VEILMOE: HIERARCHICAL ROUTING ENVELOPES FOR PRIVACY-AWARE MIXTURE-OF-EXPERTS INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse Mixture-of-Experts (MoE) models activate only a few experts per token, but input-dependent routing exposes expert activity that can reveal information about the input. We present VeilMoE, which keeps the route used for computation inside a trusted controller while exposing a coarser execution structure to the accelerator. The controller maps the native Top-k route to a hierarchical envelope; experts in each touched group receive the same quantized capacity and run through transformed SwiGLU kernels. Under the stated execution assumptions, we prove exact request-local semantics in real arithmetic, characterize the exposed routing structure, and bound slot expansion. On OLMoE-1B-7B, $a = 16$ reduces routing-only Top-1 token reconstruction from 37.29% to 4.34% and Banking77 intent inference from 84.38% to 27.47%, with a utility change of -0.22 percentage points. A trainable attacker using raw rows and the envelope has low accuracy without target-epoch training examples, although it still uses 256 target-epoch requests for checkpoint selection; target-epoch training substantially increases accuracy. Full-decoder Qwen utility changes by -0.19 to -0.03 percentage points at the tested settings. With matched grouped-expert backends, OLMoE serving costs 7.10× Plain latency in prefill and 1.83× at 64-row cached decode. The privacy guarantees cover routing structure; transformed values and physical timing remain separate sources of leakage.

1 INTRODUCTION

Mixture-of-experts (MoE) models scale capacity efficiently by routing each token to only a small subset of Expert FFNs. Sparse conditional computation has evolved from sparsely gated MoE to large-scale GShard and Switch Transformer designs (Shazeer et al., 2017; Lepikhin et al., 2020; Fedus et al., 2022), and is now central to systems such as Mixtral, DeepSeekMoE, and OLMoE (Jiang et al., 2024; Dai et al., 2024; Muennighoff et al., 2025). Its efficiency, however, comes from *input-dependent routing*: different tokens activate different experts, and the resulting expert selections and loads can encode semantic information about the input. Recent attacks exploit expert activity, shared capacity, or routing sequences to infer token equality, semantics, and token content (Hayes et al., 2024; Yona et al., 2024; Ding et al., 2025; Nuriyev & Kulp, 2026; Lv et al., 2026). Thus, protecting tensor values alone does not address the routing structure exposed by sparse execution.

Existing private-MoE approaches make different trade-offs. Cryptographic systems such as CryptoMoE and SecMoE hide routing through secure dispatch, balanced workloads, or oblivious expert selection, but move sparse expert inference into HE/MPC or two-party protocols (Zhou et al., 2026; Shen et al., 2026). Lightweight representation transformations retain ordinary accelerator kernels, yet a fixed expert permutation or feature transform preserves request-specific routing correspondence (Lin et al., 2026). Evaluating every token on every expert hides the active set at a much higher cost: an M-expert, Top-k model schedules M/k times as many expert slots. Can we reveal less about MoE routing while retaining ordinary GPU kernels for the expert FFNs?

VeilMoE uses a public hierarchy to replace the fine-grained routing trace seen by the accelerator with group-level capacities. The trusted controller computes each request's original Top-k route, then reveals the touched groups and one quantized capacity per group. It pads every expert in a

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.

touched group to that capacity with cover rows. Expert-specific transforms let the padded batches run through ordinary SwiGLU kernels; the trusted return path removes covers, reverses the output transforms, and merges results with the original routing weights. Transformed tensor values and physical timing remain separate observation channels. In admitted, losslessly executed ticks, each real assignment reaches its selected expert exactly once.

We prove that the construction preserves request-local semantics and that the accelerator-visible routing structure is determined by the envelope. For Veil-O, the same characterization extends to logical metadata under an obliviousness assumption. We also bound the additional expert work as groups become coarser. On OLMoE, $a = 16$ reduces routing-only Top-1 token reconstruction from 37.29% to 4.34% and Banking77 intent inference from 84.38% to 27.47%; utility changes by -0.22 percentage points on 3,198 items. On Qwen, a routing-feature attacker observes layers 3, 7, 11, 15, 19, and 23: its concatenated-view accuracy changes from 69.87% to 16.75% at $a = 12$ and 3.12% at $a = 60$. Qwen utility and fidelity use the complete 24-layer decoder, with 97.07–97.35% next-token agreement. OLMoE value attacks recover substantially more signal when target-epoch training examples are available. The raw-row encoder also uses target-epoch validation at both training settings.

Our contributions are:

- We introduce hierarchical envelopes that expose touched expert groups and quantized capacities instead of fine-grained routes while retaining the reference expert computation.
- We prove request-local correctness, characterize the visible routing structure, and bound the padding cost as groups become coarser. Under its obliviousness contract, Veil-O extends the structural result to logical metadata.
- We measure token reconstruction, semantic and value attacks, full-decoder Qwen utility and fidelity, and serving cost, including the effect of target-epoch attack training.

2 PRELIMINARIES

Reference operator and request locality. A request q contains N_q hidden rows $X^{(q)} \in \mathbb{R}^{N_q \times d}$. With router $W_r \in \mathbb{R}^{d \times M}$ and M experts,

$$g_i^{(q)} = x_i^{(q)} W_r, \qquad S_i^{(q)} = \mathrm{TopK}(g_i^{(q)}, k), \tag{1}$$

$$E_e(x) = \big(\mathrm{SiLU}(xW_e^g) \odot xW_e^u\big) W_e^d, \tag{2}$$

$$y_i^{(q)} = \sum_{e \in J_i^{(q)}} \beta_{i,e}^{(q)} E_e(x_i^{(q)}), \qquad J_i^{(q)} = \{e : \beta_{i,e}^{(q)} \neq 0\} \subseteq S_i^{(q)}, \tag{3}$$

where $W_e^g, W_e^u \in \mathbb{R}^{d \times h}$, $W_e^d \in \mathbb{R}^{h \times d}$, and $\beta_{i,e}^{(q)}$ is the finalized routing coefficient. Ordinary no-drop Top-k has $J_i^{(q)} = S_i^{(q)}$ and $|J_i^{(q)}| = k$. With request-local public metadata μ_q, the reference routing rule is

$$(S^{(q)}, \beta^{(q)}, J^{(q)}) = \Delta_{\mathrm{loc}}(X^{(q)}, \mu_q), \tag{4}$$

so co-resident requests do not alter a request's finalized semantic assignments. Admitted execution is lossless: every finalized real assignment is evaluated exactly once by its selected expert.

Tick-level routing trace and observable views. At tick t, admitted requests are $\mathcal{Q}_t$, $X_t = \{X^{(q)}\}_{q \in \mathcal{Q}_t}$, and $N_t = \sum_q N_q$. Define

$$\tau_t^{\mathrm{sup}} = \{(q, i, e) : e \in J_i^{(q)}\}, \qquad \tau_t^{\mathrm{sem}} = \{(q, i, e, \beta_{i,e}^{(q)}) : (q, i, e) \in \tau_t^{\mathrm{sup}}\}, \tag{5}$$

$$n_{t,e} = \sum_{q,i} \mathbf{1}\{e \in J_i^{(q)}\}, \qquad r_t = \sum_e n_{t,e} = \sum_{q,i} |J_i^{(q)}| \leq N_t k. \tag{6}$$

U_t^- is all host-visible state fixed before current routing: public model/execution configuration, hierarchy/quantizer/level, row counts/chunk limits, transformed weights/epoch identifiers, and prior logical scheduler state. It excludes current private tensors, routes, loads, envelopes, transformed values, and measured hardware/time signals. The trusted controller holds routing, coefficients, cover

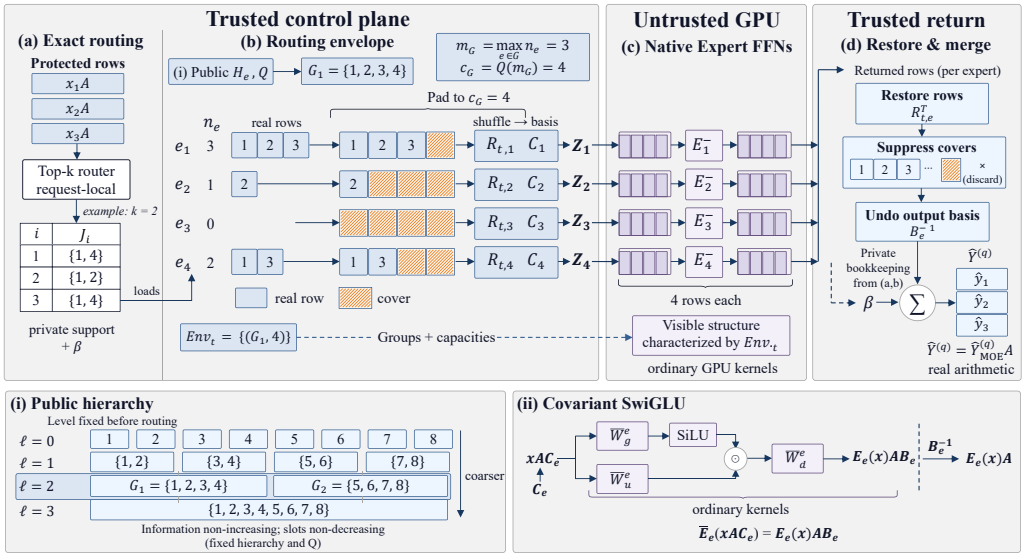

Figure 1: Overview of VeilMoE. The trusted control plane computes the exact request-local route and maps it to a coarser group-level routing envelope; equalized expert buckets are then executed with transformed native Expert-FFN kernels on the accelerator, before trusted restoration, cover suppression, and weighted merging recover the reference MoE computation.

bookkeeping, and transform secrets; the accelerator is honest-but-curious. We distinguish $\mathsf{View}_{\mathrm{str}}$ (expert IDs/order and shapes), $\mathsf{View}_{\mathrm{tee}}$ (externally visible trusted calls/addresses), $\mathsf{View}_{\mathrm{val}}$ (transformed tensors), and $\mathsf{View}_{\mathrm{time}}$ (physical timing/counters), with $\mathsf{View}_{\mathrm{meta}} = (\mathsf{View}_{\mathrm{str}}, \mathsf{View}_{\mathrm{tee}})$. Structural theorems use the finite-valued τ_t^{sup}; exactness uses τ_t^{sem}. Precise protocol-state and information-theoretic conditions are in Appendix A.8.

