CASH: SENSITIVITY-AWARE CONVEX HEAD RECONSTRUCTION FOR PRIVATE LLM INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Secure multi-party computation (MPC) enables privacy-preserving LLM inference, but its latency and communication overhead remain substantial. Despite continued progress in secure operator design and computation reduction, standard multi-head attention still evaluates each head independently, repeatedly incurring the cost of secure attention-score computation and Softmax. Our analysis reveals a pronounced low-dimensional structure across post-Softmax attention maps and shows that these maps can be effectively reconstructed from a small subset of real attention heads. We introduce CASH, a training-free framework based on these findings. CASH explicitly computes attention maps for a set of real anchor heads and reconstructs the remaining maps through fixed convex combinations. CASH further uses head-level Fisher sensitivity to allocate a global anchor-head budget and compiles the selected configuration into a static MPC execution graph. Across LLaMA-1 7B and three OPT models, CASH achieves the highest average task accuracy in 11 of 12 model–budget settings and the lowest perplexity in all 12 among the evaluated baselines. Under the evaluated configuration, CASH achieves 1.47–$1.69\times$ secure inference speedups and reduces communication by 33.6–41.9% compared with dense attention.

1 INTRODUCTION

Large language models (LLMs) increasingly serve applications involving sensitive user data, including personal assistants, document analysis, and enterprise services. Conventional cloud inference exposes client inputs, while providers must protect proprietary model parameters, creating a two-sided privacy requirement. Secure multi-party computation (MPC) addresses this requirement by enabling joint LLM inference without revealing either party's private information (Rathee et al., 2024; Zeng et al., 2025). However, its computation and communication costs far exceed plaintext inference, creating a major bottleneck for practical deployment.

Existing private Transformer inference methods reduce MPC costs by optimizing secure protocols, approximating expensive nonlinear operators, or avoiding some attention computations (Hao et al., 2022; Li et al., 2023; Pang et al., 2024; Kei & Chow, 2025; Zhang et al., 2025; Zeng et al., 2026). Most focus on individual operator costs or computation along the token dimension. Standard multi-head attention nevertheless repeats expensive Q/K projections, attention-score computation, and secure Softmax across heads. Exploiting cross-head structural redundancy to directly reduce the attention paths evaluated exactly in MPC remains underexplored. Our profiling identifies multi-head attention as a key optimization target. In a LLaMA-7B decoder block at input length $S = 512$, the Q/K–score–Softmax path accounts for 68.3% of full-block communication and 95.7% of attention communication (Figure 1(b)). Reducing exactly evaluated heads from 32 to 4 at $S = 512$ decreases this communication from 17.05 GB to 2.19 GB (Figure 1(c)), highlighting cross-head redundancy as an opportunity to reduce private-inference costs.

Attention-head pruning, sharing, and selection exploit cross-head redundancy in plaintext inference (Michel et al., 2019; Voita et al., 2019; Agarwal et al., 2024; Liao & Vargas, 2024), but their computation patterns are not naturally suited to MPC. Input-dependent grouping or selection uses inexpensive plaintext control flow, yet requires secure similarity computation, comparison, selection, and routing in MPC. In our MPC evaluation of CHAI, dynamic control increases latency by 9.8% and interaction rounds by $49.3\times$ relative to Dense; secure $\arg\min$ contributes most of the extra overhead

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

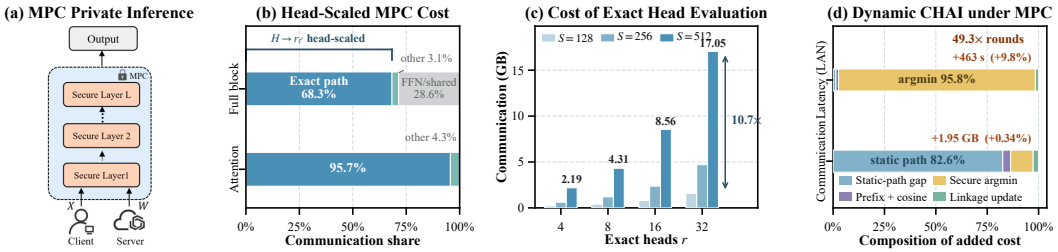

Figure 1: **Motivation for MPC-friendly head reduction.** (a) Two-party private inference protects client inputs and model parameters. (b) At input length $S = 512$, head-scaled Q/K, score, and Softmax evaluation dominates secure-attention communication. (c) Exact-path communication grows with the evaluated-head count r and sequence length S. (d) Dynamic CHAI introduces additional secure similarity, selection, and linkage operations during private inference.

(Figure 1(d)). Fewer participating heads alone therefore cannot guarantee MPC acceleration: the reduction mechanism must also avoid data-dependent online control.

