PROBABILISTIC STATE SPACE ENUMERATION IN NEURAL NETWORK COMPUTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep neural networks typically realize dense computation by propagating information through all intermediate channels, while many approximation methods rank and retain individual computational components. We ask whether dense neural computation can instead be represented exactly as an expectation over a combinatorial state space, and approximated by evaluating only its most probable configurations. We introduce a novel probabilistic state-space framework for Transformer MLPs. The framework partitions the intermediate dimension into channel groups, defines an input-dependent distribution over binary inclusion states, and uses importance-corrected masked outputs such that the full expectation exactly recovers the dense MLP output. The key algorithmic novelty is NORDER-II, which enumerates complete joint states in decreasing probability without constructing or sorting the exponentially large state space. Truncating this ordering yields an anytime approximation governed by cumulative probability mass and a state budget. On Llama-3.2-1B and Gemma-2-2B, NORDER-II at a state budget of $B = 2048$ achieves 92.4% and 93.8% MLP-output approximation accuracy, compared with 76.9% and 79.8% for Channel Top-k. When the first, middle, and last MLP layers are approximated jointly, WikiText-2 perplexity remains within 0.06% and 0.54% of dense inference. Relative to expected-count Grouped Top-k, NORDER-II yields 82.6–126.4% relative accuracy gains with only 0.18–0.19% additional local analytic FLOPs. These results establish the significance of joint-state structure beyond atomic salience and point to a new paradigm for neural computation: dense layers can be viewed as structured probabilistic inference over combinatorial computational states. This framework opens a path toward adaptive, probability-mass-controlled computation in pretrained neural networks.

1 INTRODUCTION

Transformer language models perform dense computation over many internal factors, including attention heads, token interactions, MLP channels, channel groups, and layerwise paths. A common way to reduce this computation is to rank individual factors, for example by selecting the top-k attention links or the top-k MLP channels. To connect this familiar ranking view to our state-space formulation, we distinguish two resolutions: an *atomic state* describes the singleton computation associated with one factor, whereas a *joint state* specifies, for every factor in a set, whether its contribution is included or excluded. The atomic view captures individual contributions, but many module outputs depend on several factors jointly. The resulting joint state space grows combinatorially: assigning an include/exclude decision to each of G factors yields 2^G joint states.

Before introducing joint-state traversal, we first test whether the simpler atomic view is sufficient. As a diagnostic study, we examine probability-mass truncation over singleton attention links and feed-forward channels in GPT-2 models (Radford et al., 2019) (Appendix F). The two sublayers behave differently: attention admits aggressive atomic truncation, whereas feed-forward layers require much broader singleton coverage to recover dense behavior. This asymmetry suggests that MLP computation is not well explained by a small list of individually dominant channels. It motivates the central move in this paper: from ranking atomic states toward probability-ordered traversal of *combinatorial* neural state spaces.

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.

This paper studies the following question: *Can dense neural computation be reformulated as probability-ordered traversal over a combinatorial state space?* We answer it in three steps. First, we construct an auxiliary distribution over binary inclusion states, each specifying which component contributions are retained in a masked computation; the full expectation exactly recovers the dense module output. These states are analytical constructs, not latent or model-native alternatives executed by the transformer. Second, we introduce **NORDER-II**, a neural adaptation of ORDER-II state enumeration (Li & Silvester, 1984; Lam & Li, 1986), to generate complete binary configurations in decreasing joint probability without sorting all 2^G states. Third, we truncate the ordered expectation and evaluate the resulting transformer MLP approximation.

The key distinction from atomic ranking lies both in the object being ordered and in how the retained computations are combined. Atomic ranking orders individual contributions and retains a single subset. NORDER-II instead orders *joint states* (complete assignments specifying whether each factor's contribution is included or excluded) by joint probability and sums their probability-weighted masked outputs. It extends the prefix until its cumulative mass reaches τ or the state cap B is reached, allowing us to measure how many complete configurations are needed to cover a target mass.

We evaluate local transformer MLP approximation using channel groups with input-dependent Bernoulli inclusion probabilities. Our controls isolate probability ordering (uniform random traversal), atomic salience (Channel Top-k), and group salience (Grouped Top-k, which directly sums the highest-probability groups). For local MLP approximation, NORDER-II at a budget of 2048 states reaches approximation accuracies of 92.4% on Llama and 93.8% on Gemma, compared with 76.9% and 79.8% for Channel Top-k at its 2048-channel setting, the highest-accuracy baseline reported. When the first, middle, and last MLP layers are approximated jointly at the same state budget, the resulting perplexities are only 0.06% and 0.54% above the respective dense models. At the same state budget, NORDER-II achieves 82.6–126.4% relative accuracy gains over expected-count Grouped Top-k with a negligible 0.18–0.19% increase in local analytic FLOPs, including shared group-contribution computation. These results show that a small probability-ordered prefix can faithfully approximate dense computation despite representing only a small fraction of an exponentially large state space.

Our novelty is the combination of an exact probabilistic representation of pretrained Transformer MLP computation with probability-ordered traversal of complete joint states. Our contributions are:

- **Representation.** We construct an auxiliary binary state space for grouped transformer MLPs and prove that its full expectation exactly recovers the dense MLP output.
- **Enumeration.** We develop NORDER-II to enumerate joint states in decreasing probability without sorting all 2^G configurations. Generating a prefix of B states costs $O(G \log G + B \log B)$, separate from computing the group contributions.
- **Local Evaluation.** We use atomic attention and feed-forward diagnostics to motivate the transition from singleton ranking to joint MLP states. We then characterize local MLP approximation against random traversal, Channel Top-k, and Grouped Top-k, with sensitivity analyses for grouping and the probability-transform scale.
- **Downstream Evaluation.** We evaluate next-token likelihood when the first, middle, and last MLP layers are approximated jointly.

2 Related Work

Sparse and efficient neural inference. A large body of work aims to reduce inference cost by inducing sparsity in neural networks. Traditional approaches include unstructured weight pruning (Han et al., 2015) and structured pruning of channels, filters, or units to improve hardware efficiency (He et al., 2017). In transformers, prior work has shown that many attention heads are redundant and can be removed with little impact on performance (Michel et al., 2019; Voita et al., 2019). In contrast, probability-ordered traversal truncates an auxiliary expectation at a probability-mass threshold rather than a fixed sparsity budget. The retained prefix length can therefore adapt across inputs and layers to the concentration of the constructed state distribution.