3 Related Work

Routing leakage and lightweight protection. Sparse MoE routing is input dependent, and expert specialization can make routing traces semantically informative (Yang et al., 2025). Recent attacks exploit shared capacity, accelerator activity, and routing-only metadata to infer token relations, semantics, or content (Hayes et al., 2024; Yona et al., 2024; Ding et al., 2025; Nuriyev & Kulp, 2026; Lv et al., 2026). An orthogonal line randomizes the gate itself through local-privacy constraints, trading exact deterministic routing for a privacy–accuracy trade-off (Akretche et al., 2024). Lower-cost defenses transform representations or model coordinates (Du et al., 2023; Zeng et al., 2025; Yue et al., 2021; Siyan et al., 2025; Lin et al., 2026; Roberts et al., 2025), but reusable bijections do not erase load multiplicities or repeated routing relations, and transformed values remain attackable through inversion or alignment (Dong et al., 2025a; Thomas et al., 2025; Lin et al., 2024; Kugler et al., 2021). Trusted-computing systems instead protect larger execution regions in hardware (Tramer & Boneh, 2018; Tan et al., 2025; Wang et al., 2025).

Cryptographic private inference. HE/MPC systems protect Transformer inference cryptographically (Hao et al., 2022; Li et al., 2022; Gupta et al., 2023; Luo et al., 2024; Dong et al., 2025b; Pang et al., 2024; Li et al., 2024; Lu et al., 2023; Zhang et al., 2025). CryptoMoE extends this line to sparse MoEs using balanced routing and secure dispatch/combine (Zhou et al., 2026); SecMoE uses Select-Then-Compute to conceal expert selection while retaining sparse encrypted execution (Shen et al., 2026). VeilMoE occupies a different point in this design space: it runs ordinary accelerator expert FFNs while reducing visible routing detail. Transformed values and timing remain separate observation channels.

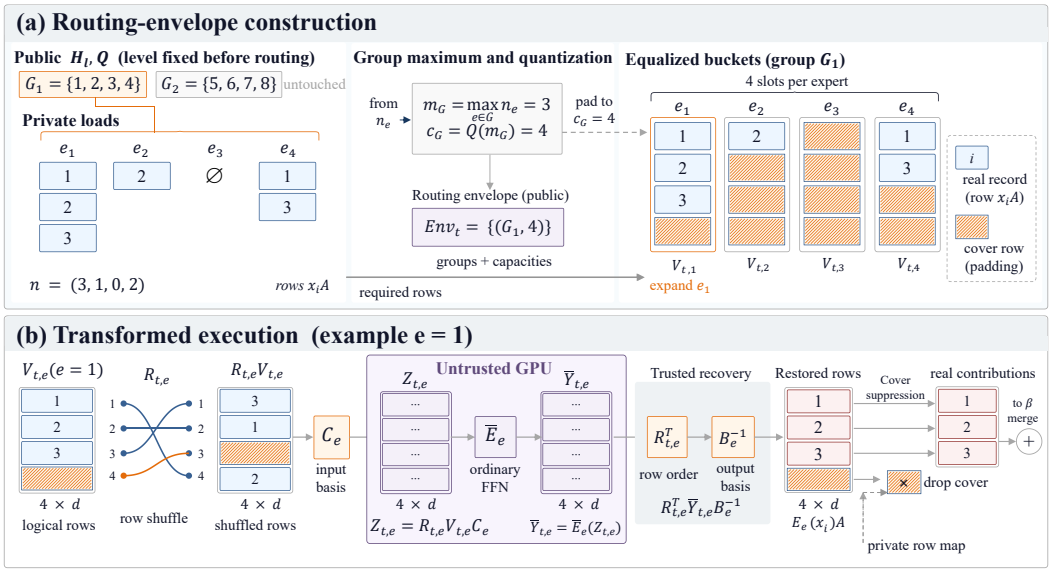

Figure 2: Routing-envelope construction and transformed Expert-FFN execution. The controller maps private expert loads to group-level quantized capacities and equalized buckets, then applies row permutation and input transforms before native GPU execution. Trusted recovery restores row order and output basis, suppresses cover rows, and merges the real contributions. The envelope characterizes accelerator-visible structure; value and timing views remain separate.

4 VeilMoE

4.1 Routing-envelope construction

VeilMoE keeps each request's exact request-local route inside the trusted controller and exposes only a group-level routing envelope to the accelerator. Figure 2 shows one tick; Algorithm 1 gives the semantic execution.

Public hierarchy and capacity rule. For $\ell = 0, \ldots, L$, let H_ℓ be nested partitions of $[M]$,

$$H_0 \preceq H_1 \preceq \cdots \preceq H_L, \quad H_0 = \{\{1\}, \ldots, \{M\}\}, \quad a_\ell = \max_{G \in H_\ell} |G|, \tag{7}$$

where $H_{\ell+1}$ coarsens H_ℓ. The public nondecreasing quantizer $Q : \mathbb{N}_0 \to \mathbb{N}_0$ satisfies $Q(0) = 0$ and

$$m \le Q(m) \le \lambda m \qquad (m \ge 1) \tag{8}$$

for public $\lambda \ge 1$. The hierarchy is calibrated once on public routing samples s. For a group G, define its padded cost and merge penalty

$$C_s(G) = |G| Q\left(\max_{e \in G} n_e^{(s)}\right), \tag{9}$$

$$w(A, B) = \mathbb{E}_s[C_s(A \cup B) - C_s(A) - C_s(B)]. \tag{10}$$

Each level is formed by deterministic low-w merging of disjoint groups. Monotonicity of Q gives $w(A, B) \ge 0$; the matching procedure and proof are in Appendix A.8. The parameter a_ℓ is a maximum group size, not an anonymity guarantee.

The merge cost depends on how groups are loaded together. If both groups require the same quantized capacity on a calibration sample, their merger adds no slots. If one requires a larger capacity, every expert in the other group must be padded to that larger value. The penalty therefore reflects both the capacity difference and the number of experts receiving additional cover rows. Averaging this quantity over calibration samples favors groups with compatible load patterns. Once the hierarchy is fixed, constructing an envelope at a selected level requires counting expert assignments, taking group maxima, and applying the public capacity quantizer.

Online envelope. At level ℓ, for each $G \in H_\ell$ define

$$m_{t,G} = \max_{e \in G} n_{t,e}, \qquad c_{t,G} = Q(m_{t,G}), \qquad \mathsf{Env}_t^{(\ell)} = \{(G, c_{t,G}) : \textstyle\sum_{e \in G} n_{t,e} > 0\}. \tag{11}$$

We write Env_t at the selected level. It discloses touched groups and quantized group capacities, but not within-group loads or token–expert correspondence.

Before quantization, the maximum expert load is the smallest common capacity that can accommodate all real assignments in a group. An average could underallocate the busiest expert, whereas the sum would reserve space for every group assignment at every expert. Equal capacities extend to group members with no real rows, removing size differences that would otherwise expose individual expert loads within a touched group. Capacity rounding can map different maxima to one disclosed value. This construction leaves the controller with the individual loads and row assignments, while the accelerator receives a shared upper bound for each group it must execute.

4.2 Exact transformed execution

Router and equalized buckets. Let $A \in \mathrm{GL}(d)$ be the common protected basis. For $\widetilde{x} = xA$, the trusted router stores $\widetilde{W}_r = A^{-1}W_r$, hence

$$\widetilde{x}\widetilde{W}_r = xW_r, \tag{12}$$

so the protected-basis Top-k route equals the reference route in real arithmetic. For expert e in touched group G, the controller forms $V_{t,e} \in \mathbb{R}^{c_{t,G} \times d}$ from its $n_{t,e}$ real rows $x_i^{(q)}A$ and $c_{t,G} - n_{t,e}$ covers, which carry zero semantic weight and are removed before merge. With fresh row permutation $R_{t,e}$ and expert-specific input map $C_e \in \mathrm{GL}(d)$,

$$Z_{t,e} = R_{t,e}V_{t,e}C_e. \tag{13}$$

Covariant SwiGLU. For $C_e, B_e \in \mathrm{GL}(d)$, intermediate permutation Π_e, and invertible diagonal D_e, store

$$\bar{W}_e^g = C_e^{-1}A^{-1}W_e^g\Pi_e, \qquad \bar{W}_e^u = C_e^{-1}A^{-1}W_e^u D_e\Pi_e, \tag{14}$$

$$\bar{W}_e^d = \Pi_e^\top D_e^{-1}W_e^d AB_e. \tag{15}$$

The transformed expert obeys

$$\bar{E}_e(xAC_e) = E_e(x)AB_e. \tag{16}$$

The placement of the intermediate transforms follows from the nonlinear structure of SwiGLU. A channel permutation commutes with SiLU because the activation acts independently on each coordinate. General linear mixing does not have this property. Diagonal scaling is applied to the linear up branch, where it passes through the elementwise product and cancels against the inverse scaling in the down projection. Both branches use the same permutation, preserving the original pairing of coordinates in the product. The remaining expert output map is removed before contributions are combined in the common protected basis, yielding the original weighted sum in that basis.

Consequently, for every real assignment (q, i, e),

$$\bar{E}_e(x_i^{(q)}AC_e)B_e^{-1} = E_e(x_i^{(q)})A, \qquad \widehat{y}_i^{(q)} = \sum_{e \in J_i^{(q)}} \beta_{i,e}^{(q)} E_e(x_i^{(q)})A = y_i^{(q)}A. \tag{17}$$

Row permutation only changes physical position; $R_{t,e}^\top$ restores logical order before cover suppression and weighted merge. The SiLU/Hadamard cancellation, row-wise equivariance, and chunk-reassembly proof are in Appendix A.8. In our implementation C_e, B_e are fast scaled-orthogonal maps with $\kappa_2 = 1$ and $O(d \log d)$ application cost; they are reusable linear transforms.

Trusted-control profiles. Veil-S may gather/scatter only real records inside the trusted component. Veil-O additionally requires the complete post-envelope packing/read/select/scatter trace to depend only on $(\mathsf{Env}_t, U_t^-, \Omega_t)$, while pre-envelope routing follows a public fixed-iteration schedule and value-dependent aborts are excluded. Appendix A.8 states the whole-tick contract; physical timing remains in $\mathsf{View}_{\mathrm{time}}$.

Algorithm 1 VeilMoE execution for one MoE layer at tick t

1: **for all** $q \in \mathcal{Q}_t$ independently **do**
2: finalize $J_i^{(q)}$ and $\beta_{i,e}^{(q)}$ with the reference router
3: **end for**
4: Compute $n_{t,e}$ and disclose Env_t by eq. (11).
5: **for all** $(G, c) \in \mathsf{Env}_t$ and $e \in G$ **do**
6: form $V_{t,e}$; sample $R_{t,e}$; evaluate $\bar{Y}_{t,e} = \bar{E}_e(R_{t,e} V_{t,e} C_e)$
7: restore $R_{t,e}^\top \bar{Y}_{t,e} B_e^{-1}$ and discard cover records
8: **end for**
9: merge real contributions with $\beta_{i,e}^{(q)}$ and return $\{\widehat{Y}^{(q)}\}_{q \in \mathcal{Q}_t}$.