We therefore propose **CASH**, a static cross-head attention reconstruction method for private MPC inference. CASH evaluates a small set of real *anchor heads* and reconstructs the remaining post-Softmax maps as fixed convex combinations of these anchors. Selecting anchors from the original model heads eliminates non-anchor Q/K projections, scores, and Softmax while retaining all logical value branches and the original output projection. Calibration determines anchors and coefficients offline, and a head permutation maps them to fixed execution positions. Online inference requires no private head selection or input-dependent routing and compiles directly into a static MPC graph.

- We introduce cross-head convex reconstruction using real anchor heads. It exploits structural redundancy in post-Softmax attention maps to reduce independent Q/K, score, and Softmax evaluations while preserving the original value and output branches.
- We develop a Head-Fisher-aware global budget allocator that jointly accounts for head-level reconstruction residuals and task sensitivity to distribute a limited evaluated-head budget across Transformer layers. Combined with offline head permutation, the entire online computation remains static and independent of the private input.
- We evaluate CASH across four models and multiple anchor budgets, measuring both predictive quality and secure-inference cost. CASH achieves the highest average accuracy in 11 of 12 matched-budget settings and the lowest perplexity in all 12, with a tunable trade-off between accuracy retention and MPC acceleration.

2 Motivation

Observation 1: Attention maps have low-dimensional cross-head structure.

Post-Softmax attention probabilities exhibit a pronounced low-dimensional structure across heads. Although LLaMA-1 7B contains 32 attention heads per layer, the resulting head-wise attention maps span a space with substantially lower effective dimension. Across all layers and the C4, WikiText-2, and BookCorpus corpora (Zhu et al., 2015), the median Rank@90 is only 3; most intermediate layers require only 1–7 singular components to capture 90% of the squared energy.

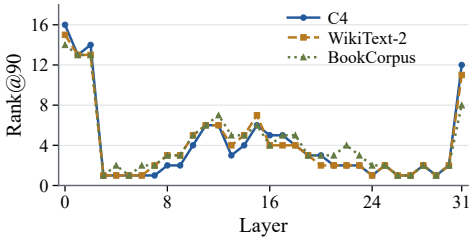

Figure 2: **Cross-head effective rank.** Layer-wise Rank@90 in LLaMA-1 7B.

More importantly, layer-wise Rank@90 profiles agree closely across corpora (Figure 2), suggesting stable cross-head structure beyond one calibration distribution. Despite many independently computed heads, their outputs concentrate in a space whose

dimension is far below the head count. This directly motivates reducing repeated per-head secure computation.

Observation 2: Real attention heads can represent this low-dimensional structure.

Low effective rank alone does not directly translate into fewer executable attention heads. While SVD provides an optimal low-rank approximation, its latent basis vectors are synthetic and do not correspond to actual pretrained heads. Consequently, SVD compression cannot be directly implemented by selectively evaluating a subset of the original Q/K and Softmax computations.

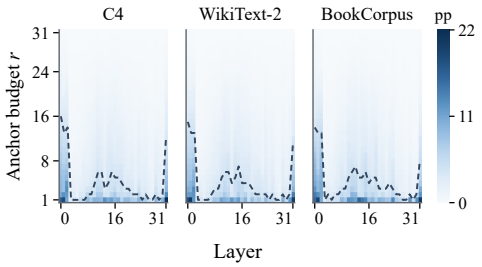

Figure 3: **Real-head reconstruction versus SVD.** Excess normalized squared error at matched r (pp); dashed curves mark Rank@90. Each corpus is fitted and evaluated on its own calibration data.

We find that a small subset of real attention heads can instead serve as reconstruction bases: the remaining attention maps can be accurately approximated through convex combinations of these anchor heads. Figure 3 compares real-head convex reconstruction against matched-rank SVD. Using 25% and 50% of heads as anchors, the median excess normalized squared error over matched-rank SVD is only 2.26 and 1.09 percentage points, respectively. With a 50% anchor budget, the median absolute normalized squared reconstruction error is only 3.54%. These results show that real heads preserve low-dimensional structure via fixed convex reconstruction, enabling MPC-friendly secure attention reduction.

3 METHOD

CASH decouples cross-head reconstruction from layer-wise budget allocation. Offline, it constructs real-anchor candidates for each layer, assesses their task impact through Head-Fisher-weighted residuals, and selects anchor counts under a global budget. Head permutation compiles the configuration into a fixed MPC execution path (Figure 4).

Setup and privacy boundary. For a frozen Transformer with L layers and H heads per layer, CASH selects anchor counts r_ℓ, sets S_ℓ, and coefficients C_ℓ subject to $\sum_{\ell=1}^{L} r_\ell = B$ and $1 \le r_\ell \le H$. The retention ratio is $\beta = B/(LH)$; B counts score–Softmax paths evaluated at a fixed input shape.

