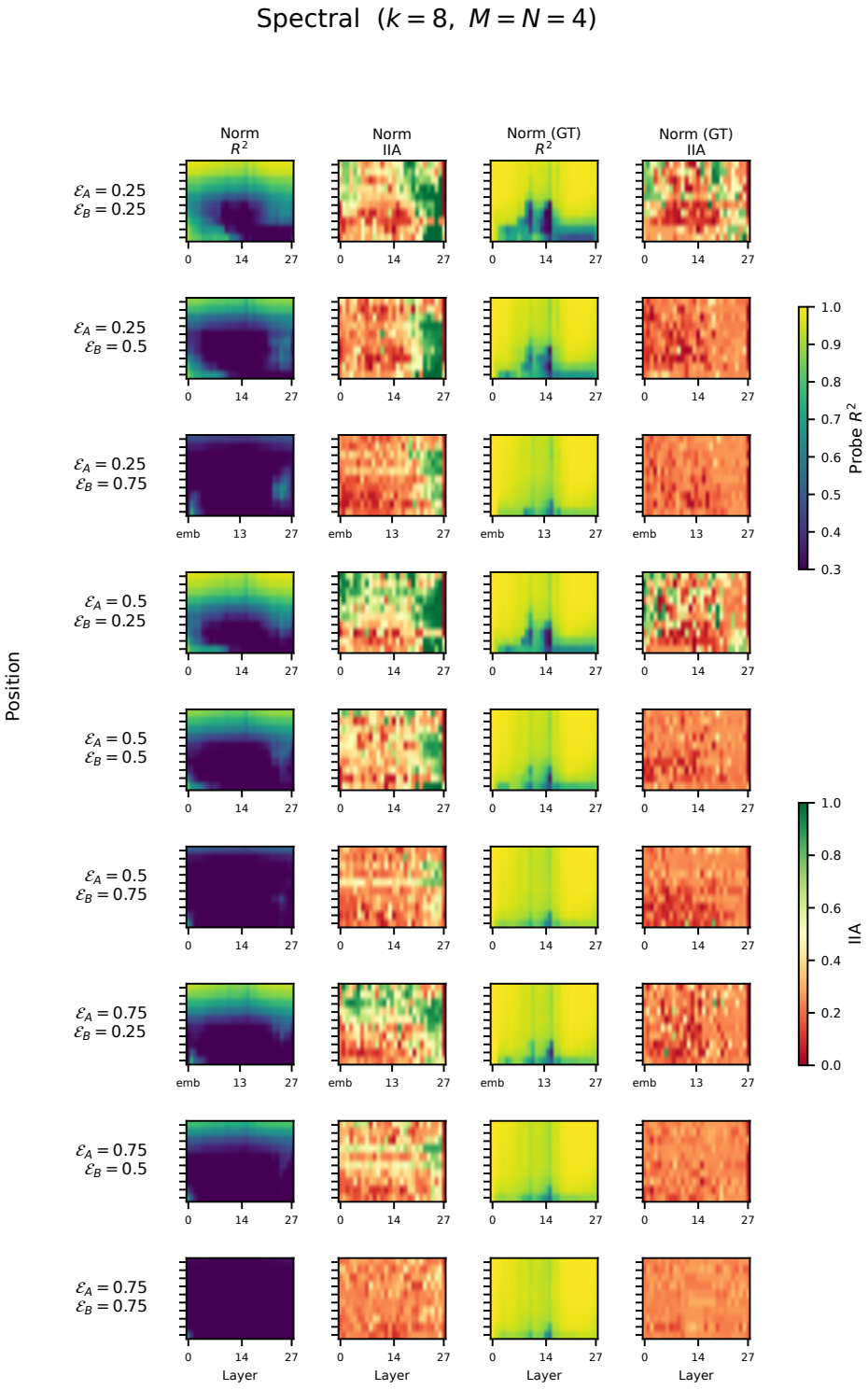


Figure 64: **Algorithm beliefs: spectral predictors ($k = 8$, $M = N = 4$).** Same format as Fig. 61, for the spectral next-token predictors Norm (normalized observable-operator recursion using empirically estimated moments) and Norm (GT), which uses the same recursion with ground-truth moments.