4.3 Guarantees and information–cost tradeoff

The following four results are the main formal chain; Appendix A.8 supplies complete proofs.

Theorem 1 (Exact semantics and cross-request non-interference)**.** *Under exact real arithmetic and admitted-tick lossless execution, every admitted request q satisfies*

$$\widehat{Y}^{(q)} = Y_{\mathrm{MoE}}^{(q)} A. \tag{18}$$

Fixing $X^{(q)}$, μ_q, the reference model, and A therefore fixes request q's semantic output independently of co-resident requests.

Equation 17 is the key derivation: expert covariance removes C_e/B_e, $R_{t,e}^\top$ restores row correspondence, covers contribute zero, and the original $\beta_{i,e}^{(q)}$ reconstruct the reference MoE sum.

When another request joins a tick, the controller may allocate larger buckets or invoke additional groups, while each request keeps its finalized expert assignments and mixing coefficients. The private row map associates every returned real record with its request, token, and expert before accumulation. Cover records are discarded through this map even when their numerical outputs are nonzero. Their exclusion is an explicit operation on trusted bookkeeping, so cover generation does not require an input that makes the expert return zero. Chunking preserves this argument when reassembly returns every real record exactly once to the controller for its weighted merge.

Theorem 2 (Exact envelope characterization of accelerator structure)**.** *For every admitted protocol-complete tick,*

$$\mathsf{View}_{\mathrm{str}\,t} = F_{\mathrm{str}}(\mathsf{Env}_t, U_t^-), \qquad \mathsf{Env}_t = G_{\mathrm{str}}(\mathsf{View}_{\mathrm{str}\,t}, U_t^-), \tag{19}$$

and hence

$$I(X_t; \mathsf{View}_{\mathrm{str}\,t} \mid U_t^-) = I(X_t; \mathsf{Env}_t \mid U_t^-) \le I(X_t; \tau_t^{\mathrm{sup}} \mid U_t^-). \tag{20}$$

Given U_t^-, the envelope fixes touched groups, expert IDs and order, and bucket and chunk shapes. Those visible IDs and shapes also recover the envelope. The structural view therefore contains the envelope's routing information but no finer support-trace detail; transformed values and timing are separate observations.

Theorem 3 (Oblivious metadata factorization for Veil-O)**.** *Under the Veil-O whole-tick contract, let Ω_t denote permitted private protocol randomness/state, independent of $(X_t, \tau_t^{\mathrm{sup}}, \tau_t^{\mathrm{sem}})$ conditional on U_t^-. Then*

$$\mathsf{View}_{\mathrm{meta}\,t} = F_{\mathrm{meta}}(\mathsf{Env}_t, U_t^-, \Omega_t), \qquad (X_t, \tau_t^{\mathrm{sem}}) \to \mathsf{Env}_t \to \mathsf{View}_{\mathrm{meta}\,t} \mid U_t^-. \tag{21}$$

Since $\mathsf{View}_{\mathrm{meta}}$ *contains* $\mathsf{View}_{\mathrm{str}}$,

$$I(X_t; \mathsf{View}_{\mathrm{meta}\,t} \mid U_t^-) = I(X_t; \mathsf{Env}_t \mid U_t^-) \le I(X_t; \tau_t^{\mathrm{sup}} \mid U_t^-). \tag{22}$$

The complete conditional-law argument and the stateful-oblivious-memory conditions are deferred to Appendix A.8.

For each hierarchy level define scheduled slots and slot expansion

$$S_t^{(\ell)} = \sum_{(G,c) \in \mathsf{Env}_t^{(\ell)}} |G| c, \qquad \rho_t^{(\ell)} = S_t^{(\ell)} / r_t \quad (r_t > 0). \tag{23}$$

If a coarse group $G' = \biguplus_j G_j$ merges children from level ℓ, assign capacity 0 to an absent (untouched) child. Monotonicity of Q gives

$$c_{t,G'} = Q\left(\max_j m_{t,G_j}\right) = \max_j c_{t,G_j}. \tag{24}$$

The parent is touched iff at least one child is touched, so the coarse envelope is computable from the fine envelope by public group merging and a max over child capacities.

Theorem 4 (Monotone envelope coarsening with bounded cost)**.** *For one fixed nested hierarchy and quantizer Q satisfying eq.* (8)*, every $\ell < L$ obeys*

$$\mathsf{Env}_t^{(\ell+1)} = \Phi_\ell(\mathsf{Env}_t^{(\ell)}), \qquad I(X_t; \mathsf{Env}_t^{(\ell+1)} \mid Z) \le I(X_t; \mathsf{Env}_t^{(\ell)} \mid Z), \tag{25}$$

$$S_t^{(\ell+1)} \ge S_t^{(\ell)}, \qquad 1 \le \rho_t^{(\ell)} \le \lambda a_\ell, \quad \rho_t^{(\ell+1)} \ge \rho_t^{(\ell)}. \tag{26}$$

Equation 24 yields the deterministic post-processing; data processing gives the information ordering. The slot inequalities follow pointwise from $m_{t,G} \le Q(m_{t,G}) \le \lambda m_{t,G}$ and $|G| \le a_\ell$; the complete derivation is in Appendix A.8.

Perfect active-set hiding lower bound. Under assignment-preserving plaintext execution, let $M_{\text{feas}}(u)$ be the number of experts that can be active for some private input under fixed public state $U_t^- = u$. Any exact mechanism whose invoked-expert distribution is independent of the real active set must satisfy

$$\rho_t^{\text{mech}} \ge \max\{1, M_{\text{feas}}(u)/r_t\}. \tag{27}$$

For ordinary no-drop Top-k decode with full expert feasibility, $r_t = N_t k$ and $M_{\text{feas}}(u) = M$, so $\rho_t^{\text{mech}} \ge \max 1, M/(N_t k)$. The proof formalizes why exact active-set hiding under expert-preserving plaintext execution must schedule every expert that is feasible under the same public state for every such admissible decode tick.

The hierarchy makes the trade-off explicit: coarser levels reveal no more routing structure but schedule at least as many expert slots. At low decode concurrency, the lower bound shows why perfect active-set hiding requires substantial extra work. A serving system can choose a hierarchy level from public pre-routing state without changing each request's MoE computation. Section 5.5 measures the resulting runtime at the tested operating point.

5 Experiments

We evaluate attacks on routing structure and transformed values, model utility and fidelity, and serving cost on matched backends.

5.1 Experimental Setup

Setup. We use OLMoE-1B-7B (Muennighoff et al., 2025) (64 routed experts, Top-8, 16 MoE layers) and Qwen1.5-MoE-A2.7B (60 routed experts, Top-4, 24 MoE layers plus a shared expert). Veil uses the CostAware hierarchy and next-power-of-two capacity rounding. Router, transformation, and merge arithmetic use FP32; expert FFNs use BF16. OpenWebText token-reconstruction splits are request-disjoint. The primary semantic attacks use a standardized-logistic full-envelope classifier with three seeds and train/test splits of 20,000/7,600 for AG News and 9,996/3,080 for Banking77. We test Layer-8 raw-row and all-16-layer sketch value attacks separately. Utility uses 3,198 ARC-Challenge, HellaSwag, and MMLU items (Clark et al., 2018; Zellers et al., 2019; Hendrycks et al., 2020). Runtime uses one 48G RTX 4090 and matched grouped-expert backends at $a = 16$, with prefill $B = 4, L = 512$ and cached decode at $H = 128$, $N \in \{1, 64\}$.

5.2 Main Results

As shown in Table 1, StaticPerm closely tracks Plain, whereas Veil substantially suppresses fine-grained routing attacks. At $a = 16$, Top-$1/5/10$ token reconstruction falls from 37.29/51.84/56.82% to 4.34/14.29/21.64%; Top-1 is close to the 3.28% prior baseline. Banking77 inference decreases

Table 1: Structural attack–cost tradeoff on OLMoE. OWT reports Top-k token reconstruction, while AG News and Banking77 report semantic attack accuracy and AUROC. Lower attack scores indicate stronger protection; ρ denotes the mean expert-slot expansion relative to unpadded execution.

	OpenWebText token recovery ↓			AG News			Banking77		
Method	Top-1	Top-5	Top-10	Acc. ↓	AUROC ↓	ρ	Acc. ↓	AUROC ↓	ρ
Prior / chance	3.28	13.60	20.61	25.00	$0.5000^{\dagger}$	—	1.30	$0.5000^{\dagger}$	—
Plain	37.29	51.84	56.82	88.42	0.9767	1.00	84.38	0.9948	1.00
StaticPerm	37.34	51.83	56.83	88.38	0.9765	1.00	84.32	0.9948	1.00
Veil ($a = 2$)	6.38	16.95	23.36	87.26	0.9721	1.965	78.57	0.9916	1.877
Veil ($a = 4$)	5.37	15.67	22.50	87.99	0.9732	2.748	60.23	0.9666	2.742
Veil ($a = 8$)	4.82	14.87	21.89	86.49	0.9670	3.637	42.24	0.9277	3.772
Veil ($a = 16$)	**4.34**	**14.29**	**21.64**	**84.22**	**0.9579**	**4.644**	**27.47**	**0.8796**	**4.992**
Veil ($a = 32$)	3.82	14.30	21.59	78.96	0.9325	5.766	14.48	0.8003	6.211
Veil ($a = 64$)	3.42	13.86	20.90	65.17	0.8584	7.215	7.40	0.7130	7.477

Table 2: OLMoE value attacks under target-epoch training. Semantic attack accuracy (%) is reported for a Layer-8 trainable raw-row-plus-envelope attacker and an all-16-layer value-sketch attacker. Results compare no target-epoch training examples ($K = 0$) with substantial same-epoch training exposure ($K = 1024$).

Backbone	Attacker view	Dataset	$K = 0$	$K = 1024$
OLMoE-1B-7B	Trainable raw rows + envelope (Layer 8)	Banking77	1.40	54.74
	Trainable raw rows + envelope (Layer 8)	AG News	24.97	86.13
	All-16-layer value sketch	Banking77	1.85	52.00
	All-16-layer value sketch	AG News	27.18	87.90

from 84.38% to 27.47% at $4.992\times$ mean slot expansion. AG News is more resistant to group coarsening, decreasing only from 88.42% to 84.22% at $4.644\times$, indicating substantial task-dependent residual envelope leakage.

5.3 Joint-View Adversarial Evaluation

To examine transformed values alongside routing structure, we evaluate two OLMoE attackers: a trainable encoder using Layer-8 raw rows and the envelope, and an all-16-layer value sketch. Because both the representation and layer set differ, we vary the target-epoch training cap K within each configuration.