We consider semi-honest two-party MPC inference. Inputs, model parameters, coefficients, and activations remain private; the designated output goes to the client. Architecture, tensor shapes, the mask, and per-layer anchor counts are public. The owner calibrates and permutes heads offline without separately disclosing original anchor indices. Security and preprocessing assumptions follow the MPC backend; the supplementary PUMA evaluation inherits its three-party assumptions (Appendix B).

3.1 CONVEX HEAD RECONSTRUCTION

Given an anchor budget r_ℓ for layer ℓ, CASH reconstructs heads at the post-Softmax attention-probability interface. With head dimension d_h and causal mask $\mathcal{M}$, standard attention computes

$$P_{\ell,h} = \mathrm{Softmax}\left(\frac{Q_{\ell,h}K_{\ell,h}^{\top}}{\sqrt{d_h}} + \mathcal{M}\right), \qquad O_{\ell,h} = P_{\ell,h}V_{\ell,h}, \tag{1}$$

For real anchors $S_\ell = (a_1, \ldots, a_{r_\ell})$ and coefficients $C_\ell \in \mathbb{R}^{H \times r_\ell}$, CASH reconstructs

$$\widehat{P}_{\ell,h}(x) = \sum_{j=1}^{r_\ell} c_{\ell,hj} P_{\ell,a_j}(x), \qquad c_{\ell,hj} \ge 0, \quad \sum_{j=1}^{r_\ell} c_{\ell,hj} = 1. \tag{2}$$

For anchor heads, the corresponding coefficient rows are fixed to one-hot vectors. Because all heads share the same causal mask, convex combinations preserve nonnegativity, row normalization, and

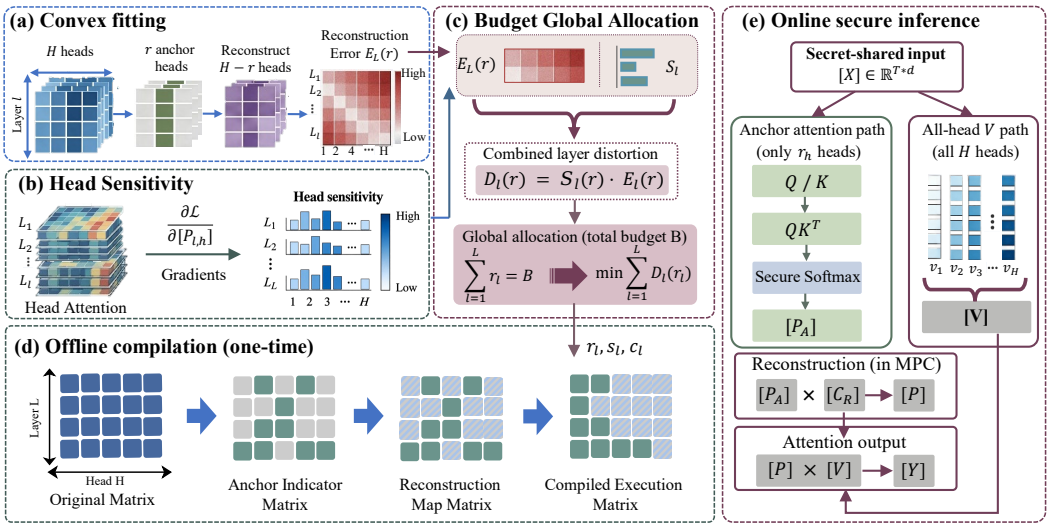

Figure 4: **CASH overview.** Offline calibration constructs real-anchor candidates and estimates Head Fisher; global allocation determines layer-wise anchor budgets, and head permutation maps the final configuration to fixed execution positions. Online MPC explicitly evaluates anchor attention and reconstructs the remaining attention maps while retaining all value and output branches.

causal support, so reconstruction requires no additional Softmax. Single-head sharing is the one-hot special case of this formulation.

Each logical head retains its original value branch, $\widehat{O}_{\ell,h} = \widehat{P}_{\ell,h} V_{\ell,h}$, followed by the original head concatenation and output projection. CASH therefore compresses the generation of independent attention maps while retaining the output branch associated with each head.

Anchors and reconstruction coefficients. We flatten and concatenate valid causal calibration probabilities into $A_\ell \in \mathbb{R}^{H \times M}$. A nested ordering supports different budgets: the first anchor is farthest from the head centroid; each subsequent anchor is farthest from the current anchor convex hull. Distances use the Gram matrix $G_\ell = A_\ell A_\ell^\top$, avoiding explicit operations on high-dimensional attention vectors.