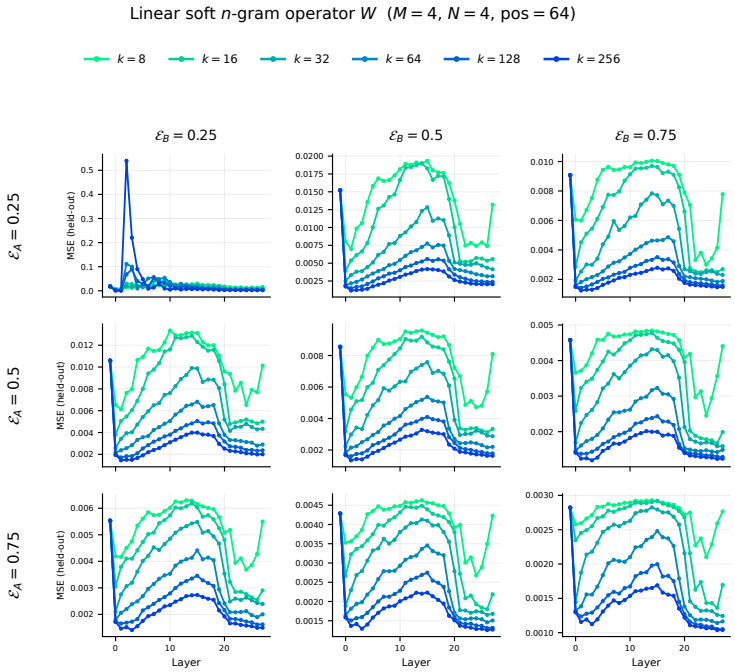


Figure 65: **Linear soft n-gram operator W (window 4, $M{=}N{=}4$).** Rows are transition entropy $\mathcal{E}(\mathbf{A})$, columns emission entropy $\mathcal{E}(\mathbf{B})$. Each cell shows the held-out MSE of the recovered weight $W \in \mathbb{R}^{K\times K\cdot 4}$ vs. layer at position 64, one curve per probe dimension k. y-axes are autoscaled per cell. Qwen3-1.7B.

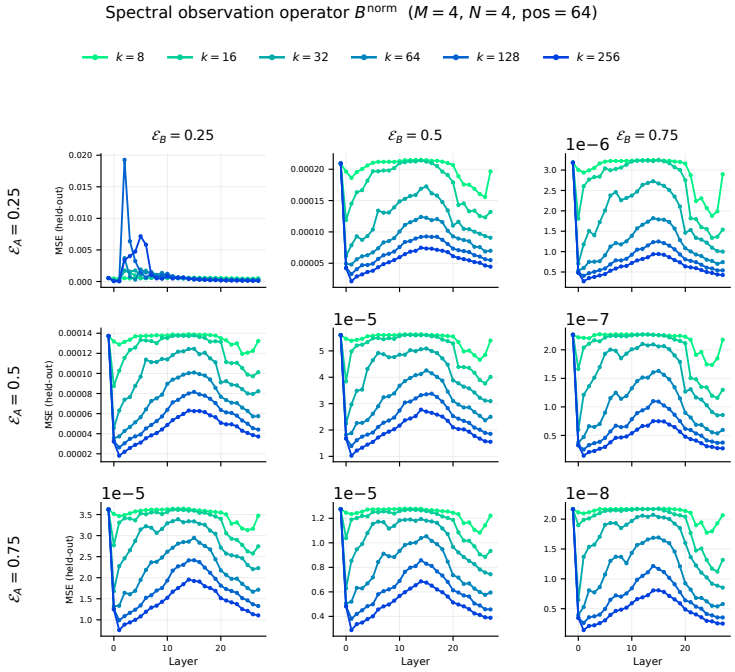


Figure 66: **Spectral observation operator B^{norm} ($M{=}N{=}4$).** As Fig. 65, for the normalized observation operator $B_k^{\mathrm{norm}} = \hat{P}_2^{\top}\hat{P}_3[k]$. Note the MSE scale shrinks sharply with emission entropy (from $\sim 10^{-4}$ at $\mathcal{E}(\mathbf{B}){=}0.25$ to $\sim 10^{-8}$ at 0.75) as the operator becomes near-uniform.