With no target-epoch training examples, the Layer-8 raw-row-plus-envelope attacker scores 1.40% on Banking77 and 24.97% on AG News. At nominal $K = 1024$, these scores rise to 54.74% and 86.13%. At both settings, checkpoint selection uses 256 held-out target-epoch validation requests; $K = 0$ excludes target-epoch examples from training, not validation. The all-16-layer sketch gives 1.85/27.18% at $K = 0$ and 52.00/87.90% at $K = 1024$ on Banking77/AG News. Appendix A.2 gives the matched frozen-encoder comparison.

The gain from target-epoch training shows that reusable value maps can leak input information. To quantify the deployment cost of rekeying, we use measured one-layer medians of 61.8 ms for transform generation and 488.8 ms for expert reparameterization. Extrapolating over all 16 layers gives an estimated whole-model rekey cost of 8.81 s, corresponding to 68.8, 17.2, and 4.3 ms per request when amortized over 128, 512, and 2,048 served requests, respectively.

5.4 Utility and Cross-Model Evidence

Table 3 summarizes structural attack diagnostics, utility, and finite-precision fidelity. Qwen routing-feature attacks concatenate six observed layers; its transformed-execution utility and fidelity use the full 24-layer decoder. OLMoE utility changes by only -0.22 pp at $a = 16$, with 95.90% next-token agreement; the paired 95% interval $[-0.66, +0.22]$ includes zero. A 360-case request-isolation test also shows no detectable excess of Veil-only changes ($p = 0.4421$). On Qwen, utility changes by only -0.06, -0.03, and -0.19 pp at $a = 2, 12, 60$, while next-token agreement remains 97.07–97.35%. In contrast, the six-layer routing-feature attack falls from 59.22% at $a = 2$ to 16.75% at

$a = 12$ and 3.12% at $a = 60$. Thus, across both backbones, coarser routing substantially reduces the tested structural leakage without a comparable loss in downstream utility. Appendix A.3 provides the full utility, fidelity, and request-isolation breakdowns.

Table 3: Structural attacks, utility, and finite-precision fidelity. Banking77 uses the primary OLMoE structural view and the Qwen routing-feature view over layers 3, 7, 11, 15, 19, and 23. Qwen utility and fidelity execute all 24 routed layers at each reported a. The Δ pp column uses unrounded accuracies.

		Banking77	Utility ↑					Fidelity ↑ / ↓	
Backbone	Method	Attack acc. ↓	ARC-C	Hella	MMLU	Overall	Δ pp	Agree	Rel.L_2
OLMoE-1B-7B	Plain	84.38	46.50	43.70	38.01	42.90	0.00	100.00	0.00000
	Veil $a = 16$	27.47	46.50	43.20	37.82	42.68	-0.22	95.90	0.03395
Qwen1.5-MoE-A2.7B	Plain	69.87	43.69	56.90	40.74	46.87	0.00	100.00	0.00000
	Veil $a = 2$	59.22	43.69	56.70	40.74	46.81	-0.06	97.35	0.01923
	Veil $a = 12$	16.75	43.69	56.80	40.74	46.84	-0.03	97.23	0.01921
	Veil $a = 60$	3.12	43.43	56.60	40.74	46.69	-0.19	97.07	0.01919

5.5 Optimized System Efficiency

Table 4 reports end-to-end runtime at the representative $a = 16$ operating point using the same optimized grouped-expert backend for Plain and VeilMoE. During prefill, VeilMoE incurs a 7.10× latency overhead with a 6.63× slot expansion. Their close ratio indicates that the dominant cost is the additional padded expert work required by the envelope, rather than a change in the underlying Expert-FFN backend.

The decode regime is more favorable. At $N = 1$, VeilMoE costs 2.36× Plain, whereas at $N = 64$ the overhead falls to 1.83× as more real assignments share the padded execution schedule. The corresponding slot expansion also drops from 7.25× to 2.78×. Thus, VeilMoE's serving cost is strongly workload dependent under the tested workloads and concurrency levels: low-concurrency execution remains expensive, while concurrent decode amortizes much of the structural-padding cost. These measurements use constant zero covers and therefore characterize the canonical serving path rather than the separate value-leakage configurations. Appendix A.4 provides p95 latency, kernel counts, memory usage, and component-level attribution.

Table 4: OLMoE runtime at $a = 16$ on one RTX 4090. Plain and VeilMoE use the same grouped-expert backend; ρ is mean slot expansion and latency overhead is the VeilMoE-to-Plain mean-latency ratio. VeilMoE uses constant zero covers.

Workload	Plain mean	Plain p95	VeilMoE mean	VeilMoE p95	ρ	Latency overhead
Prefill $B = 4, L = 512$	150.26	156.34	1066.92	1084.55	6.63	7.10×
Decode $N = 1$	74.05	75.51	174.49	193.55	7.25	2.36×
Decode $N = 64$	100.03	100.66	182.83	195.71	2.78	1.83×

6 Conclusion

VeilMoE keeps request-local routing in a trusted controller and exposes a coarser group-level envelope to accelerator expert FFNs. Under the stated assumptions, it preserves real-arithmetic semantics and bounds both structural disclosure and extra expert work; the stronger Veil-O metadata result requires oblivious trusted control. On OLMoE, coarser groups reduce structural attack success by different amounts across tasks. Value attacks, however, recover signal from reusable transforms when target-epoch training examples are available. Utility changes little at the tested OLMoE and Qwen settings. On a matched grouped-expert backend, OLMoE costs 1.83× Plain at 64-row decode and 7.10× in prefill. Transformed values and physical timing remain outside the routing-structure guarantee.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish writing and to assist with deriving, checking, and presenting mathematical formulas and proof arguments. We did not use generative AI tools to generate experimental results, and other disclosure categories are not applicable to this work. We reviewed all AI-assisted work. In particular, the authors independently checked AI-assisted mathematical derivations and proofs for correctness and consistency with the stated assumptions, and reviewed and revised all AI-assisted text for technical accuracy, clarity, and consistency with the cited literature and reported experiments. We take responsibility for the final content of this work, including text, claims, proofs, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Section 4 specifies the construction, transformed Expert-FFN equations, and execution pseudocode; Appendix A.8 gives the assumptions and derivations. Section 5 states the model backbones, datasets, precision, attacks, and canonical runtime settings. The appendix expands these experiments with exposure counts, the matched encoder comparison, utility and fidelity breakdowns, and runtime diagnostics; repeated main-table measurements use the same values and displayed precision. Appendices A.5–A.7 describe separate diagnostic studies and their own comparison protocols.

REFERENCES

Wissam Akretche, Frédéric LeBlanc, and Mario Marchand. Tighter risk bounds for mixtures of experts. *arXiv preprint arXiv:2410.10397*, 2024.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Yu Wu, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 1280–1297, 2024.

Ruyi Ding, Tianhong Xu, Xinyi Shen, Aidong Adam Ding, and Yunsi Fei. Moecho: Exploiting side-channel attacks to compromise user privacy in mixture-of-experts llms. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2159–2173, 2025.

Tian Dong, Yan Meng, Shaofeng Li, Guoxing Chen, Zhen Liu, and Haojin Zhu. Depth gives a false sense of privacy:{LLM} internal states inversion. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 1629–1648, 2025a.

Ye Dong, Wen-jie Lu, Yancheng Zheng, Haoqi Wu, Derun Zhao, Jin Tan, Zhicong Huang, Cheng Hong, Tao Wei, Wen-Guang Chen, et al. Puma: Secure inference of llama-7b in five minutes. *Security and Safety*, 4:2025014, 2025b.

Minxin Du, Xiang Yue, Sherman SM Chow, Tianhao Wang, Chenyu Huang, and Huan Sun. Dp-forward: Fine-tuning and inference on language models with differential privacy in forward pass. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2665–2679, 2023.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Kanav Gupta, Neha Jawalkar, Ananta Mukherjee, Nishanth Chandran, Divya Gupta, Ashish Panwar, and Rahul Sharma. Sigma: Secure gpt inference with function secret sharing. *Cryptology ePrint Archive*, 2023.

Meng Hao, Hongwei Li, Hanxiao Chen, Pengzhi Xing, Guowen Xu, and Tianwei Zhang. Iron: Private inference on transformers. *Advances in neural information processing systems*, 35:15718–15731, 2022.

Jamie Hayes, Ilia Shumailov, and Itay Yona. Buffer overflow in mixture of experts. *arXiv preprint arXiv:2402.05526*, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Kai Kugler, Simon Münker, Johannes Höhmann, and Achim Rettinger. Invbert: Reconstructing text from contextualized word embeddings by inverting the bert pipeline. *arXiv preprint arXiv:2109.10104*, 2021.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020.

Dacheng Li, Rulin Shao, Hongyi Wang, Han Guo, Eric P Xing, and Hao Zhang. Mpcformer: fast, performant and private transformer inference with mpc. *arXiv preprint arXiv:2211.01452*, 2022.

Zhengyi Li, Kang Yang, Jin Tan, Wen-jie Lu, Haoqi Wu, Xiao Wang, Yu Yu, Derun Zhao, Yancheng Zheng, Minyi Guo, et al. Nimbus: Secure and efficient two-party inference for transformers. *Advances in Neural Information Processing Systems*, 37:21572–21600, 2024.

Yu Lin, Qizhi Zhang, Quanwei Cai, Jue Hong, Wu Ye, Huiqi Liu, and Bing Duan. An inversion attack against obfuscated embedding matrix in language model inference. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 2100–2104, 2024.

Yu Lin, Qizhi Zhang, Wenqiang Ruan, Daode Zhang, Jue Hong, Ye Wu, Hanning Xia, Yunlong Mao, and Sheng Zhong. Towards privacy-preserving llm inference via covariant obfuscation (technical report). *arXiv preprint arXiv:2603.01499*, 2026.

Wen-jie Lu, Zhicong Huang, Zhen Gu, Jingyu Li, Jian Liu, Cheng Hong, Kui Ren, Tao Wei, and WenGuang Chen. Bumblebee: Secure two-party inference framework for large transformers. *Cryptology ePrint Archive*, 2023.

Jinglong Luo, Yehong Zhang, Zhuo Zhang, Jiaqi Zhang, Xin Mu, Hui Wang, Yue Yu, and Zenglin Xu. Secformer: Fast and accurate privacy-preserving inference for transformer models via smpc. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13333–13348, 2024.

Bo Lv, Zhiheng Xu, KeDong Xiu, Ruyi Ding, Tianhang Zheng, Zhibo Wang, and Kui Ren. Routescan: A non-intrusive approach to auditing moe llms safety via expert routing telemetry. *arXiv preprint arXiv:2605.24817*, 2026.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, et al. Olmoe: Open mixture-of-experts language models. In *International Conference on Learning Representations*, volume 2025, pp. 62061–62121, 2025.