The first r heads in this ordering form the candidate anchor set $S_{\ell,r}$. For each candidate, we solve

$$C_{\ell,r} \in \underset{C \in \mathcal{C}(S_{\ell,r})}{\arg\min} \left\| A_\ell - C A_{\ell,S_{\ell,r}} \right\|_F^2, \tag{3}$$

where $\mathcal{C}(S)$ enforces simplex and anchor-identity constraints. Small convex quadratic programs depending only on G_ℓ yield cached anchors, coefficients, and per-head residuals for each layer–budget pair. Head Fisher assesses the task importance of these errors during allocation, leaving candidate construction unchanged. Calibration and probability-preservation details appear in Appendices A and A.3.

3.2 From Reconstruction Error to Budget Allocation

Given each layer's reconstruction candidates, how should we allocate the limited global anchor budget? A direct strategy assigns more anchors to higher-rank layers using Rank@90. However, Rank@90 measures attention-map compressibility without capturing the effect of reconstruction perturbations on predictions. Our preliminary experiments show substantial accuracy loss from Rank@90-based allocation, while different layer-wise budgets with the same reconstruction mechanism substantially improve accuracy (Appendix C). Allocation should therefore account for head-level task sensitivity.

Reconstruction residuals measure how much an attention map changes, whereas budget allocation concerns how that change affects model loss. Inspired by Fisher-weighted compression (Hsu et al., 2022), we use head-level Fisher information to weight reconstruction residuals, jointly capturing reconstructability and task sensitivity.

Figure 5 compares proxy scores with measured next-token NLL increases for 128 single layer candidates, covering $r \in \{4, 8, 16, 24\}$ across 32 layers while keeping all other layers unchanged. The diagonal indicates perfect agreement between proxy and measured damage rankings. On the first eight C4 calibration blocks, adding Head Fisher to reconstruction residuals raises Spearman correlation from 0.486 to 0.866. We therefore use Fisher-weighted residuals as the allocation cost.

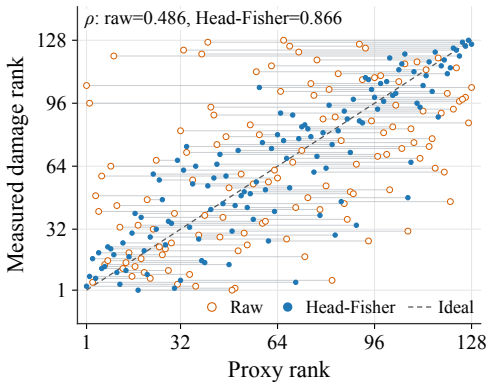

Figure 5: **Head Fisher improves candidate ranking.** Proxy ranks versus measured single-layer NLL-increase ranks. Head Fisher raises Spearman ρ from 0.486 to 0.866.

3.3 Head-Fisher-Aware Distortion

CASH scores each reconstruction candidate using two complementary quantities: how much each head's attention map changes and how sensitive the loss is to that change. Both are calibrated against the original frozen model. Let $N = |\mathcal{D}_{\text{cal}}|$, $\widehat{A}_{\ell}^{(r)} = C_{\ell,r} A_{\ell,S_{\ell,r}}$, and $M = \sum_{x \in \mathcal{D}_{\text{cal}}} |\Omega_x|$, where Ω_x contains valid causal coordinates. We measure the squared residual $e_{\ell,h}(r)$ and estimate Head Fisher $s_{\ell,h}$ from the summed next-token cross-entropy $\mathcal{L}(x)$:

$$e_{\ell,h}(r) = \frac{1}{N} \left\| A_{\ell,h} - \widehat{A}_{\ell,h}^{(r)} \right\|_2^2, \tag{4}$$

$$s_{\ell,h} = \frac{1}{M} \sum_{x \in \mathcal{D}_{\text{cal}}} \left\| \nabla_{P_{\ell,h}(x)} \mathcal{L}(x) \right\|_{\Omega_x}^2 . \tag{5}$$

Gradients are collected at the post-Softmax, pre-PV interface, without constructing a full Fisher matrix or updating model parameters. The residual varies with the candidate budget r, whereas sensitivity is computed once per head and reused across candidates. Their combination gives the layer–budget distortion

$$D_{\ell}(r) = \sum_{h=1}^{H} s_{\ell,h}\, e_{\ell,h}(r). \tag{6}$$

A sensitive head contributes little when it is already reconstructed accurately; additional anchors are valuable when they reduce large residuals on sensitive heads. Thus $D_{\ell}(r)$ supplies a common allocation cost that accounts for both reconstructability and task sensitivity, without changing anchor search or coefficient fitting. It is a sensitivity-weighted proxy rather than an exact loss increase. Appendix A.2 specifies normalization.