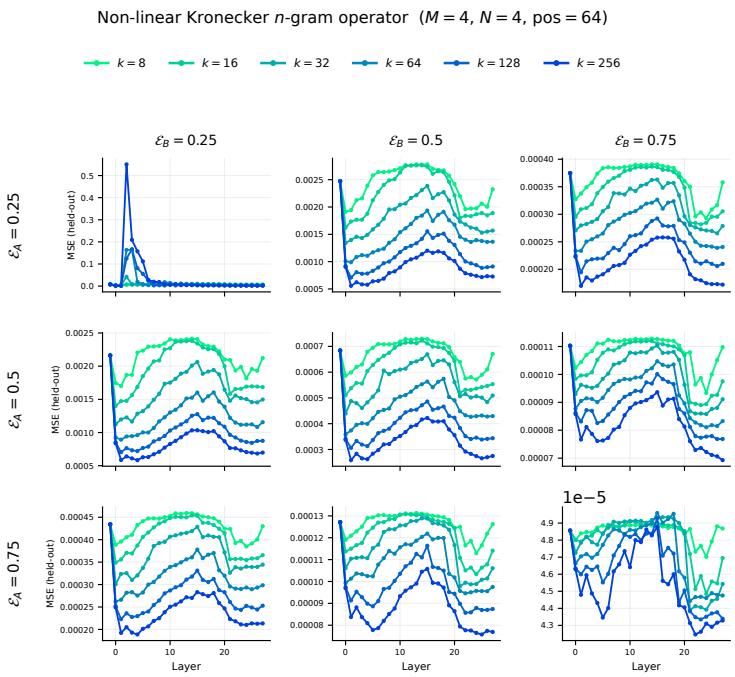


Figure 67: **Non-linear Kronecker n-gram operator (window 4, $M{=}N{=}4$).** As Fig. 65, for the Kronecker n-gram weight (product features over the last 4 observations, K^4 columns).

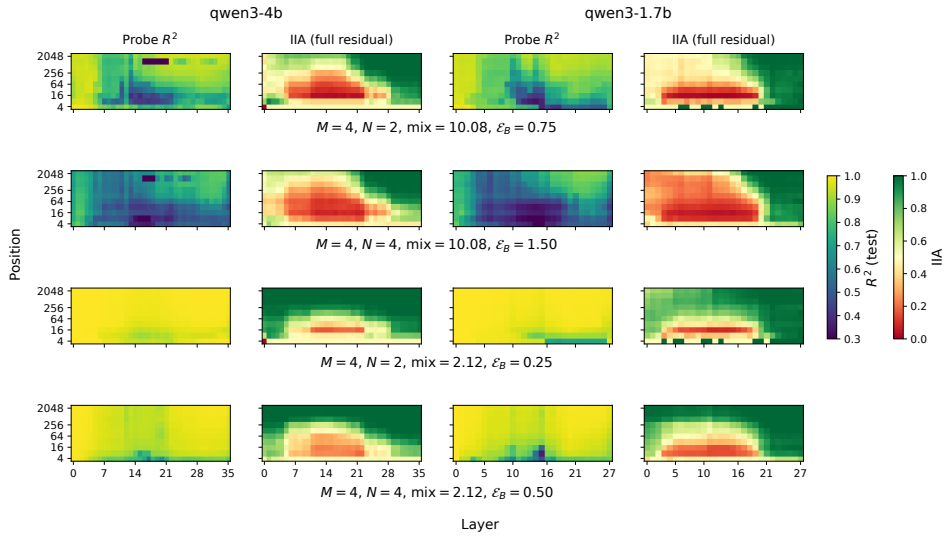


Figure 68: **Comparison across model sizes.** Oracle-belief probe R^2 and full-residual interchange-intervention accuracy (IIA) for Qwen3-1.7B and Qwen3-4B.