Conditional computation and mixture models. Conditional computation methods, such as mixture-of-experts models, dynamically route inputs to subsets of parameters (Shazeer et al., 2017). These models explicitly represent computation as a probabilistic mixture over experts and typically

require architectural changes and specialized training procedures. Our framework is related to these methods but differs in key ways. First, it does not introduce new routing mechanisms or parameters; it constructs an auxiliary distribution from pretrained-model activations. Second, it targets internal neural state spaces rather than learned expert modules. Finally, NORDER-II supports analysis- or inference-time enumeration, whereas conditional computation methods usually require retraining.

Discrete subset sampling. SIMPLE studies differentiable fixed-cardinality subset sampling during learning, using discrete k-subsets in the forward pass and exact marginals for gradients (Ahmed et al., 2023). Our framework instead uses complete, variable-cardinality Bernoulli configurations; NORDER-II orders them to a mass target or state budget, without gradients or a cardinality constraint. Appendix G gives the classical enumeration background and further comparisons with input-dependent sparsity, sparse attention, and analyses of effective model usage.

3 METHODOLOGY

3.1 PROBLEM SETTING AND PROBABILISTIC STATE SPACES

We adopt this perspective by constructing an *auxiliary probabilistic state space* for a neural module. Given an input or intermediate activation x, let $\mathcal{S}$ be a finite set of deterministic masked computations. A state $s \in \mathcal{S}$ specifies which constructed factors contribute and induces an output $f(x, s)$. These states need not be latent alternatives or model-native states executed by the pretrained network. We associate them with an input-dependent mass $P_\theta(s \mid x)$, $s \in \mathcal{S}$, satisfying $\sum_{s\in\mathcal{S}} P_\theta(s \mid x) = 1$.

The state computation and probability model are constructed so that the dense module output has the exact representation

$$y(x) \;=\; \sum_{s\in\mathcal{S}} P_\theta(s \mid x)\, f(x, s). \tag{1}$$

Our MLP construction uses complete binary configurations over channel groups, with heuristic Bernoulli salience scores and importance-corrected state computations. The inclusion probabilities determine traversal order and finite-prefix quality, while the importance correction preserves the exact full expectation. Appendix C provides controlled illustrations of this abstraction, including an exact comparison between NORDER-II and exhaustive enumeration and a trained one-hidden-layer network with group-level binary states.

3.2 TRANSFORMER MLP INSTANTIATION

We instantiate the auxiliary states inside a gated transformer MLP. Let

$$h(x) = \phi(W_{\text{gate}}x) \odot W_{\text{up}}x. \tag{2}$$

We partition its intermediate coordinates into G nearly equal contiguous index sets $\{I_g\}_{g=1}^{G}$ and define

$$c_g(x) = W_{\text{down}}[:, I_g]h_{I_g}(x), \qquad y(x) = \sum_{g=1}^{G} c_g(x). \tag{3}$$

For independent $z_g \sim \text{Bernoulli}(\pi_g)$, define the importance-corrected masked output

$$F(x, z) = \sum_{g=1}^{G} \frac{z_g}{\pi_g} c_g(x), \tag{4}$$

where the independent joint-state probability is

$$P(z \mid x) = \prod_{g=1}^{G} \pi_g^{z_g}(1 - \pi_g)^{1-z_g}. \tag{5}$$

Theorem 3.1 (Representation equivalence)**.** *For any $\pi_g \in (0, 1]$, the auxiliary state computation in Eq.* (4) *satisfies $\mathbb{E}_z[F(x, z)] = y(x)$.*

Proof. By linearity of expectation and $\mathbb{E}[z_g] = \pi_g$, $\mathbb{E}_z[F(x,z)] = \sum_{g=1}^{G} \frac{\mathbb{E}[z_g]}{\pi_g} c_g(x) = \sum_{g=1}^{G} c_g(x) = y(x)$. □

We construct group probabilities from contribution norms. Let $\mu_n(x)$ and $\sigma_n(x)$ be the across-group mean and standard deviation. The probability model is

$$\begin{aligned} n_g(x) &= \|c_g(x)\|_2, \\ \widetilde{n}_g(x) &= \frac{n_g(x) - \mu_n(x)}{\max(\sigma_n(x), \epsilon)}, \\ \pi_g(x) &= \pi_{\min} + (1 - 2\pi_{\min})\mathrm{sigmoid}(\alpha \widetilde{n}_g(x) + \beta). \end{aligned} \tag{6}$$

This monotone transform uses $\alpha = 2$, $\beta = 0$, $\pi_{\min} = 0.05$, and $\epsilon = 10^{-8}$, yielding $\pi_g = 0.05 + 0.90\,\mathrm{sigmoid}(2\widetilde{n}_g)$. The scale α controls how strongly relative contribution size affects inclusion, while $\beta = 0$ assigns probability 0.5 to a group whose contribution norm equals the across-group mean. The floor $\pi_{\min}$ keeps probabilities away from zero and one and limits importance weights; ϵ prevents division by zero or near-zero standard deviations. This construction is a heuristic, input-adaptive salience model rather than a model-native probability.

The identity holds for every positive choice of π_g: grouping and the probability transform affect traversal order and finite-prefix efficiency, not full-expectation correctness. The variables z_g define an auxiliary representation rather than latent states executed by the transformer. Appendix A works through all eight masks for three MLP groups, making the importance correction and exact expectation explicit.

3.3 Probability-Mass Truncation

The approximation restricts Eq. (1) to a probability-ordered prefix. The threshold τ denotes cumulative probability mass, not a percentile of states or a fixed retained fraction. Unlike fixed Top-k selection, it adapts the number of retained states to the input-dependent mass distribution.

Given a mass threshold $\tau \in (0, 1]$, the truncation selects the shortest probability-ordered prefix $\widehat{\mathcal{S}} \subset \mathcal{S}$ such that

$$\sum_{s \in \widehat{\mathcal{S}}} P_\theta(s \mid x) \;\geq\; \tau. \tag{7}$$

The dense expectation in Eq. (1) is then approximated by

$$\widehat{y}(x) \;=\; \sum_{s \in \widehat{\mathcal{S}}} P_\theta(s \mid x)\, f(x,s). \tag{8}$$

Theorem 3.2 (Mass truncation bound)**.** *Assume that $\|f(x,s)\| \leq B(x)$ for all $s \in \mathcal{S}$. Let $\widehat{\mathcal{S}} \subset \mathcal{S}$ satisfy $\sum_{s \in \widehat{\mathcal{S}}} P_\theta(s \mid x) \geq \tau$. Then the approximation in Eq.* (8) *obeys*

$$\|y(x) - \widehat{y}(x)\| \;\leq\; (1 - \tau)\, B(x). \tag{9}$$

The proof is given in Appendix B.

This probability-mass view leaves model parameters unchanged and alters only the aggregation of the auxiliary masked computations. It serves as an approximation framework and as a diagnostic lens for where the constructed state distribution concentrates.