3.4 Global Budget Allocation

The cached distortions turn budget allocation into a finite optimization problem: choosing $r_{\ell} \in \mathcal{R}_{\ell} \subseteq \{1, \ldots, H\}$ also selects that layer's anchors and coefficients. CASH minimizes total distortion subject to exactly B evaluated heads:

$$\min_{r_1,\ldots,r_L} \sum_{\ell=1}^{L} D_{\ell}(r_{\ell}) \quad \text{subject to} \quad \sum_{\ell=1}^{L} r_{\ell} = B, \qquad r_{\ell} \in \mathcal{R}_{\ell}. \tag{7}$$

Rather than choosing each layer independently, we solve this additive objective offline by dynamic programming. Let $F_{\ell}(b)$ be the minimum distortion for the first ℓ layers using exactly b anchors. With $F_0(0) = 0$ and infeasible states set to $+\infty$,

$$F_{\ell}(b) = \min_{\substack{r \in \mathcal{R}_{\ell} \\ r \le b}} \left\{ F_{\ell-1}(b - r) + D_{\ell}(r) \right\}. \tag{8}$$

Backtracking from $F_L(B)$ retrieves the selected budgets, anchors, and coefficients. This couples layer-wise choices: additional anchors are assigned by their effect on total distortion, not by sensitivity or rank alone. With every head count available, this takes $O(LBH)$ state transitions. Optimality is with respect to the cached candidate set and additive distortion objective (Appendix A.2).

Static MPC execution. The owner compiles the selected configuration before secret-sharing it: each layer's r_ℓ anchors are permuted into its first r_ℓ positions, with consistent reordering of Q/K/V blocks, the output projection, and coefficients. Public layer-wise counts then fix the online path, avoiding input-dependent similarity computation, head selection, or routing.

Online inference evaluates Q/K, scores, and secure Softmax only for anchors. For one layer, flatten the anchor maps into $\mathbf{P}_S \in \mathbb{R}^{r\times m}$ and collect non-anchor coefficient rows in $C_R \in \mathbb{R}^{(H-r)\times r}$, where m counts materialized query–key positions. Writing $[\cdot]$ for secret sharing,

$$[\widehat{\mathbf{P}}_R] = [C_R]\,[\mathbf{P}_S], \qquad [\widehat{O}_h] = [\widehat{P}_h]\,[V_h]. \tag{9}$$

Both coefficients and attention maps remain private, so reconstruction still requires secure matrix multiplication. The resulting static graph replaces non-anchor Q/K, score, and Softmax paths while retaining all value branches and the original output projection.

4 Evaluation

4.1 Experimental Setup

Models and baselines. We evaluate LLaMA-1 7B (Touvron et al., 2023) and OPT-1.3B/2.7B/6.7B (Zhang et al., 2022). Baselines include Dense attention, Michel head pruning (Michel et al., 2019), CHAI (Agarwal et al., 2024), Shared Attention (SA) (Liao & Vargas, 2024), and MARILL head merging (HM) (Rathee et al., 2024). All methods are evaluated with frozen model weights, without fine-tuning or distillation. We use budgets $\beta \in \{50\%, 25\%, 12.5\%\}$, defined as the retained fraction of Dense score–Softmax paths.

Calibration and quality metrics. Unless stated otherwise, calibration uses 128 C4 sequences of 1024 tokens (Raffel et al., 2020). We evaluate zero-shot performance on PIQA, HellaSwag, ARC-Challenge, ARC-Easy, and BoolQ (Bisk et al., 2020; Zellers et al., 2019; Clark et al., 2018; 2019), reporting five-task average accuracy (Avg.) and WikiText-2 perplexity (PPL) (Merity et al., 2017). Task-level plots use normalized accuracy for the first four tasks and accuracy for BoolQ. Quality retention is defined as $q = \mathrm{Avg.}/\mathrm{Avg.}_{\mathrm{Dense}}$.

System Setup The main experiments use two-party SHAFT (Kei & Chow, 2025) with batch size one and sequence lengths $S \in \{128, 256, 512\}$. The two servers each provide 25 vCPUs on an Intel Xeon Platinum 8481C processor and 90 GB RAM, connected over a LAN with 1-Gbps bandwidth and 0.5-ms RTT. Offline calibration uses an NVIDIA RTX 4090 GPU. We report end-to-end latency in seconds and communication in decimal GB. We also evaluate different secure inference backends under their standard deployment configurations. Further experimental details and supporting analyses are provided in Appendix D.