Amir Nuriyev and Gabriel Kulp. Expert selections in moe models reveal (almost) as much as text. *arXiv preprint arXiv:2602.04105*, 2026.

Qi Pang, Jinhao Zhu, Helen Möllering, Wenting Zheng, and Thomas Schneider. Bolt: Privacy-preserving, accurate and efficient inference for transformers. In *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 4753–4771. IEEE, 2024.

Jay Roberts, Kyle Mylonakis, Sidhartha Roy, and Kaan Kale. Learning obfuscations of llm embedding sequences: Stained glass transform. *arXiv preprint arXiv:2506.09452*, 2025.

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. *arXiv preprint arXiv:1701.06538*, 2017.

Bowen Shen, Yuyue Chen, Peng Yang, Bin Zhang, Xi Zhang, and Zoe L Jiang. Secmoe: Communication-efficient secure moe inference via select-then-compute. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 25286–25294, 2026.

Li Siyan, Vethavikashini Chithrra Raghuram, Omar Khattab, Julia Hirschberg, and Zhou Yu. Papillon: Privacy preservation from internet-based and local language model ensembles. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3371–3390, 2025.

Yifan Tan, Cheng Tan, Zeyu Mi, and Haibo Chen. Pipellm: Fast and confidential large language model services with speculative pipelined encryption. In *Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1*, pp. 843–857, 2025.

Rahul Krishna Thomas, Louai Zahran, Erica Choi, Akilesh Potti, Micah Goldblum, and Arka Pal. Hidden no more: Attacking and defending private third-party llm inference. In *ICML*, volume 267, pp. 59434–59469, 2025.

Florian Tramer and Dan Boneh. Slalom: Fast, verifiable and private execution of neural networks in trusted hardware. *arXiv preprint arXiv:1806.03287*, 2018.

Chenxu Wang, Danqing Tang, Changxu Ci, Junjie Huang, Yankai Xu, Fengwei Zhang, Jiannong Cao, Jie Song, Shoumeng Yan, Tao Wei, et al. ccai: a compatible and confidential system for ai computing. In *Proceedings of the 58th IEEE/ACM International Symposium on Microarchitecture*, pp. 340–353, 2025.

Xingyi Yang, Constantin Venhoff, Ashkan Khakzar, Christian Schroeder De Witt, Puneet K Dokania, Adel Bibi, and Philip Torr. Mixture of experts made intrinsically interpretable. *arXiv preprint arXiv:2503.07639*, 2025.

Itay Yona, Ilia Shumailov, Jamie Hayes, and Nicholas Carlini. Stealing user prompts from mixture of experts. *arXiv preprint arXiv:2410.22884*, 2024.

Xiang Yue, Minxin Du, Tianhao Wang, Yaliang Li, Huan Sun, and Sherman SM Chow. Differential privacy for text analytics via natural text sanitization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3853–3866, 2021.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 4791–4800, 2019.

Ziqian Zeng, Jianwei Wang, Junyao Yang, Zhengdong Lu, Haoran Li, Huiping Zhuang, and Cen Chen. Privacyrestore: Privacy-preserving inference in large language models via privacy removal and restoration. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10821–10855, 2025.

Yancheng Zhang, Jiaqi Xue, Mengxin Zheng, Mimi Xie, Mingzhe Zhang, Lei Jiang, and Qian Lou. Cipherprune: Efficient and scalable private transformer inference. In *International Conference on Learning Representations*, volume 2025, pp. 52967–52985, 2025.

Yifan Zhou, Tianshi Xu, Jue Hong, Ye Wu, and Meng Li. Cryptomoe: Privacy-preserving and scalable mixture of experts inference via balanced expert routing. *Advances in Neural Information Processing Systems*, 38:57480–57503, 2026.

A SUPPLEMENTARY MATERIAL

We first give the protocols and additional measurements behind the main tables, then present separate studies of structural views, cover policies, sequence attacks, and systems behavior.

A.1 STRUCTURAL EVALUATION PROTOCOL

The OLMoE structural results use the protocol of Table 1: the CostAware hierarchy, next-power-of-two capacity rounding, and a standardized-logistic attacker on the full envelope. Semantic accuracies are averaged over three seeds. The attack train/test splits contain 20,000/7,600 AG News requests and 9,996/3,080 Banking77 requests. The same splits, hierarchy, quantizer, and attacker are used throughout the reported a sweep.

At $a = 16$, Banking77 accuracy is 27.47% with mean slot expansion 4.992, and AG News accuracy is 84.22% with mean slot expansion 4.644. OpenWebText Top-$1/5/10$ token recovery is 4.34/14.29/21.64%. Increasing a from 2 to 64 changes Banking77 accuracy from 78.57% to 7.40%, whereas AG News changes from 87.26% to 65.17%. The contrast motivates the separate diagnostics below, which examine group identity, capacity, and public request length.

Cost denominator. For a nonempty tick, slot expansion is $\rho_t = S_t/r_t$, where r_t counts real expert assignments; mean slot expansion averages the evaluated observations. The prior/chance row is an attack reference and has no execution cost. Padding a request to a fixed public length defines a different schedule: for L_{pub} scheduled token rows and N useful token rows, an all-expert endpoint has scheduled ratio M/k but useful-row ratio $ML_{\mathrm{pub}}/(kN)$. These ratios coincide only when $L_{\mathrm{pub}} = N$.

A.2 VALUE-ATTACK CONFIGURATIONS AND ENCODER COMPARISON

Table 2 reports two OLMoE attack configurations: Layer-8 raw rows plus the envelope, and an all-16-layer value sketch. We expand both using the same settings as the main table. Since the representation and layer set both change, comparisons across these configurations do not isolate a layer-count effect.

Target-epoch exposure. For the OLMoE raw/frozen configuration, K is the requested target-epoch training cap. The target-epoch pools contain 1,024 Banking77 and 2,048 AG News requests. The last 256 requests are reserved for validation before training examples are selected. Table 5 gives the realized counts. The frozen and trainable encoders use the same validation slice, including at $K = 0$; thus $K = 0$ means no target-epoch training examples.

Table 5: Realized target-epoch exposure for the OLMoE Layer-8 raw/frozen experiment. Training and validation are disjoint.

Dataset	Nominal K	Training	Validation
Banking77	0	0	256
Banking77	1024	768	256
AG News	0	0	256
AG News	1024	1024	256

Frozen versus trainable encoder. Both encoders pool expert-row features and concatenate the 9-d public envelope. Table 6 repeats the trainable results from Table 2 alongside the frozen comparison. Without target-epoch training examples, both score near the class priors. With such examples, the trainable encoder recovers more signal than the frozen one. The Banking77 value-ablation score is 1.14% at nominal $K = 1024$.

Dataset (prior)	Attacker	$K = 0$	$K = 1024$
Banking77 (1.30)	Frozen ϕ	1.25	1.40
	Trainable raw	**1.40**	54.74
AG News (25.00)	Frozen ϕ	26.04	39.17
	Trainable raw	**24.97**	86.13

Table 6: Semantic accuracy (%) for frozen versus trainable raw-row attackers. Both arms are raw-tensor-plus-envelope classifiers (9-d public envelope concatenated with pooled expert-row features), not value-only; K is the requested target-epoch training cap.

All-layer views. The all-16-layer OLMoE configuration uses a 128-dimensional per-layer JL sketch. Its Banking77 accuracies are 1.85% and 52.00%, and its AG News accuracies are 27.18% and 87.90%, at the two training-cap settings in Table 2. Unlike the Layer-8 encoder, the sketch does not use the target-epoch validation slice.

Rekey-cost estimate. The main-text estimate uses measured one-layer component medians on shape-matched initialized OLMoE experts: 61.8 ms for transform generation and 488.8 ms for expert reparameterization. Multiplying their sum by 16 gives the 8.81 s full-model estimate. Spreading it over 128, 512, or 2,048 served requests gives 68.8, 17.2, or 4.3 ms per request. This is a cost calculation based on component timings; the served-request interval is separate from the attacker training count K.

A.3 Utility, fidelity, and request isolation

The utility breakdowns use the same 3,198 items and operating points as Table 3. Fidelity uses prompts as its unit of evaluation, while request isolation compares paired cases.

Benchmark	Plain	Veil $a = 16$	Δ
ARC-Challenge	46.50	46.50	0.00
HellaSwag	43.70	43.20	-0.50
MMLU	38.01	37.82	-0.19
Overall	42.90	42.68	-0.22

Table 7: OLMoE multiple-choice utility on 3,198 items.

The aggregate paired 95% interval for Veil minus Plain is $[-0.66, +0.22]$ percentage points (McNemar $p = 0.4011$).

Request isolation. The 360-case study compares Plain-alone, Plain-mixed, Veil-alone, and Veil-mixed on the same victims. There are 325 cases with neither method changing, 11 Plain-only changes, 16 Veil-only changes, and 8 changes in both. The paired contingency test gives $p = 0.4421$.

Qwen utility and fidelity. All 24 Qwen routed MoE blocks use the envelope/packing/covariant-execution path. The always-on shared expert remains outside the routed envelope; its FFN is separately transformed and its gate is evaluated in the original semantic coordinates. The main utility table reports $a = 2, 12, 60$ separately.

On the same 3,198 utility items, Plain accuracy is 46.87%. Veil obtains 46.81% at $a = 2$ (-0.06 pp; paired 95% interval $[-0.47, 0.34]$; McNemar $p = 0.878$), 46.84% at $a = 12$ (-0.03 pp; $[-0.44, 0.41]$; $p = 1.0$), and 46.69% at $a = 60$ (-0.19 pp; $[-0.63, 0.25]$; $p = 0.488$). Differences are computed before rounding the displayed accuracies.

Table 8 expands the main fidelity columns on 256 prompts per setting. No setting produces NaN/Inf. A separate generation diagnostic gives exact matches on 27/32, 31/32, and 25/32 prompts at $a = 2, 12, 60$; these prompt-level counts are not the next-token agreement metric.

Configuration	Next-token agree.	Rel. L_2	KL
Qwen-Veil $a = 2$	97.35%	0.01923	0.265550
Qwen-Veil $a = 12$	97.23%	0.01921	0.256488
Qwen-Veil $a = 60$	97.07%	0.01919	0.268584

Table 8: Full 24-layer Qwen transformed-execution fidelity on 256 prompts per setting. Agreement and relative L_2 use the same displayed precision as the main table; KL is an additional diagnostic.

Qwen structural attack view. Table 9 reports the six layers observed by the routing-feature attacker: 3, 7, 11, 15, 19, and 23. The concatenated row combines these six views and reproduces the main-table result. This observed layer set is distinct from the 24-layer decoder used for utility and fidelity.