G.15 Patching at Different Timesteps t

The interventions in the main text patch a single token position. To study where in the sequence the belief driving the next-token prediction is stored, we instead patch contiguous windows of residual activations. For a given source-target pair, we overwrite the target residual stream with the source residual stream at a fixed layer over a window of n consecutive positions and then evaluate the resulting next-observation prediction at timestep t. We use full-residual interventions ($k = d_{\text{model}}$), corresponding to direct residual replacement. Results are reported as *IIA efficacy*.

We compare two window placements. In the *non-shifted* setting, the patch window $[t - n + 1 : t]$ includes the query position t (Fig. 69). In the *shifted* setting, the window $[t - n : t - 1]$ excludes t and patches only the preceding positions (Fig. 70). Comparing these interventions isolates the extent to which predictive information is stored locally at the current token versus distributed across earlier context. We sweep window sizes $n \in \{1, 2, 4, 6\}$ across all layers and timesteps for the nine HMM configurations with $\mathcal{E}(\mathbf{A}), \mathcal{E}(\mathbf{B}) \in \{0.25, 0.5, 0.75\}$ at $N = 4$.

Two consistent patterns emerge. First, patching windows that include the query position almost completely transfers the source prediction, even for $n = 1$, with efficacy increasing further as the window expands. This indicates that the belief used for prediction is largely represented in the residual stream at the current token. Second, shifted windows excluding the query position have little effect when $n = 1$, but their efficacy grows substantially with window size, particularly at later layers and later timesteps. This growth is most pronounced in high-emission-entropy regimes, where the current observation provides less information about the latent state and belief must be accumulated over longer contexts. In these settings, patching the preceding 4–6 positions is often sufficient to transplant the source belief. The contrast between shifted and non-shifted interventions therefore quantifies the degree to which predictive information is stored locally versus distributed across context, mirroring the observation-dominated and belief-dominated regimes discussed in the main text.