4.2 Quality at Matched Budgets

Cross-model quality. Table 1 shows that CASH achieves the highest Avg. in 11 of 12 model–budget settings and the lowest PPL in all 12 among the evaluated compression methods. At 50%, CASH retains 88.8–93.6% of Dense Avg. across the four models, corresponding to absolute losses of 3.25–6.67 percentage points. On LLaMA-1 7B, its Avg. is 64.33%, exceeding the strongest compressed baseline by 12.01 points. At 25%, CASH leads the strongest baseline by 1.11–10.71 points across models, but its Avg. remains 11.28–19.69 points below Dense and its PPL is 4.36–$12.18\times$ Dense PPL. The only Avg. exception is OPT-2.7B at 12.5%, where CASH scores 34.33% versus CHAI's 34.73%, although CASH has lower PPL (465.24 versus 640.80). Across models, CASH preserves substantially more predictive quality than the evaluated compression baselines at matched attention budgets, with the largest advantages appearing at moderate compression levels.

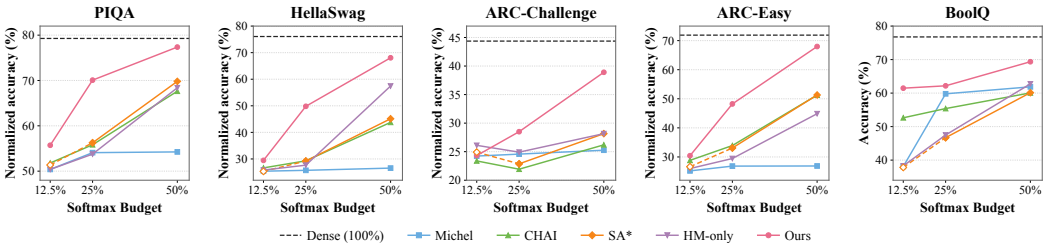

Figure 6: Task-level quality on LLaMA-1 7B across compression budgets.Dashed lines denote Dense.

Table 1: **Model quality across compression budgets.** Avg. is the average accuracy (%) over five downstream tasks; PPL is WikiText-2 perplexity.

Model	Budget	Michel		CHAI		SA		HM		Ours	
		Avg. (%)↑	PPL↓	Avg. (%)↑	PPL↓	Avg. (%)↑	PPL↓	Avg. (%)↑	PPL↓	Avg. (%)↑	PPL↓
LLaMA-1 7B	Full (1024)	69.67	5.63	69.67	5.63	69.67	5.63	69.67	5.63	69.67	5.63
	50% (512)	38.96	17.19	49.78	32.84	50.90	24.78	52.32	17.51	**64.33**	**9.85**
	25% (256)	38.20	129.01	39.27	238.46	37.62	230.00	36.66	749.05	**49.98**	**24.54**
	12.5% (128)	32.66	909.94	36.65	519.12	33.22	1097.14	33.32	561.61	**40.28**	**104.70**
OPT-1.3B	Full (768)	50.98	14.52	50.98	14.52	50.98	14.52	50.98	14.52	50.98	14.52
	50% (384)	44.02	125.08	43.83	49.96	38.50	112.76	44.66	58.97	**47.73**	**24.48**
	25% (192)	38.01	318.44	36.68	272.66	34.02	779.19	38.59	788.49	**39.70**	**69.79**
	12.5% (96)	34.70	540.80	34.46	865.08	32.42	1205.17	34.46	1365.19	**35.99**	**302.69**
OPT-2.7B	Full (1024)	54.87	12.39	54.87	12.39	54.87	12.39	54.87	12.39	54.87	12.39
	50% (512)	44.12	34.51	44.45	51.36	40.38	98.98	49.77	44.83	**51.19**	**24.71**
	25% (256)	36.89	458.31	37.48	329.50	33.79	863.60	36.86	338.20	**42.44**	**150.88**
	12.5% (128)	34.58	730.77	**34.73**	640.80	31.95	1410.18	32.32	1167.87	34.33	**465.24**
OPT-6.7B	Full (1024)	59.62	10.78	59.62	10.78	59.62	10.78	59.62	10.78	59.62	10.78
	50% (512)	47.57	45.45	50.51	31.48	43.76	79.35	51.93	30.43	**52.95**	**27.81**
	25% (256)	36.95	483.58	39.06	248.98	32.24	638.30	35.43	355.68	**43.95**	**122.71**
	12.5% (128)	34.59	782.84	34.65	655.36	31.83	1120.89	32.53	1131.69	**35.30**	**636.84**

Task-level quality. Figure 6 provides a complementary task-level view on LLaMA-1 7B. CASH leads the compressed baselines on all five tasks at both 50% and 25%, indicating that the aggregate advantage is consistent across different capabilities rather than driven by a single task. The response to budget reduction nevertheless varies across tasks: HellaSwag and ARC-Easy show larger declines from 50% to 25%, whereas PIQA and BoolQ are more stable. BoolQ remains comparatively robust at 12.5%, where CASH continues to lead the compressed baselines.