For a probability-ordered prefix $\mathcal{P}$, NORDER-II uses

$$\hat{y}_{\mathcal{P}}(x) = \sum_{z \in \mathcal{P}} P(z \mid x) F(x,z). \tag{10}$$

The same estimator admits a coefficient-collapsed computational form. Substituting Eq. (4) into Eq. (10) and exchanging the finite sums gives

$$\begin{aligned} \hat{y}_{\mathcal{P}}(x) &= \sum_{g=1}^{G} a_g(\mathcal{P}, x) c_g(x), \\ a_g(\mathcal{P}, x) &= \frac{1}{\pi_g} \sum_{z \in \mathcal{P}} P(z \mid x) z_g. \end{aligned} \tag{11}$$

Thus the retained prefix first induces scalar coefficients over groups, after which the G contribution vectors are combined only once. After the group contributions $\{c_g(x)\}_{g=1}^{G}$ are available, aggregation costs $O(BG + Gd)$ with a dense scan over a B-state prefix and $O(A(\mathcal{P}) + Gd)$ with sparse active-state accumulation, where $A(\mathcal{P}) = \sum_{z\in\mathcal{P}} \sum_g z_g$ is the number of active group occurrences in the prefix and d is the output dimension. This avoids constructing a separate d-dimensional masked output for every retained state. The full state space recovers $y(x)$ exactly; for a truncated prefix, omitted mass controls the worst-case error under the bounded-output condition in Theorem 3.2.

3.4 Joint Binary Neural State Spaces

The main object of this work is a joint binary state space over neural factors. For a module and input x, let there be G binary factors $z = (z_1, \ldots, z_G), \qquad z_g \in \{0, 1\}$. In our transformer MLP experiments, each factor corresponds to a group of intermediate channels. More generally, factors may correspond to heads, blocks, channel groups, or other module-local components.

For each factor, let $\pi_g = P(z_g = 1 \mid x)$. The most likely per-factor value defines the base mode $b_g = \arg\max_{v\in\{0,1\}} P(z_g = v \mid x)$. Any joint state can be represented as a flip subset $S \subseteq \{1, \ldots, G\}$: factors in S are flipped away from their base values, and all other factors remain at the base mode. Define $p_g^{\text{base}} = \max(\pi_g, 1 - \pi_g), \quad p_g^{\text{alt}} = \min(\pi_g, 1 - \pi_g)$, and $r_g = \frac{p_g^{\text{alt}}}{p_g^{\text{base}}}, \qquad \lambda_g = -\log r_g$. The base state has no flips. A flip subset S has probability

$$P(S \mid x) = P(\text{base} \mid x) \exp\left(-\sum_{g\in S} \lambda_g\right). \tag{12}$$

Thus ordering joint states by probability is equivalent to ordering flip subsets by the additive penalty $\sum_{g\in S} \lambda_g$.

3.5 NORDER-II Enumeration

Algorithm 1 NORDER-II for Joint Binary Neural States

Require: Factor probabilities $\pi_1, \ldots, \pi_G$, target mass τ, state cap B
Ensure: Probability-ordered joint-state prefix
1: Compute base modes b_g, penalties λ_g, and sorted order of factors
2: Initialize prefix $\mathcal{P} \leftarrow \emptyset$, cumulative mass $m \leftarrow 0$
3: Initialize heap with the empty flip subset $S = \emptyset$
4: **while** heap is nonempty and $m < \tau$ and $|\mathcal{P}| < B$ **do**
5: Pop the subset S with smallest penalty
6: Convert S to its binary state and append it to $\mathcal{P}$
7: Add its probability from Eq. (12) to m
8: Push valid ORDER-II children of S that have not been visited
9: **end while**
10: **return** $\mathcal{P}$

NORDER-II enumerates flip subsets in increasing additive penalty. It first sorts the factor penalties λ_g increasingly. Starting from the empty subset, it maintains a priority queue of candidate subsets. For a subset whose largest sorted index is j, the ORDER-II child rule proposes two next subsets: replacing j with $j + 1$, and expanding the subset with $j + 1$. This gives a compact heap traversal of the ordered subset lattice.

Here τ targets cumulative joint-state probability mass, not a state percentile or a fixed retained fraction. Sorting the G flip penalties costs $O(G \log G)$. Each of at most B generated states requires one heap removal and at most two insertions in a heap of size $O(B)$, giving $O(B \log B)$ traversal cost and total enumeration overhead $O(G \log G + B \log B)$. Candidate and visited-state storage is proportional to the generated prefix; explicit G-bit masks require $O(BG)$ bits. These costs exclude computing the group contributions and probabilities, which uses every group in the present instantiation.

NORDER-II differs from atomic ranking in two ways. First, it ranks complete binary states rather than individual factors. Second, it is controlled by a target mass and a state cap, so the generated prefix reflects both the concentration of the probability distribution and the computational budget.

4 Experimental Setup

We evaluate NORDER-II through local transformer MLP approximation and downstream next-token likelihood. The experiments test whether probability-ordered joint states provide a better approximation than random traversal, atomic channel ranking, and deterministic group-salience controls.

4.1 Model and Data

We report results on Llama-3.2-1B (Meta, 2024) and Gemma-2-2B (Gemma Team, 2024). We evaluate a deterministic set of validation examples from the WikiText-2 validation split (Merity et al., 2016). For each example, we evaluate the final non-padding token. For Llama, we inspect layers 0, 8, and 15; for Gemma, we inspect layers 0, 13, and 25. These correspond to early, middle, and late MLP layers in each model. For the downstream evaluation, we score 4,096 WikiText-2 next-token targets while jointly replacing the MLP outputs at these three layers. We compare each model only with its own dense reference because Llama and Gemma use different tokenizers.

4.2 Joint-State Setup and Baselines

For NORDER-II, we partition the MLP intermediate dimension into $G \in \{8, 12, 16\}$ contiguous channel groups. We evaluate target masses $\tau \in \{0.80, 0.90, 0.95, 0.99\}$ and state caps $B \in \{16, 32, 64, 128, 256, 512, 1024, 2048, 4096\}$. Unless specified otherwise, tabulated comparisons use $\tau = 0.95$. We use $G = 12$ as the primary setting because it is the most interpretable regime under this fixed maximum state budget: the joint state space has $2^{12} = 4096$ states, large enough that random traversal is nontrivial but small enough that NORDER-II can reach the target mass. By contrast, $G = 8$ has only 256 states, so random traversal enumerates the full state space at budgets above 256, while $G = 16$ has 65,536 states and is used to assess scaling with a larger joint state space.