Routing view	Plain	StaticPerm	Veil $a = 12$	Veil $a = 60$
Layer 3	44.94	44.94	6.36	1.69
Layer 7	47.40	47.40	3.64	1.04
Layer 11	46.10	46.10	4.68	1.43
Layer 15	44.68	44.68	5.32	1.43
Layer 19	40.00	40.52	5.71	1.04
Layer 23	33.64	33.25	4.94	1.30
Six-layer concat.	69.87	69.87	16.75	3.12

Table 9: Banking77 routing-feature attack accuracy (%) on the same six observed Qwen layers and concatenated view as in the main table.

A.4 Canonical runtime diagnostics

Table 10 expands Table 4 using the same canonical measured run. Both Plain and VeilMoE use the matched grouped-expert backend on one 48G RTX 4090. Veil uses $a = 16$ and constant zero covers; Plain disables envelope padding and value transforms.

Prefill uses $B = 4, L = 512$, three warmups, and eight measured repeats. Cached decode uses history $H = 128$, one new row per request, $N = 1$ or 64 concurrent rows, two warmups, and eight measured repeats. The mean, p95, and slot expansion below are the same values as in the main table; kernel counts and peak memory expand those workloads.

The latency ratios are 7.10 for prefill, 2.36 for $N = 1$ decode, and 1.83 for $N = 64$ decode. The prefill ratio is close to the 6.63 slot expansion; decode overhead falls as concurrency rises.

Component attribution. For canonical prefill, grouped gate/up and down GEMMs account for 139 and 137 ms, gathers for about 40 ms, scatter-add merge for 39 ms, copies for 27 ms, and the fused signed FWHT for 18 ms. These CUDA categories sum to roughly 400 ms of the 1,066.92 ms wall-clock latency. The remainder includes other kernels and CPU-side envelope orchestration; the run also involves roughly 193 small device-to-host synchronizations.

The serving benchmark uses zero covers, while the value attacks use separate settings. The reported latency therefore describes the zero-cover configuration and does not quantify value or timing leakage.

Workload	Method	Mean (ms)	p95 (ms)	Kernels	ρ	Peak mem. (GiB)
Prefill $B = 4, L = 512$	Plain	150.26	156.34	3,063	1.00	25.44
	VeilMoE	1066.92	1084.55	3,591	6.63	27.85
Decode $N = 1$	Plain	74.05	75.51	2,804	1.00	24.95
	VeilMoE	174.49	193.55	3,316	7.25	25.11
Decode $N = 64$	Plain	100.03	100.66	2,868	1.00	27.91
	VeilMoE	182.83	195.71	3,380	2.78	28.07

Table 10: Runtime diagnostics on an RTX 4090, canonical $a = 16$ run. Plain and VeilMoE share the same grouped-expert backend.

A.5 Independent structural diagnostics

These studies examine which envelope features carry semantic signal and how capacity rules affect it. They use separate attack protocols, described with each result; Table 1 remains the reference for the primary standardized-logistic evaluation.

Decomposing the structural view. We train a three-seed MLP on touched-group identities, quantized capacities, the full envelope, or exact expert loads. The train/test splits contain 12,000/4,000 AG News requests and 10,003/3,080 Banking77 requests. Table 11 compares these views within this study. At $a = 16$, touched-group accuracy is at the uniform class prior, while capacities remain predictive. Capacity-only and full-envelope scores are close for this MLP.

Dataset	a	Touched	Capacity	Envelope	Exact load
AG News	2	69.29	89.26	88.30	90.28
	16	25.00	85.94	85.70	90.28
	64	25.00	68.67	68.32	90.28
Banking77	2	60.94	74.99	74.45	83.37
	16	1.30	28.74	28.39	83.37
	64	1.30	8.16	7.75	83.37

Table 11: Semantic accuracy (%) in the independent three-seed MLP view-decomposition study. Each row compares views under this study's protocol.

Figure 3 shows the same within-study comparison across group sizes. Exact loads remain an informative reference as the envelope views change.

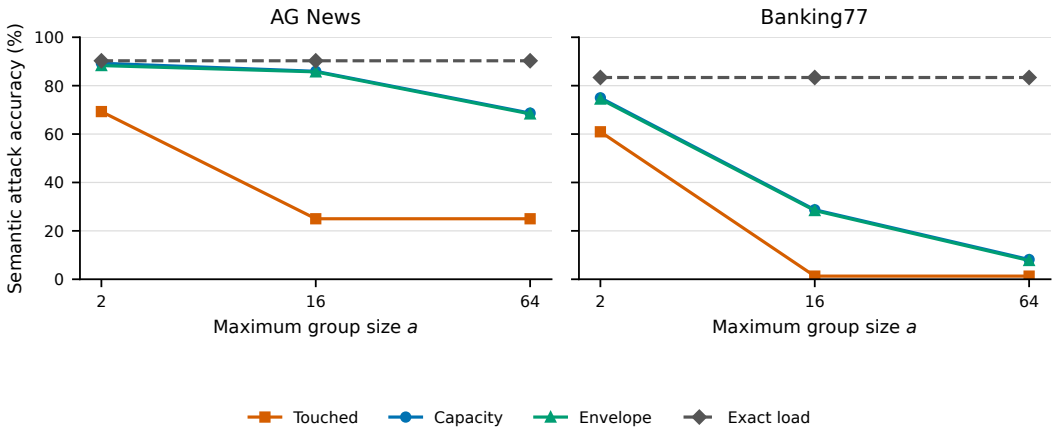

Figure 3: Independent structural-view study from Table 11. Each panel uses its own dataset's MLP cohort; the four curves differ only in the attacker's structural view within that cohort. Accuracy is lower when the tested attacker recovers less semantic information.

Table 12 compares logistic and MLP classifiers at $a = 16$ using the same splits. Its MLP rows repeat the corresponding rows of Table 11. Both classifiers recover more signal from capacities than from touched groups alone.

Dataset	Attacker	Touched	Capacity	Envelope	Exact load
AG News	Logistic	25.00	84.48	84.48	87.13
	MLP	25.00	85.94	85.70	90.28
Banking77	Logistic	1.30	27.66	27.63	83.73
	MLP	1.30	28.74	28.39	83.37

Table 12: Attacker-class comparison at $a = 16$ for the independent view-decomposition study. Accuracy is in percent.

Public-length control. A separate MLP study uses 6,000 training and 2,000 test requests per dataset. Length-only accuracy is 30.05% on AG News and 2.23% on Banking77. At $a = 16$, envelope-only accuracy is 85.97%/26.13%; adding length gives 85.53%/24.55%. On fixed-length subsets, envelope accuracy is 82.78% at length 54 for AG News and 20.61% at length 9 for Banking77. The fixed-length results show that envelope predictors still recover signal when length is held constant, although these subsets differ from the full test population.

Joint hierarchy–capacity exploration. Table 13 varies a and the capacity rule in a separate Banking77 study. It holds one calibrated CostAware hierarchy fixed, uses all-layer routing-load features, and has 10,003 training and 3,080 test requests. For each seed, we take the highest test accuracy over capacity/envelope views and logistic/MLP classifiers, then average those maxima over three seeds. These accuracies are optimistic attack summaries because the strongest combination is selected on the test set. We compare quantizers within this grid, whose cohort and aggregation differ from the primary evaluation.

	Quantizer Q: attack accuracy (%) / ρ			
a	Identity	Pow2	MultipleOf4	Fixed
8	54.50 / 3.05	45.58 / 3.84	36.85 / 4.24	17.11 / 6.41
16	39.09 / 3.81	29.02 / 4.94	27.87 / 4.90	**4.61 / 12.57**
32	25.53 / 4.64	15.66 / 6.13	17.20 / 5.75	2.01 / 25.04

Table 13: Independent fixed-hierarchy (a, Q) study on Banking77. Cells report the three-seed mean of the per-seed best attack accuracy and slot expansion. Bold marks the operating point discussed in the text, not the minimum in the table.

Figure 4 plots each table cell as attack accuracy against its own slot expansion. Lines connect the tested $a = 8, 16, 32$ settings within a quantizer; their positions are not measurements under the main table's protocol.

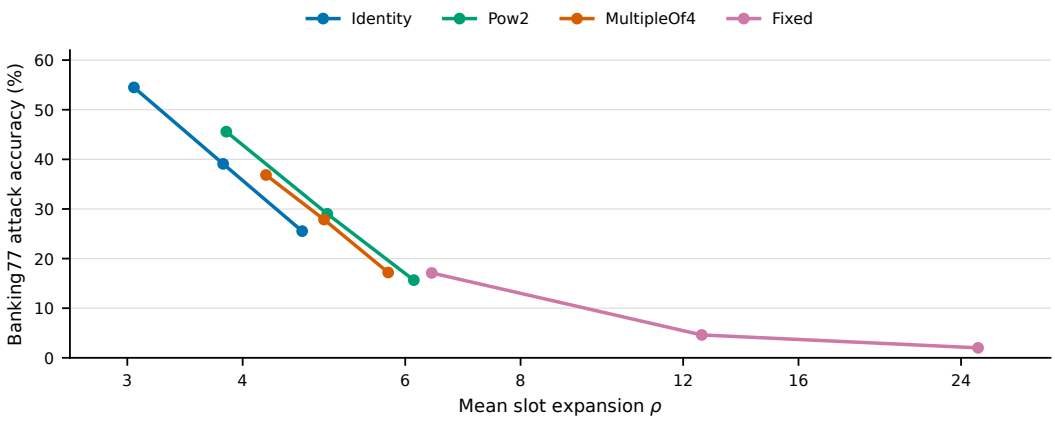

Figure 4: Banking77 cost and attack accuracy from the independent fixed-hierarchy grid in Table 13. Each line connects $a = 8, 16, 32$ within one quantizer, in that order. Both axes refer only to this grid's protocol.

Identity uses $Q(n) = n$, Pow2 rounds up to a power of two, and MultipleOf4 rounds up to a multiple of four. The Fixed rule is a capacity floor: $Q(n) = \max(n, a)$ for $n > 0$, with $Q(0) = 0$; it is not constant when $n > a$. Within this grid, changing from $(a = 8, \text{Identity})$ to $(a = 16, \text{Fixed})$ reduces the best-observed attack accuracy from 54.50% to 4.61% and raises ρ from 3.05 to 12.57. This changes both controls, so the difference is not an isolated effect of either a or Q.

Quantizer refinement. Numerically larger capacities do not necessarily disclose less information. Write $Q_i \preceq Q_j$ when Q_j is a deterministic function of Q_i on the evaluated integer-load support. Table 14 records this relation for the tested quantizers. The multiple-of-two, four, and eight rules form a refinement chain, whereas power-of-two rounding is incomparable with these rules beyond Identity. The data-processing inequality therefore does not order the attack accuracies of all pairs of quantizers.