At 12.5%, the performance gaps narrow on HellaSwag and the ARC tasks, and HM slightly exceeds CASH on ARC-Challenge. CASH achieves a LLaMA-1 7B Avg. of 40.28% versus Dense's 69.67%, with WikiText-2 PPL increasing from 5.63 to 104.70. Even at this most constrained budget, CASH retains the lowest PPL among the compressed baselines. The task-level view therefore shows a consistent advantage at 50% and 25%, while differences among compressed methods become smaller as the attention budget is further reduced.

4.3 Private-Inference Efficiency and Quality–Cost Trade-off

Higher-retention operating point. Figure 7(b) highlights a higher-retention configuration on LLaMA-1 7B. At a 50% budget, CASH achieves 64.33% Avg. versus Dense's 69.67%, retaining 92.0% of the five-task mean with a 5.34-point absolute loss. Meanwhile, the corresponding configuration reduces end-to-end secure inference latency by 32.0% at $S = 512$. WikiText-2 PPL increases from 5.63 to 9.85 (Table 1). These results show that CASH preserves most of the Dense model's predictive quality while providing substantial secure-inference savings, forming a practical high-retention operating point on the quality–cost trade-off.

Matched-budget comparison. At $S = 512$ and $\beta = 50\%$, CASH achieves 1.47–1.69× secure inference speedups and 33.6–41.9% lower communication than Dense (Table 2). Mean plaintext accuracy retention is 92.2% across four models, versus 73.9% for SA at nearly identical cost: latency and communication are each less than 0.7% higher than SA's. These results establish better quality at

comparable measured cost, not lossless preservation of Dense model quality at this budget. The costs include the secure matrix multiplications required for private reconstruction, so the reported savings are net of reconstruction overhead.

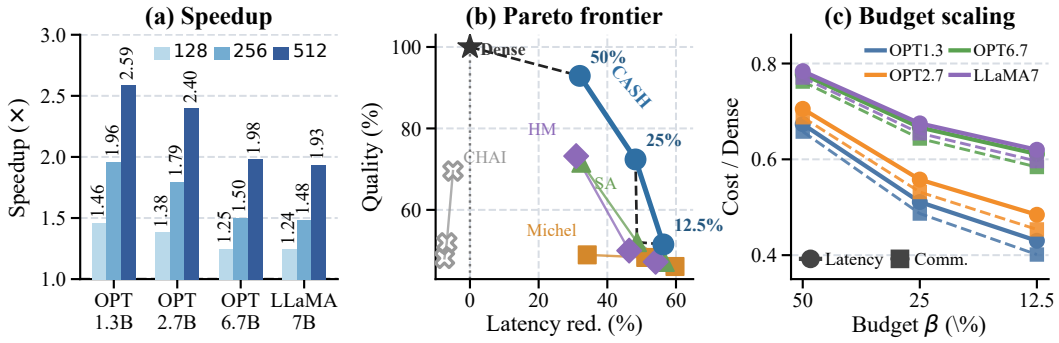

Figure 7: **Secure inference efficiency and quality–cost trade-off.** (a) CASH latency speedup at $\beta = 25\%$ across models and sequence lengths. (b) Quality retention versus latency reduction on LLaMA-1 7B at $S = 512$. (c) Latency and communication at $S = 256$, normalized to Dense.

Table 2: **Secure-inference efficiency at $S = 512$ and $\beta = 50\%$.** Parentheses give latency speedups or communication-reduction factors relative to SHAFT.

Method	$\overline{Q/Q_{\text{Dense}}}$ ↑	OPT-1.3B		OPT-2.7B		OPT-6.7B		LLaMA-1-7B	
		Latency (s)↓	Comm. (GB)↓	Latency (s)↓	Comm. (GB)↓	Latency (s)↓	Comm. (GB)↓	Latency (s)↓	Comm. (GB)↓
SHAFT	100.0%	3896 (1.00×)	473.2 (1.00×)	5508 (1.00×)	663.1 (1.00×)	6690 (1.00×)	782.3 (1.00×)	6856 (1.00×)	802.6 (1.00×)
Michel	75.6%	2265 (1.72×)	271.1 (1.75×)	3302 (1.67×)	390.9 (1.70×)	4349 (1.54×)	498.8 (1.57×)	4514 (1.52×)	519.1 (1.55×)
CHAI	80.9%	4152 (0.94×)	475.5 (0.995×)	5848 (0.94×)	666.2 (0.995×)	7029 (0.95×)	785.8 (0.996×)	7195 (0.95×)	806.1 (0.996×)
SA	73.9%	2294 (1.70×)	274.3 (1.73×)	3354 (1.64×)	396.4 (1.67×)	4467 (1.50×)	509.6 (1.54×)	4633 (1.48×)	529.8 (1.51×)
HM	85.1%	2310 (1.69×)	275.1 (1.72×)	3390 (1.62×)	398.9 (1.66×)	4570 (1.46×)	518.2 (1.51×)	4735 (1.45×)	538.4 (1.49×)
CASH	**92.0%**	**2302 (1.69×)**	**274.7 (1.72×)**	**3367 (1.64×)**	**397.0 (1.67×)**	**4482 (1.49×)**	**510.6 (1.53×)**	**4663 (1.47×)**	**533.1 (1.51×)**

$\overline{Q/Q_{\text{Dense}}}$ is the mean plaintext quality retention across the four models.