We compare against random, atomic-channel, and group-salience baselines. **Random joint states** samples unique binary states uniformly from the 2^G joint state space and uses the same local estimator as NORDER-II. **Channel Top-k** is an atomic baseline over the original MLP channels, not over NORDER-II groups. For each token, it ranks intermediate channels by $|h_j(x)| \|W_{\text{down}}[:, j]\|_2$. It keeps the top k channels. We evaluate $k \in \{64, 128, 256, 512, 1024, 2048\}$ out of the model's intermediate MLP dimension (8192 for Llama and 9216 for Gemma). **Grouped Top-k** uses the same $G = 12$ nearly equal contiguous groups, contributions $c_g(x)$, and probabilities $\pi_g(x)$ as NORDER-II at every evaluated token and layer. Let $T_m(x) = \mathrm{TopM}_g\{\pi_g(x)\}$ contain the m largest-probability groups. The deterministic baseline returns

$$\hat{y}_{\mathrm{GroupTop}-k}(x) = \sum_{g \in T_m(x)} c_g(x). \tag{13}$$

We evaluate $m = 1$ and the expected-count rule

$$m(x) = \mathrm{clip}\left(\left\lfloor \sum_{g=1}^{G} \pi_g(x) + \frac{1}{2} \right\rfloor, 1, G - 1 \right). \tag{14}$$

In the primary $G = 12$ setting, this clips m to $[1, 11]$; the upper bound prevents the control from becoming the full dense MLP. Because Eq. (6) is monotone in $\|c_g(x)\|_2$, ranking by π_g is equivalent to ranking by contribution norm. Equation (13) is deterministic and uncorrected: the probabilities rank the groups and determine m, but do not scale the retained contributions. By contrast, an importance-corrected expected mask is exactly the full auxiliary expectation in Theorem 3.1 and therefore reproduces $y(x)$. We omit it as an uninformative approximation control.

4.3 METRICS

We report approximation accuracy $A = 1 - \|\hat{y} - y\|_2 / \|y\|_2$, expressed as a percentage and measuring agreement with the dense MLP output. Relative accuracy gains are $100(A/A_{\text{baseline}} - 1)\%$. We also report cosine similarity, cumulative probability mass covered by the joint-state prefix, and whether the target mass was reached within the evaluated state budget. The downstream evaluation reports next-token negative log-likelihood (NLL) and perplexity.

5 RESULTS

5.1 MAIN COMPARISON IN THE PRIMARY JOINT-STATE SETTING

Table 1 combines the main baselines in the primary $G = 12$ regime. The comparison distinguishes the unit being selected: NORDER-II and random traversal select complete joint states, Channel Top-k selects individual MLP channels, and Grouped Top-k selects groups. This distinction matters, but the empirical pattern is direct. At budget 512, NORDER-II reaches approximation accuracy of 78.6% on Llama and 72.3% on Gemma, compared with 12.5%/12.6% for random joint states and 52.4%/57.9% for Channel Top-k. At budget 2048, NORDER-II reaches the $\tau = 0.95$ mass target with approximation accuracy of 92.4% on Llama and 93.8% on Gemma, compared with 50.0%/50.2% for random joint states and 76.9%/79.8% for Channel Top-k.

Table 1: Local transformer MLP comparisons at $G = 12$ and $\tau = 0.95$. Budget denotes states for joint-state methods and channels for Channel Top-k; Grouped Top-k has no joint-state budget. Acc. is approximation accuracy; Cos. is cosine similarity. Mass is cumulative joint-state prefix mass and is not defined for Channel or Grouped Top-k. Ch./grp. frac. is the retained channel or group fraction.

Model	Budget	Method	Unit	Acc. (%) ↑	Cos.	Mass	Ch./grp. frac.
Llama	512	NORDER-II	joint states	78.6	0.994	0.843	–
Llama	512	Random	joint states	12.5	0.989	0.125	–
Llama	512	Ch. Top-k	channels	52.4	0.869	–	0.063
Llama	2048	NORDER-II	joint states	92.4	0.999	0.950	–
Llama	2048	Random	joint states	50.0	0.998	0.501	–
Llama	2048	Ch. Top-k	channels	76.9	0.969	–	0.250
Llama	–	Grp. Top-k (exp.)	groups	40.8	0.813	–	0.461
Llama	–	Grp. Top-k ($m = 1$)	groups	9.6	0.451	–	0.083
Gemma	512	NORDER-II	joint states	72.3	0.995	0.768	–
Gemma	512	Random	joint states	12.6	0.995	0.126	–
Gemma	512	Ch. Top-k	channels	57.9	0.909	–	0.056
Gemma	2048	NORDER-II	joint states	93.8	1.000	0.950	–
Gemma	2048	Random	joint states	50.2	0.999	0.503	–
Gemma	2048	Ch. Top-k	channels	79.8	0.979	–	0.222
Gemma	–	Grp. Top-k (exp.)	groups	51.4	0.885	–	0.441
Gemma	–	Grp. Top-k ($m = 1$)	groups	20.9	0.642	–	0.083

The Grouped Top-k rows isolate deterministic group salience using the same groups and probabilities. Relative to expected-count Grouped Top-k, NORDER-II increases approximation accuracy from 40.8% to 78.6% on Llama and from 51.4% to 72.3% on Gemma at $B = 512$, relative gains of 92.5% and 40.7%. At $B = 2048$, accuracies reach 92.4% and 93.8%, relative gains of 126.4% and 82.6%; retaining only the highest-probability group yields 9.6% and 20.9%. Thus probability-ordered joint-state aggregation adds approximation value beyond group salience alone.

5.2 DOWNSTREAM NEXT-TOKEN EVALUATION

At $B = 512$, NORDER-II yields perplexities 10.707 on Llama and 37.961 on Gemma, about 5% above their dense references (10.168 and 36.236), whereas expected-count Grouped Top-k reaches 16.832 and 58.552 (66% and 62% above dense). At $B = 2048$, NORDER-II reaches 10.175 and 36.430, only 0.06% and 0.54% above the corresponding dense models. These are within-model comparisons because tokenization differs.

Table 2: Next-token NLL and perplexity on 4,096 WIKITEXT-2 targets when the first, middle, and last MLP layers are approximated jointly. Grouped Top-k is the expected-count baseline.