$Q_i \preceq Q_j$	Id.	Mult2	Mult4	Mult8	Pow2
Identity	1	1	1	1	1
MultipleOf2	0	1	1	1	0
MultipleOf4	0	0	1	1	0
MultipleOf8	0	0	0	1	0
NextPowerOfTwo	0	0	0	0	1

Table 14: Partition-refinement diagnostic. Entry 1 means that the column quantizer is determined by the row quantizer on the evaluated load support.

A.6 Independent cover and sequence diagnostics

Distinguishing real and cover rows. Table 15 compares three cover configurations in an offline MLP row-classification study. It measures real/cover discrimination and recovery of expert loads, rather than the semantic accuracy measured by the raw-row attacker in Table 2. CONST and COV-P are highly separable in this study. Context-matched COV-X lowers row-classification AUROC and load-recovery R^2, but the AUROCs of 0.9143 and 0.9322 still show substantial distinguishability. These results do not imply that real and cover rows are indistinguishable.

Dataset	Cover	Real/Cover AUROC (MLP)	Hard R^2	Soft R^2	Soft RMSE (slots)
AG News	CONST	0.9979	0.9986	0.9987	0.1612
AG News	COV-P	0.9969	0.9927	0.9934	0.3615
AG News	COV-X	0.9143	0.1219	0.2027	3.9567
Banking77	CONST	0.9976	0.9989	0.9990	0.1751
Banking77	COV-P	0.9972	0.9956	0.9957	0.3561
Banking77	COV-X	0.9322	0.3572	0.4611	4.0085

Table 15: Real/cover row classification and expert-load recovery in the independent offline MLP study.

The serving benchmark uses constant zero covers, whereas COV-X is evaluated offline for row classification. The attack and runtime results describe these separate configurations.

Multi-tick envelope accumulation. Rekeying a value transform does not remove the public envelope history. We study cumulative decode envelopes on 2,000 Banking77 test requests with two training seeds. A sequence model consumes the per-tick history, while an MLP receives only the endpoint cumulative envelope. Table 16 compares these structural views within the sequence cohort; neither row reuses the primary single-tick attack baseline.

Banking77 view	$T = 1$	$T = 8$	$T = 16$	$T = 32$	$T = 128$
Envelope sequence	17.50	23.53	30.03	32.98	29.10
Envelope MLP	17.53	18.78	24.18	21.90	15.83
Best structural mean	17.53	23.53	30.03	32.98	29.10

Table 16: Independent multi-tick Banking77 study: accuracy (%), averaged over two seeds. The last row is the per-T maximum of the two displayed model means, not a detector chosen before testing. All displayed attackers use structural envelopes only.

Figure 5 traces the two tested structural attackers across decode ticks. The curves use the same cohort as Table 16.

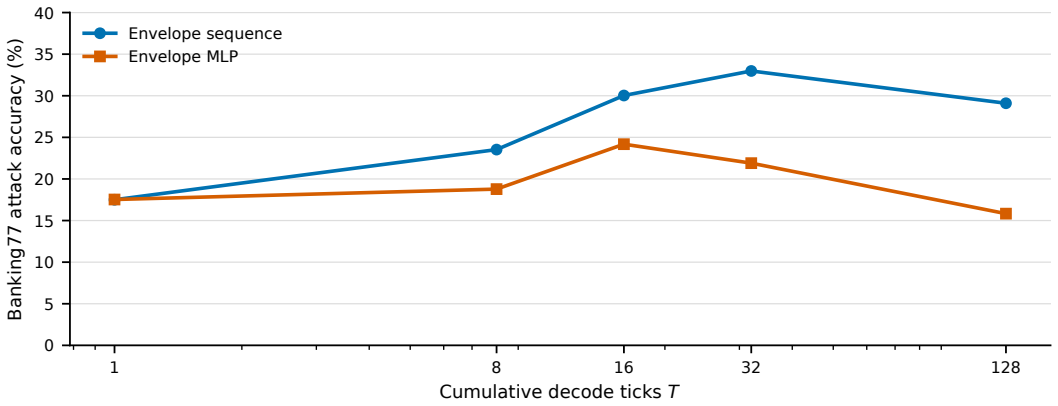

Figure 5: Banking77 attack accuracy from the independent multi-tick study in Table 16. The sequence attacker sees per-tick envelopes; the endpoint MLP sees the final cumulative envelope. Lines join the tested tick counts.

Sequence accuracy rises from 17.50% at $T = 1$ to 32.98% at $T = 32$, then falls to 29.10% at $T = 128$. The observed curve is therefore not monotone. It shows how this structural attacker uses envelope history across the tested ticks.

A.7 INDEPENDENT TIMING AND PUBLIC-STATE DIAGNOSTICS

Timing-side-channel control. This study records adapter wall-clock and GPU-kernel times with the $a = 16$ runtime on a separate cohort: 1,200 shadow-training, 400 shadow-validation, and 2,000 target-test requests per dataset, $B = 1$, maximum length 96, all 16 layers, and two training seeds. We compare timing-only features, a compact 144-d envelope, and their combination using this study's own envelope reference.

Dataset (prior)	Timing only	Envelope	Envelope + timing
Banking77 (1.30)	1.05 / 0.85	14.98	12.93
AG News (25.00)	43.15 / 31.38	83.13	82.48–83.13

Table 17: Accuracy (%) in the independent timing study. Timing-only entries are wall-clock / GPU time. Envelope-plus-timing reports the result or range across the tested timing-feature variants; it is not a confidence interval.

Adding timing features does not improve mean accuracy for the tested predictors. On Banking77 the joint model scores below the envelope-only model, though an attacker can always ignore the extra features. Timing alone predicts AG News above the uniform prior. Cache, TLB, PCIe, and hardware-counter channels are outside this timing measurement.

Public-state budgeted scheduling. We examine hierarchy selection in a 20,000-tick AG News/Banking77 trace replay. The scheduler uses only public phase, token count, and batch size. A measurement-calibrated cost model predicts the cost of each choice; the same model selects the hierarchy level and evaluates the replay. Table 18 reports these predictions, not per-tick execution times.

Budget (ms)	Mean a	Predicted mean (ms)	Predicted violation rate
800	58.4	283	0.00
1500	61.3	337	0.00
2500	63.6	386	0.00

Table 18: Cost-model replay of public-state hierarchy selection. Zero predicted violations describe decisions evaluated with the selection model; they are not measured serving-budget compliance.

Larger budgets permit larger mean group sizes under this model. The replay describes the scheduler's choices, while Table 4 reports measured latency for a separate serving configuration.

Logical metadata traces. An abstract trace prototype evaluates Veil-O and Veil-S on 433 requests (Table 19). The fixed-schedule profile produces one call sequence with 64 calls per request; the trace-indexed profile produces 432 distinct sequences. The prototype records logical calls rather than physical memory accesses or timing.

Profile	Distinct call traces	Calls / request
Veil-O	1	64.00 ± 0.00
Veil-S	432	32.97 ± 3.58

Table 19: Independent logical-trace prototype on 433 requests. Calls per request are reported as mean $\pm$ standard deviation.

A.8 PROOFS AND FORMAL DETAILS

Support-trace finiteness. Conditioned on fixed public row counts and finite M, k, $\tau_t^{\sup}$ is finite-valued, whereas the coefficient-bearing trace can be continuous. In particular,

$$I(X_t; \tau_t^{\sup} \mid U_t^- = u) \le N_t \log \left(\sum_{j=0}^{k} \binom{M}{j} \right),$$

which motivates using $\tau_t^{\sup}$ in the structural information bounds. Algorithm 1 gives the main-text semantic execution; the derivations below justify its hierarchy, covariance, and factorization claims.

Hierarchy construction and nonnegative merge penalty. Starting from H_0, each next level forms a deterministic minimum-weight matching over the current groups using the calibration cost and merge penalty in eqs. (9) and (10); matched pairs are merged, an unmatched group (if any) is carried forward, and ties are broken by a fixed public ordering. This procedure depends only on the public calibration workload. To see that the merge penalty is nonnegative, let $m_j = \max_{e \in G_j} n_e^{(s)}$ for two disjoint groups and suppose without loss of generality that $m_1 \ge m_2$. Then, sample by sample,

$$C_s(G_1 \cup G_2) - C_s(G_1) - C_s(G_2) = (|G_1| + |G_2|)Q(m_1) - |G_1|Q(m_1) - |G_2|Q(m_2) \quad (28)$$

$$= |G_2|\big(Q(m_1) - Q(m_2)\big) \ge 0. \quad (29)$$

Taking expectation over the public calibration sample gives $w(G_1, G_2) \ge 0$.

Monotone envelope coarsening and cost. Fix $\ell < L$ and a coarse group $G' \in H_{\ell+1}$. Let $G_1, \ldots, G_s \in H_\ell$ be the fine groups whose disjoint union is G', and define $m_j = \max_{e \in G_j} n_{t,e}$, using $m_j = 0$ for an untouched child. Monotonicity of Q implies

$$Q\left(\max_j m_j \right) = \max_j Q(m_j). \quad (30)$$

Indeed, if j^* attains the maximum of the m_j, then $Q(m_j) \le Q(m_{j^*})$ for all j. Therefore the coarse touched bit is the OR of the child touched bits and its disclosed capacity is the maximum of the child capacities. Applying this independently to every $G' \in H_{\ell+1}$ constructs the deterministic public map Φ_ℓ in eq. (25).

For cost monotonicity, let $m_* = \max_j m_j$. Then

$$|G'|Q(m_*) = \sum_{j=1}^{s} |G_j| Q(m_*) \ge \sum_{j=1}^{s} |G_j| Q(m_j). \quad (31)$$

Summing over coarse groups yields $S_t^{(\ell+1)} \ge S_t^{(\ell)}$; because the real contribution count r_t is level-independent, the same ordering holds for $\rho_t^{(\ell)}$ whenever $r_t > 0$. Since the coarse envelope is a deterministic post-processing of the fine envelope, conditional data processing gives the information inequality in Theorem 4. The result orders levels of one nested hierarchy with one fixed Q; it does not compare arbitrary partitions or quantizers.

Static relabeling preserves unlabeled routing structure. Let the symmetric group S_M act on expert labels of the support trace $\tau_t^{\sup}$. The canonical unlabeled trace $\mathrm{Can}(\tau_t^{\sup})$ is a chosen representative of the orbit $\{\sigma(\tau_t^{\sup}) : \sigma \in S_M\}$. For every fixed permutation π, $\pi(\tau_t^{\sup})$ belongs to the same orbit, so $\mathrm{Can}(\pi(\tau_t^{\sup})) = \mathrm{Can}(\tau_t^{\sup})$. Thus a reusable bijection changes expert names but not equality/repetition structure, load multiplicities, or cross-position expert sharing. This statement is structural; it does not claim that an unknown reusable key leaves mutual information numerically unchanged before conditioning on that key.