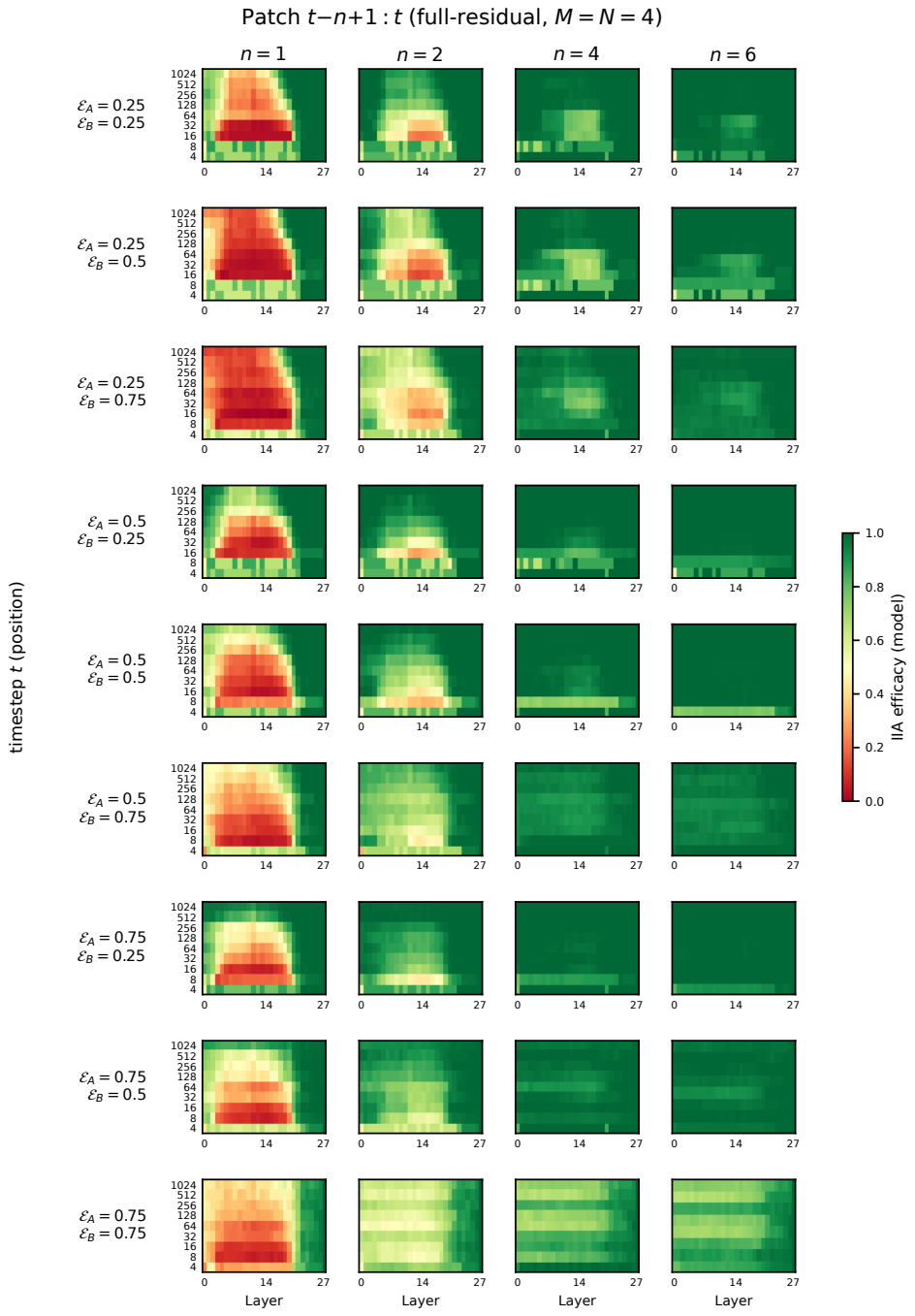


Figure 69: **Window patching including the query timestep.** IIA efficacy of a full-residual interchange intervention that overwrites the target residual stream over the window $[t-n+1:t]$, which includes the query position t, and evaluates the resulting next-observation prediction at t. Rows correspond to the nine HMM configurations with $\mathcal{E}(\mathbf{A}), \mathcal{E}(\mathbf{B}) \in \{0.25, 0.5, 0.75\}$ at $N=4$, while columns vary the window size $n \in \{1, 2, 4, 6\}$. Each panel is a layer (x: $0, \ldots, 27$) $\times$ query timestep (y: $4, \ldots, 1024$) heatmap of IIA efficacy (RdYlGn; green $=1$ indicates complete recovery of the source prediction, red $=0$ indicates no effect). Efficacy is high even for $n=1$, demonstrating that the prediction-relevant belief is largely encoded in the residual representation at the current token. Results are shown for Qwen3-1.7B.