Model	Method	B	NLL	PPL
Llama	Dense	–	2.3193	10.168
Llama	NORDER-II	512	2.3709	10.707
Llama	NORDER-II	2048	2.3199	10.175
Llama	Grp. Top-k	–	2.8233	16.832
Gemma	Dense	–	3.5900	36.236
Gemma	NORDER-II	512	3.6366	37.961
Gemma	NORDER-II	2048	3.5954	36.430
Gemma	Grp. Top-k	–	4.0699	58.552

5.3 STATE BUDGETS TRACK PROBABILITY MASS

For $G = 12$, NORDER-II reaches $\tau = 0.95$ consistently across the selected layers: about 1098, 1153, and 1049 states for Llama layers 0, 8, and 15, and about 1366, 1704, and 1067 states for Gemma layers 0, 13, and 25. These are roughly 26–28% of the full 2^{12} state space for Llama and 26–42% for Gemma. For $G = 16$, 4096 states constitute 6.25% of the 65,536-state space, but coverage depends on probability concentration rather than state fraction alone. At this budget, NORDER-II covers cumulative mass 0.804 for Llama and 0.686 for Gemma, showing that these distributions remain too diffuse to reach $\tau = 0.95$ within the evaluated budget. Thus $G = 12$ demonstrates successful joint-state traversal, whereas $G = 16$ exposes the scaling pressure of larger, less concentrated combinatorial spaces. Appendix E reports the complete $G = 16$ results.

5.4 SENSITIVITY TO GROUPING AND PROBABILITY SCALE

Across models and $G \in \{8, 12, 16\}$, ten independently seeded, size-preserving random partitions change mean approximation accuracy by at most 0.83 percentage points from contiguous grouping. Table 5 in Appendix D reports sensitivity to the probability-transform scale at $B = 2048$; $\alpha = 2$ is best on both models. The appendix also specifies the partition and probability-scale protocols used to test robustness to these construction choices.

5.5 LOCAL FLOP ACCOUNTING

Table 3 reports local analytic per-token-layer FLOPs at $G = 12$, including the shared group-contribution computation. At $B = 2048$, NORDER-II gains 126.4% and 82.6% in approximation accuracy over expected-count Grouped Top-k on Llama and Gemma, respectively, with 0.19% and 0.18% more local FLOPs. The small aggregation increment follows from Eq. (11): retained states reuse the group contribution vectors, adding mainly scalar coefficient accumulation before one final group-vector blend.

Table 3: Local per-token-layer FLOP accounting at $G = 12$. Total FLOPs include shared group-contribution computation and method-specific aggregation. Relative gains and FLOP increases use expected-count Grouped Top-k as the within-model baseline; mass is cumulative joint-state prefix mass.

Model	Method	Setting	Acc. (%) ↑	Mass	Total FLOPs (M)	Agg. FLOPs (k)	Relative acc. gain (%)	FLOP increase (%)
Llama	Grp. Top-k	exp. count	40.8	–	33.564	9.3	–	–
Llama	NORDER-II	$B = 512$	78.6	0.843	33.614	59.4	+92.5	+0.15
Llama	NORDER-II	$B = 2048$	92.4	0.950	33.628	73.5	+126.4	+0.19
Gemma	Grp. Top-k	exp. count	51.4	–	42.477	9.9	–	–
Gemma	NORDER-II	$B = 512$	72.3	0.768	42.533	65.3	+40.7	+0.13
Gemma	NORDER-II	$B = 2048$	93.8	0.950	42.553	86.1	+82.6	+0.18

6 DISCUSSION

Joint-state aggregation adds value beyond salience. Random traversal tests whether probability order matters, while atomic Channel Top-k tests whether singleton channel scores suffice. Grouped Top-k holds the auxiliary groups and inclusion probabilities fixed but replaces the probability-weighted joint-state prefix with one uncorrected deterministic sum of the most salient groups. Relative to expected-count Grouped Top-k, NORDER-II increases approximation accuracy by 92.5% and 40.7% at $B = 512$ and by 126.4% and 82.6% at $B = 2048$ on Llama and Gemma, respectively. This comparison isolates the approximation value of joint-state aggregation; it is not a compute-matched speed comparison.

Exact representation and finite-prefix sensitivity. The full importance-corrected expectation equals the dense MLP output for any positive inclusion probabilities. Grouping, contribution norms, the sigmoid transform, and independent Bernoulli factors are practical choices that affect traversal order and finite-prefix quality, not exactness. Random-partition sensitivity of at most 0.83 percentage points suggests that the result is not tied to contiguous groups, while $\alpha = 2$ performing best among the tested scales shows that probability scale still matters for truncated prefixes. Probabilities near zero or one concentrate mass, whereas those near 0.5 produce diffuse state spaces. Thus mass concentration, not G alone, determines the required prefix: $G = 12$ reaches the 0.95 target within the evaluated budget, whereas $G = 16$ does not.

Downstream evidence in the measured setting. The three-layer intervention tests whether local approximations remain accurate when applied jointly across the network. At $B = 2048$, perplexity is 0.06% above dense for Llama and 0.54% for Gemma and substantially lower than expected-count Grouped Top-k. This supports near-dense next-token likelihood in the measured setting and motivates evaluating open-ended generation and broader layer interventions.

Enumeration and aggregation cost. The $O(G \log G + B \log B)$ bound covers state enumeration after the group probabilities are available. The cost accounting in Section 5.5 shows that NORDER-II adds less than 0.2% total local FLOPs over expected-count Grouped Top-k in the measured settings. This small overhead suggests that optimized contribution scoring, contribution reuse, and probability-ordered aggregation can be developed as complementary implementation directions.

Broader research directions. Two extensions would let cumulative probability mass determine how much of a combinatorial space is retained. In structured neural computation graphs, states could represent valid sparse subgraphs, adapting computation to probability concentration. This requires graph-valid state construction and low-overhead scoring. In autoregressive decoding, states could represent multi-token continuations; retaining them to a cumulative-mass target could provide an adaptive alternative to fixed-width beam search. Both directions require probability models and enumeration methods that handle dependencies and structural constraints.

7 CONCLUSION

We studied whether dense neural computation can be reformulated as probability-ordered traversal over a combinatorial state space. For additive modules, auxiliary Bernoulli masks with importance correction give an exact expectation equal to the dense output. NORDER-II traverses this state space by enumerating flip subsets in increasing additive log-penalty without sorting all configurations.

In transformer MLP experiments, NORDER-II improves local approximation over random joint traversal, atomic Channel Top-k, and deterministic Grouped Top-k controls. At a sufficiently large state budget, jointly approximated MLP layers retain near-dense WikiText-2 perplexity, supporting this representation and approximation framework. Future work will pursue lower-overhead scoring, richer state models, adaptive grouping, and broader evaluation of structured sparse execution and cumulative-mass decoding.

AI USE STATEMENT

AI assistants supported literature search, code implementation, interpretation of results, drafting, and language revision. The authors are responsible for the content of the submitted manuscript.