Exact expert covariance and request-local equivalence. For a real assignment (q, i, e), the accelerator receives $x_i^{(q)} AC_e$ up to row permutation. Under the transformed weights in eq. (15),

the gate and up branches are $x_i^{(q)} W_e^g \Pi_e$ and $x_i^{(q)} W_e^u D_e \Pi_e$. Because SiLU is elementwise, $\mathrm{SiLU}(a\Pi_e) = \mathrm{SiLU}(a)\Pi_e$, and because D_e is diagonal,

$$(\mathrm{SiLU}(a)\Pi_e) \odot (bD_e\Pi_e) = \big((\mathrm{SiLU}(a) \odot b)D_e\big)\Pi_e.$$

Multiplication by $\Pi_e^\top D_e^{-1} W_e^d A B_e$ cancels the permutation and diagonal scaling, yielding $E_e(x_i^{(q)})AB_e$. Applying B_e^{-1} therefore recovers $E_e(x_i^{(q)})A$.

A row permutation commutes with every deterministic row-wise expert, and evaluating consecutive chunks then concatenating them in public chunk order equals evaluating the full row matrix. After logical record restoration, covers are suppressed and each real record contributes exactly $\beta_{i,e}^{(q)} E_e(x_i^{(q)})A$. Hence

$$\widehat{y}_i^{(q)} = \sum_{e \in J_i^{(q)}} \beta_{i,e}^{(q)} E_e(x_i^{(q)})A = y_i^{(q)} A,$$

which proves Theorem 1. By eq. (4), two admitted executions that keep $X^{(q)}$, μ_q, the reference model, and A fixed finalize the same coefficients/support for q. Admitted-tick lossless execution then makes the semantic result independent of all co-resident request contents.

Accelerator structural factorization and exact envelope leakage. Fix a realization $U_t^- = u$. The envelope is a deterministic function of $(\tau_t^{\mathrm{sup}}, u)$ because exact loads are functions of the support trace and the group partition/quantizer are public. Once Env_t is fixed, the set of touched groups, every invoked expert within those groups, each logical bucket capacity, invocation order, and deterministic public chunk decomposition are fixed. Row-permutation randomness changes which logical record occupies a physical slot but not the structural fields included in $\mathsf{View}_{\mathrm{str}t}$. Therefore $\mathsf{View}_{\mathrm{str}t} = F_{\mathrm{str}}(\mathsf{Env}_t, u)$.

Conversely, under the protocol in Algorithm 1, all and only experts belonging to touched groups are invoked, and every expert in a touched group receives the same logical bucket capacity. Given the public partition/chunk rule in u, the observed expert IDs and total bucket/chunk shapes therefore recover each touched group and its capacity. Hence $\mathsf{Env}_t = G_{\mathrm{str}}(\mathsf{View}_{\mathrm{str}t}, u)$. Conditioned on U_t^-, Env_t and $\mathsf{View}_{\mathrm{str}t}$ are thus deterministic functions of each other, so

$$I(X_t; \mathsf{View}_{\mathrm{str}t} \mid U_t^-) = I(X_t; \mathsf{Env}_t \mid U_t^-).$$

Finally, Env_t is a deterministic function of τ_t^{sup} given U_t^-, which gives the remaining inequality in eq. (20). Using the finite-valued support trace avoids relying on a potentially infinite mutual-information quantity induced by continuous deterministic routing coefficients.

Veil-O trusted-access factorization. The subtlety is randomized or stateful oblivious memory access: an ORAM-style address trace need not be a deterministic function of (Env_t, U_t^-) alone, and a persistent hidden position-map/stash state may itself depend on earlier logical accesses. We therefore state Theorem 3 in terms of a whole-tick primitive contract. The fixed-domain variable Ω_t represents all fresh coins and any hidden primitive state entering the tick that the contract permits us to absorb; its conditional independence assumption is explicit rather than inferred from the word "ORAM." Envelope-sized permutations and randomized-access coins are deterministic derivations from this fixed-domain source, so random-object dimension does not depend on the private envelope.

Under the whole-tick contract preceding Theorem 3, the trusted execution can be partitioned into two phases. Before the envelope is released, router evaluation, Top-k/coefficient finalization, exact-load aggregation, and envelope construction follow schedules determined only by the public candidate shape in U_t^-. After release, the *entire* packing/read/select/scatter primitive is required to expose a trace determined only by (Env_t, U_t^-) and Ω_t, including the effect of any allowed hidden state. Fixed-size accelerator transfers and complete returned-chunk reads are part of this contract. Private cover synthesis, when used, must satisfy the same whole-tick dependence. Value-dependent aborts are excluded by the protocol-complete premise.

Thus there is a deterministic map F_{meta} for which

$$\mathsf{View}_{\mathrm{meta}t} = F_{\mathrm{meta}}(\mathsf{Env}_t, U_t^-, \Omega_t).$$

Since $\Omega_t \perp (X_t, \tau_t^{\mathrm{sup}}, \tau_t^{\mathrm{sem}}) \mid U_t^-$ and Env_t is a function of $(\tau_t^{\mathrm{sup}}, U_t^-)$, also $\Omega_t \perp (X_t, \tau_t^{\mathrm{sem}}, \mathsf{Env}_t) \mid U_t^-$. Marginalizing over Ω_t yields

$$\mathcal{L}(\mathsf{View}_{\mathrm{meta}t} \mid X_t, \tau_t^{\mathrm{sem}}, \mathsf{Env}_t, U_t^-) = \mathcal{L}(\mathsf{View}_{\mathrm{meta}t} \mid \mathsf{Env}_t, U_t^-), \tag{32}$$

so $X_t \to \mathsf{Env}_t \to \mathsf{View}_{\mathrm{meta}t}$ is a Markov chain conditional on U_t^-. The reverse deterministic relation needed for equality follows because $\mathsf{View}_{\mathrm{meta}t}$ contains $\mathsf{View}_{\mathrm{str}t}$, which recovers Env_t by Theorem 2; this proves eq. (22). Physical cache/TLB state, DRAM row-buffer effects, DMA timing, PCIe contention, and wall-clock duration are excluded by definition.

Logical-time multi-tick composition. For the logical-time composition statement, let the public execution plan be $P_t = p(\mathsf{Env}_t, U_t^-)$ and assume

$$U_{t+1}^- = \Gamma(U_t^-, P_t, E_t), \tag{33}$$

where E_t is an exogenous public logical-time event. Crucially, Γ does not consume measured completion time, hardware counters, tensor values, real/cover labels, or any finer routing trace. Under the single-tick Veil-O contracts, suppose additionally that the oblivious primitives are stateless/fresh-state across ticks, or that every persistent hidden-state update is itself a deterministic function only of the prior public envelope-derived plan and a global input-independent master tape. Under this extra sequence-level condition, induction gives a pathwise representation

$$\big((U_t^-, \mathsf{View}_{\mathrm{meta}t})\big)_{t=1}^{T} = F_T\big(U_1^-, \mathsf{Env}_{1:T}, E_{1:T}, \Omega_{1:T}\big) \tag{34}$$

for some deterministic F_T. The base case is Theorem 3; eq. (33) then fixes U_{t+1}^- from envelope-derived public state and E_t, closing the induction.

Equation 34 is a logical-transcript factorization, not a claim that physical inter-tick gaps are hidden. Generic persistent ORAM state whose update depends on fine private logical addresses is *not* covered merely by sampling fresh per-tick coins; it requires a separate sequence-level obliviousness theorem/contract. Nor, without an explicit model for adaptively chosen future inputs and public events, do we infer an additional multi-tick mutual-information bound. A scheduler that feeds measured duration or hardware feedback into future public state also falls outside this composition statement.

Perfect active-set hiding lower bound. Fix $U_t^- = u$ and let $\mathcal{K}(u)$ be the feasible family of active-expert sets. Perfect active-set hiding is stated pointwise over feasible private executions rather than through a conditional law on a possibly zero-probability event: there is one distribution μ_u such that the invoked-set distribution induced by the mechanism's internal randomness equals μ_u for every feasible private execution under u. Exactness additionally requires the execution's active set K to satisfy $K \subseteq A_{\mathrm{inv},t}$ almost surely. Because $A_{\mathrm{inv},t}$ ranges over the finite power set of $[M]$, every set in $\mathrm{supp}(\mu_u)$ has positive mass. Therefore, for any feasible execution with active set K, almost-sure containment under the same μ_u implies $K \subseteq A$ for every $A \in \mathrm{supp}(\mu_u)$. Applying this to every $K \in \mathcal{K}(u)$ gives

$$\bigcup_{K \in \mathcal{K}(u)} K \subseteq A \qquad \text{for every } A \in \mathrm{supp}(\mu_u).$$

Let $M_{\mathrm{feas}}(u)$ be the cardinality of this union. Here *assignment-preserving plaintext execution* means that the trusted component evaluates no Expert-FFN contribution itself, every required semantic assignment is mapped injectively to a server-side row evaluated by its required physical expert, and coded/composite experts, cross-assignment output reuse, and cryptographic oblivious parameter selection are excluded. Thus $A_{\mathrm{inv},t}$ counts only experts receiving at least one row slot. Every support element of μ_u schedules at least $M_{\mathrm{feas}}(u)$ rows, while injectivity independently gives $S_t^{\mathrm{mech}} \geq r_t$. Dividing by any realized $r_t > 0$ proves eq. (27). If every expert is active for some feasible input under the same u, then the union is $[M]$ and $M_{\mathrm{feas}}(u) = M$. Without the injectivity/no-reuse clause, the invoked-set conclusion remains valid but the per-assignment row-count corollary need not hold.

Bounded group-padding overhead. Fix ℓ and $r_t > 0$. For each touched group G, $m_{t,G} \geq 1$, so eq. (8) applies. The lower bound is pointwise:

$$\sum_{e \in G} n_{t,e} \leq |G| m_{t,G} \leq |G| Q(m_{t,G}) = |G| c_{t,G}. \tag{35}$$

Summing over touched groups gives $r_t \le S_t$. For the upper bound,

$$S_t = \sum_{G \text{ touched}} |G| Q(m_{t,G}) \tag{36}$$

$$\le \lambda \sum_{G \text{ touched}} |G| m_{t,G} \tag{37}$$

$$\le \lambda a_\ell \sum_{G \text{ touched}} m_{t,G} \tag{38}$$

$$\le \lambda a_\ell \sum_{G \text{ touched}} \sum_{e \in G} n_{t,e} = \lambda a_\ell r_t. \tag{39}$$

Dividing by r_t proves the slot-expansion part of Theorem 4. The symbol a_ℓ controls the worst-case multiplicative padding bound; it should not be interpreted as a guaranteed anonymity-set cardinality.