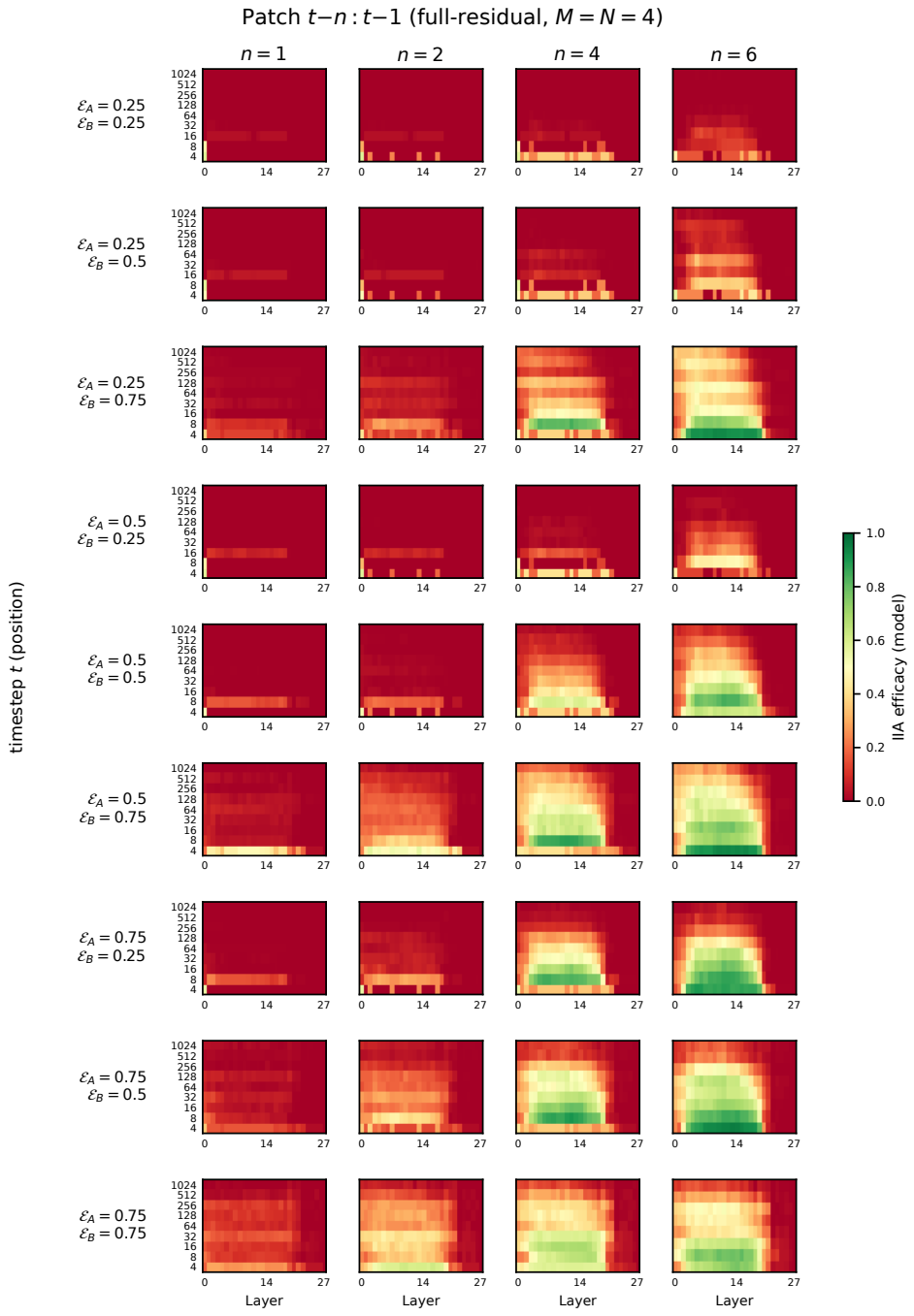


Figure 70: **Window patching excluding the query timestep.** As in Fig. 69, but the patched window $[t-n:t-1]$ is shifted one step left so that the query position t remains *untouched* and only the n preceding positions are overwritten, while the readout is still performed at t. At $n=1$, the intervention is near-inert (mostly red), indicating that the prediction does not rely solely on the immediately preceding token. Efficacy increases with window size n, with stronger effects at later layers and timesteps, particularly in high-emission-entropy regimes (larger $\mathcal{E}_B$, lower rows), where belief must be integrated over a longer context. Results are shown for Qwen3-1.7B using full-residual swaps ($k=d_{\text{model}}$).