ETHICS STATEMENT

This work introduces NORDER-II, a method for analyzing neural computation through probability-ordered traversal of joint internal state spaces. The main immediate impact is scientific: the method provides a tool for measuring how probability mass is distributed over combinatorial neural states and for comparing joint-state structure against atomic sparsity baselines.

NORDER-II does not require retraining and does not alter the underlying model parameters. The experiments include local MLP approximation and downstream perplexity after approximating selected layers. Future optimized implementations may broaden the practical applications of probability-ordered neural computation.

The method inherits the behavioral risks of the underlying language models being analyzed. It may improve interpretability and diagnostics by exposing which joint states dominate local computations, but it should not be interpreted as a safety guarantee or as a mechanism for removing harmful model behavior.

REPRODUCIBILITY STATEMENT

Sections 3 and 4 specify the joint-state construction, enumeration procedure, models, dataset split, evaluated layers and target count, baselines, budgets, and metrics; Appendix D gives the sensitivity protocols. Appendix C reports checks against exhaustive state ordering and a controlled network experiment; Theorem 3.2 gives the assumptions of the mass-truncation bound, and Appendix B gives its proof.

REFERENCES

Kareem Ahmed, Zhe Zeng, Mathias Niepert, and Guy Van den Broeck. SIMPLE: A gradient estimator for k-subset sampling. In *International Conference on Learning Representations*, 2023.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6541–6549, 2017.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with Performers. In *International Conference on Learning Representations*, 2021.

Gemma Team. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Ankit Gupta, Guy Dar, Shaya Goodman, David Ciprut, and Jonathan Berant. Memory-efficient transformers via top-k attention. In *Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing*, pp. 39–52. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.sustainlp-1.5.

Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1389–1397, 2017.

Le Hou, Richard Yuanzhe Pang, Tianyi Zhou, Yuexin Wu, Xinying Song, Xiaodan Song, and Denny Zhou. Token dropping for efficient BERT pretraining. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3774–3784. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.262.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2020.

Y. F. Lam and Victor O. K. Li. An improved algorithm for performance analysis of networks with unreliable components. *IEEE Transactions on Communications*, 34(5):496–497, 1986.

Victor O. K. Li and John A. Silvester. Performance analysis of networks with unreliable components. *IEEE Transactions on Communications*, 32(10):1105–1110, 1984.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Meta. Llama 3.2: Revolutionizing edge AI and vision with open, customizable models, 2024. URL `https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/`.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019. URL `https://cdn.openai.com/better-language-models/language-models.pdf`.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems*, volume 34, pp. 13937–13949, 2021.

Simone Scardapane, Alessandro Baiocchi, Alessio Devoto, Valerio Marsocci, Pasquale Minervini, and Jary Pomponi. Conditional computation in neural networks: Principles and research trends. *Intelligenza Artificiale*, 18(1):175–190, 2024. doi: 10.3233/IA-240035.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5797–5808. Association for Computational Linguistics, 2019. doi: 10.18653/v1/P19-1580.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big Bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, volume 33, pp. 17283–17297, 2020.

APPENDIX: SUPPLEMENTARY MATERIAL

A WORKED EXAMPLE: FROM THREE MLP GROUPS TO JOINT STATES

Partition the MLP hidden coordinates into three disjoint groups I_1, I_2, I_3, and let $c_g(x)$ denote the output contribution of the coordinates in I_g. The dense output is $y(x) = c_1(x) + c_2(x) + c_3(x)$. Choose arbitrary inclusion probabilities $\pi_g \in (0, 1]$ and independent $z_g \sim \text{Bernoulli}(\pi_g)$. Written as $z_1 z_2 z_3$, the eight auxiliary masks are

$$\{000, 001, 010, 011, 100, 101, 110, 111\}.$$

For example, $z = (1, 0, 1)$ produces

$$F(x, (1, 0, 1)) = \frac{c_1(x)}{\pi_1} + \frac{c_3(x)}{\pi_3}. \tag{15}$$

Summing over all eight masks gives

$$\mathbb{E}_z[F(x, z)] = \sum_{g=1}^{3} \frac{\mathbb{E}[z_g]}{\pi_g} c_g(x) = y(x). \tag{16}$$

The full auxiliary expectation is therefore exact; approximation begins only when it is truncated to a probability-ordered prefix. Appendix C instantiates the same construction in a trained one-hidden-layer network.

B APPROXIMATION GUARANTEE

This truncation admits a simple worst-case guarantee: under a bounded-contribution assumption, the approximation error is at most proportional to the omitted probability mass $(1 - \tau)$ (Theorem 3.2).

Probability-mass truncation admits a simple guarantee that directly links probability mass coverage to approximation error. The bound in Theorem 3.2 formalizes this as a controlled truncation of an expectation.

Proof of Theorem 3.2. By definition,

$$y(x) - \widehat{y}(x) = \sum_{s \notin \widehat{\mathcal{S}}} P_\theta(s \mid x)\, f(x, s).$$

Applying the triangle inequality,

$$\|y(x) - \widehat{y}(x)\| \le \sum_{s \notin \widehat{\mathcal{S}}} P_\theta(s \mid x)\, \|f(x, s)\|.$$

Using the uniform bound $\|f(x, s)\| \le B(x)$,

$$\|y(x) - \widehat{y}(x)\| \le B(x) \sum_{s \notin \widehat{\mathcal{S}}} P_\theta(s \mid x) \le (1 - \tau)\, B(x),$$

because the selected set has probability mass at least τ. □

C CONTROLLED JOINT-STATE ILLUSTRATIONS

This appendix gives two controlled illustrations supporting the methodology in Section 3 and Appendix A. The first verifies that NORDER-II selects the same ordered high-probability states as exhaustive enumeration in independent binary state spaces. The second instantiates the one-hidden-layer FFN construction with a trained classifier and group-level hidden-unit states.

Correctness against exhaustive enumeration. We sample independent factor probabilities and compare NORDER-II with exhaustive enumeration for $G \in \{4, 8, 12, 16\}$ and $\tau \in \{0.80, 0.90, 0.95, 0.99\}$. Exhaustive enumeration first lists all 2^G states, sorts them by probability, and selects the smallest ordered set whose cumulative mass reaches τ. In all configurations, NORDER-II selects the same ordered set without sorting the full state space, preserves non-increasing probability order, produces no duplicate states, and stops as soon as the cumulative mass reaches τ. Table 4 reports representative results at $\tau = 0.95$. In the table, selected states denotes the smallest number of highest-probability states whose cumulative probability mass reaches τ. Because exhaustive sorting is an exact oracle for this finite state space, exact agreement with its selected ordered set verifies that NORDER-II returns the correct probability order and stopping point. Importantly, exhaustive enumeration is used only for validation in this controlled setting. **NORDER-II itself does not enumerate all 2^G states: it generates states in probability order and stops once the selected states reach the target mass.** Thus the selected-state count and fraction indicate how many states are actually visited before stopping.

Table 4: Controlled exhaustive-enumeration check at $\tau = 0.95$. NORDER-II selects the same ordered states as exhaustive sorting in all rows.

G	2^G	Selected	Frac.	Match
4	16	10	0.625	Yes
8	256	113	0.441	Yes
12	4096	1774	0.433	Yes
16	65536	12684	0.194	Yes

Trained one-hidden-layer network. We train a one-hidden-layer classifier on a two-class nonlinear classification problem and then freeze its parameters. Hidden units are partitioned into groups, inducing a 2^G joint binary state space over group inclusion patterns. The dense validation accuracy is 0.965. The purpose is not to establish a benchmark result, but to show that the FFN construction in Appendix A applies to learned hidden representations. We use *logit KL* to denote the KL divergence between the softmax output distributions induced by the dense and prefix-approximated logits; lower is better, and zero means that the distributions are identical. As the joint-state budget increases, probability-ordered prefixes reduce this divergence relative to random prefixes (Figure 1).

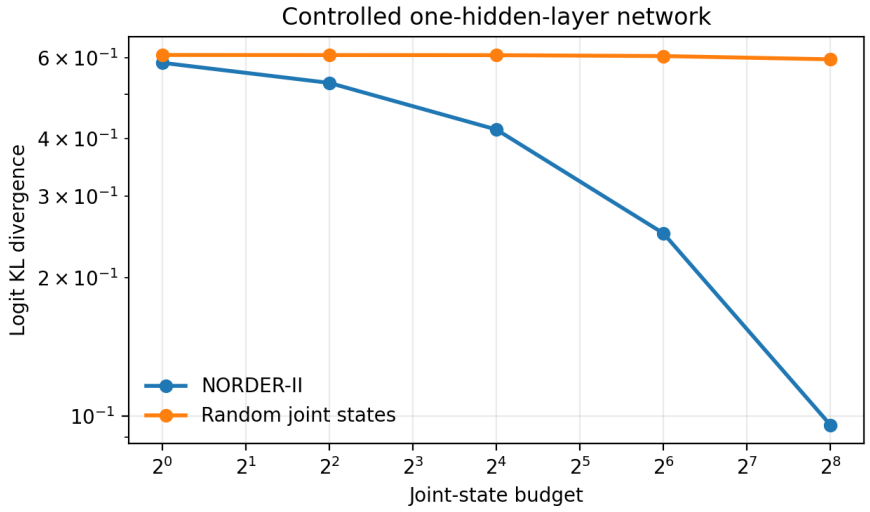

Figure 1: Controlled trained-network illustration. Probability-ordered joint prefixes reduce logit KL (lower is better) relative to random prefixes as the joint-state budget increases.

D Sensitivity Protocols and Results

We compare each contiguous partition with ten independently seeded, size-preserving random partitions for $G \in \{8, 12, 16\}$ under the same local evaluation protocol. At $B = 2048$, we also vary the probability-transform scale $\alpha \in \{1, 2, 4\}$ while holding $\beta = 0$ and $\pi_{\min} = 0.05$ fixed.

Table 5: Mean MLP-output approximation accuracy (%) at $B = 2048$ for probability-transform scales α. Higher is better.

Model	$\alpha = 1$	$\alpha = 2$	$\alpha = 4$
Llama	91.6	**92.4**	89.4
Gemma	92.8	**93.8**	91.3

E Joint-State Scaling Across Group Counts

Table 6 reports selected joint-state results for $G \in \{12, 16\}$. The $G = 12$ setting is the primary setting, where NORDER-II reaches $\tau = 0.95$ within the evaluated state budget. The $G = 16$ setting evaluates a larger joint state space: NORDER-II still strongly outperforms random traversal, but often does not reach $\tau = 0.95$ under the 4096-state budget.

Table 6: Selected joint-state local transformer MLP approximation results at $\tau = 0.95$ for $G = 12$ and $G = 16$. Acc. is approximation accuracy. Mass is covered joint probability mass.

Model	G	Budget	Method	Acc. (%) ↑	Mass
Llama	12	512	NORDER-II	78.6	0.843
Llama	12	512	Random	12.5	0.125
Llama	12	2048	NORDER-II	92.4	0.950
Llama	12	2048	Random	50.0	0.501
Llama	16	2048	NORDER-II	62.4	0.685
Llama	16	2048	Random	3.0	0.030
Llama	16	4096	NORDER-II	75.5	0.804
Llama	16	4096	Random	6.3	0.063
Gemma	12	512	NORDER-II	72.3	0.768
Gemma	12	512	Random	12.6	0.126
Gemma	12	2048	NORDER-II	93.8	0.950
Gemma	12	2048	Random	50.2	0.503
Gemma	16	2048	NORDER-II	50.7	0.553
Gemma	16	2048	Random	3.1	0.031
Gemma	16	4096	NORDER-II	64.8	0.686
Gemma	16	4096	Random	6.3	0.063

F Atomic-State Diagnostic Experiments

This appendix reports the atomic probability-mass diagnostics that motivate the transformer MLP study in the main text. We use GPT-2-XL on WikiText-2 for consistency with the main experiments' language-modeling corpus. These diagnostics are not the proposed method: they rank singleton attention links or singleton feed-forward channels. Their role is to show why MLP computation is a natural target for the joint-state formulation developed in the paper.

The diagnostic question is whether singleton states already provide an adequate probability-ordered description of transformer sublayers. If they do, then a small high-mass prefix of atomic states should recover dense language-modeling behavior. If they do not, then the sublayer may require broader or more structured state descriptions. The contrast between attention and feed-forward layers in Tables 7 and 8 is used in the main text as motivation for moving from atomic MLP-channel ranking to joint MLP group states.

An *atomic state* is a singleton computational unit inside a fixed sublayer. In self-attention, for each query position and head, an atomic state is one key–value interaction; its probability mass is the corresponding attention weight. Atomic attention truncation drops low-mass key–value terms while preserving the original attention weights for retained terms. In a feed-forward layer with hidden activation vector $a(x)$ and output projection W_2, the atomic state for channel j is the vector contribution $W_2[:, j]a_j(x)$, where $a_j(x)$ is the scalar hidden-channel activation and $W_2[:, j]$ is the corresponding output-projection column. Its mass is induced by the activation magnitude and normalized across channels.

For a normalized atomic mass vector p, the mass threshold τ specifies the smallest probability-ordered prefix $\mathcal{I}$ such that

$$\sum_{i \in \mathcal{I}} p_i \geq \tau. \tag{17}$$

The selected fraction reports $|\mathcal{I}|/N$, averaged over the relevant positions, heads, and layers. Thus τ controls covered probability mass, while the selected fraction measures how concentrated the atomic state distribution is. Atomic-Attn and Atomic-FFN apply this rule to attention links and feed-forward channels, respectively, selecting the smallest probability-ordered atomic prefix that reaches τ. Table 7 reports Dense and Atomic-Attn, while Table 8 reports Dense and Atomic-FFN. All rows are evaluated without retraining by patching the pretrained GPT-2-XL forward pass. We concatenate WikiText-2 evaluation texts, evaluate causal next-token negative log-likelihood in overlapping windows of length 1024 with stride 512, and report perplexity $\exp(\mathrm{NLL})$, NLL, and mean selected fraction.

Table 7: Atomic probability-mass truncation applied to self-attention layers in GPT-2-XL using WIKITEXT-2.

Method	τ	PPL	NLL	Sel. frac.
Dense	–	15.535	2.743	1.000
Atomic-Attn	0.50	18.614	2.924	0.015
Atomic-Attn	0.60	16.891	2.827	0.022
Atomic-Attn	0.70	16.182	2.784	0.032
Atomic-Attn	0.80	15.734	2.756	0.048
Atomic-Attn	0.90	15.545	2.744	0.081
Atomic-Attn	0.95	15.539	2.743	0.118
Atomic-Attn	0.99	15.533	2.743	0.216

Table 8: Atomic probability-mass truncation applied to feed-forward layers in GPT-2-XL using WIKITEXT-2.

Method	τ	PPL	NLL	Sel. frac.
Dense	–	15.535	2.743	1.000
Atomic-FFN	0.50	9312.123	9.139	0.356
Atomic-FFN	0.60	8441.614	9.041	0.438
Atomic-FFN	0.70	7670.426	8.945	0.539
Atomic-FFN	0.80	8252.224	9.018	0.682
Atomic-FFN	0.90	1686.188	7.430	0.867
Atomic-FFN	0.95	19.304	2.960	0.950
Atomic-FFN	0.99	15.695	2.753	0.993

Atomic-Attn recovers near-dense behavior after selecting a small fraction of attention links, whereas Atomic-FFN requires near-complete singleton coverage to recover dense-level perplexity. This contrast motivates the transition from atomic MLP-channel prefixes to probability-ordered traversal over joint MLP group states.

G EXTENDED RELATED WORK

Our probability-ordered joint-state framework applies to pretrained models across architectures. NORDER-II is its enumeration algorithm, which generates joint states in decreasing probability

without sorting the full state space. We relate the framework to work on efficient neural computation, sparsity, and probabilistic approximation, clarifying differences in scope and formulation.

G.1 Classical State Enumeration

Our work is inspired by classical *state enumeration* methods developed in reliability theory for analyzing large systems with unreliable components. In this literature, a system is modeled as a collection of components whose joint failure configurations define an exponentially large state space. Algorithms such as ORDER and ORDER-II enumerate system states in descending order of probability mass and approximate expected system performance by evaluating only the most probable states, terminating once sufficient probability mass has been covered (Li & Silvester, 1984; Lam & Li, 1986). These methods yield accurate approximations with strong empirical convergence properties while avoiding exhaustive enumeration.

G.2 Input-Dependent Sparsity

More recent methods induce input-dependent sparsity at inference time, including early-exit architectures and adaptive token pruning in transformers (Scardapane et al., 2024). DynamicViT, for example, removes redundant image patches in vision transformers (Rao et al., 2021). Related work on efficient pretraining drops less informative tokens from intermediate BERT layers and restores the full sequence at the final layer (Hou et al., 2022). These approaches typically rely on fixed hyperparameters such as a target sparsity ratio or confidence threshold.

G.3 Sparse and Approximate Attention

Efficient attention mechanisms have been extensively studied, especially for transformers with long sequences. Broadly, existing approaches reduce the quadratic cost of attention by either restricting the attention pattern a priori or by approximating the attention computation. Representative methods include: Locality-based attention: limiting each token to attend only to a neighborhood or block of tokens (or a fixed pattern). For example, sparse transformers use block-local and strided attention to achieve $O(n\sqrt{n})$ complexity instead of $O(n^2)$ (Child et al., 2019). The Longformer (Beltagy et al., 2020) and BigBird (Zaheer et al., 2020) models similarly employ sliding window attention (plus occasional global tokens) to handle long contexts efficiently. Low-rank or kernelized approximations: replacing the full softmax attention with a factorized or kernel-based computation. Linformer approximates the $n \times n$ attention matrix by a low-rank decomposition (projecting keys/values to a lower dimension) (Wang et al., 2020), while Performer and related linear transformers use random feature maps to linearize the softmax kernel (Choromanski et al., 2021). These methods trade off some accuracy for significantly reduced time and memory complexity (often linear in sequence length). Hashing-based attention: using randomized hashing to limit pairwise comparisons. The Reformer introduced locality-sensitive hashing (LSH) to group similar query-key pairs, achieving attention that is much more memory-efficient and faster on long sequences (Kitaev et al., 2020). By hashing tokens into buckets, each token attends only to others in the same bucket, reducing the effective sequence length in attention computation. Top-k sparse attention: selecting only the top-k scoring keys for each query to compute attention. In this approach, all attention scores are first computed, then all but the k largest are masked out, yielding a sparse attention matrix (Gupta et al., 2021). This top-k attention can serve as a plug-in replacement for vanilla attention, providing a highly accurate approximation while using only a fraction of the connections (Gupta et al., 2021). Notably, such methods can be applied at inference to pretrained models without additional training, as they simply truncate the softmax to its most significant entries. Existing sparse and approximate attention methods reduce computation by restricting or approximating collections of individual interactions. Our framework instead operates on complete inclusion/exclusion configurations over multiple factors, which NORDER-II orders by joint probability before truncation at a cumulative-mass target. The key distinction is therefore between operating on atomic interactions or prescribed patterns and traversing a probability-ranked space of complete joint configurations.

G.4 Analysis of Effective Model Usage

Several recent works analyze how much of a neural network is effectively utilized, studying phenomena such as attention concentration, head redundancy, and neuron specialization (Michel et al., 2019; Voita et al., 2019; Bau et al., 2017). Our framework unifies analysis and intervention within a single probability-ordered view. By explicitly selecting mass-covering state prefixes, it measures effective complexity and provides a controlled way to restrict computation, enabling systematic study of approximation trade-offs.