Sequence-length scaling. At a 25% budget, speedup rises from 1.24–1.46× at $S = 128$ to 1.93–2.59× at $S = 512$ (Figure 7(a)). For each non-anchor head, CASH replaces md_h score-product multiplications with mr reconstruction multiplications and eliminates its secure Softmax evaluation, where m is the number of query–key positions. As sequence length increases, the cost of these attention operations grows more rapidly than token-wise components, making the savings from head reconstruction increasingly significant. Consistently, reducing the anchor budget further lowers both latency and communication across all four models (Figure 7(c)).

Cross-protocol evaluation. We further evaluate CASH across different secure-inference backends using PUMA (3PC) (Dong et al., 2025; Mohassel & Rindal, 2018) and BumbleBee (2PC) (Lu et al., 2025; Huang et al., 2022). Experiments use the complete LLaMA-1 7B attention module at $S = 8$ with global budgets $B = 512$, 256, and 128. As shown in Table 3, CASH reduces both latency and communication at every evaluated budget on both backends. On PUMA, it achieves 1.65–1.95× latency speedups and 1.49–2.07× communication-reduction factors; on BumbleBee, the corresponding ranges are 1.27–1.53× and

Table 3: Cross-protocol cost on LLaMA-1 7B at $S = 8$.

	PUMA (3PC)		BumbleBee (2PC)	
B	Lat. (s)	Comm. (GB)	Lat. (s)	Comm. (GB)
Dense	163.86 (1.00×)	0.510 (1.00×)	264.49 (1.00×)	1.461 (1.00×)
512	99.56 (1.65×)	0.343 (1.49×)	208.92 (1.27×)	1.353 (1.08×)
256	**84.20** (1.95×)	0.277 (1.84×)	184.27 (1.44×)	1.296 (1.13×)
128	85.50 (1.92×)	**0.246** (2.07×)	**173.41** (1.53×)	**1.266** (1.15×)

1.08–1.15×. The variation in gains reflects differences in the cost composition of the underlying protocols. These results indicate that CASH's efficiency gains transfer across both 2PC and 3PC backends rather than being specific to SHAFT.

4.4 Ablation and Robustness

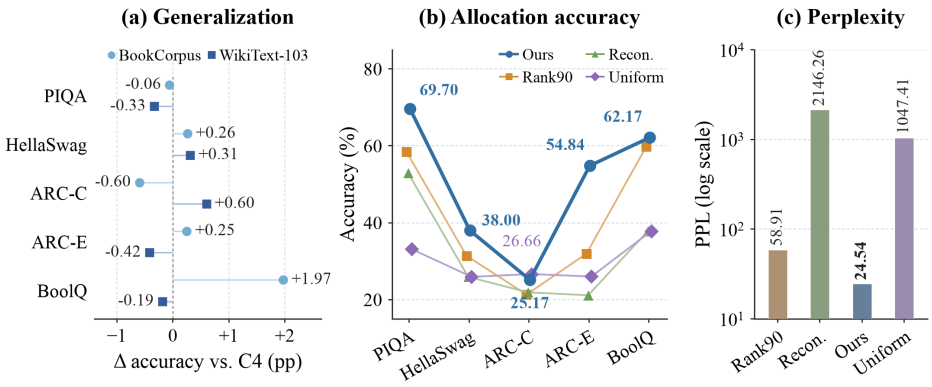

Figure 8: **Allocation ablation and calibration robustness on LLaMA-1 7B.** (a) Task-wise accuracy differences from C4 calibration for BookCorpus and WikiText-103 at a 50% budget. (b) Task accuracy and (c) WikiText-2 perplexity under different allocation strategies at $B = 256$.

Allocation strategy. We evaluate CASH's layer-wise budget allocator at a fixed total budget of $B = 256$ from two complementary perspectives. We first compare Head-Fisher with uniform allocation to examine the benefit of layer-wise budget allocation. Both use the same anchor and coefficient candidates, while uniform assigns the same budget to each layer and only optimizes reconstruction within each layer. Head-Fisher achieves substantially better task accuracy and reduces WikiText-2 PPL from 1047.41 to 24.54, showing that distributing the budget adaptively across layers is important. We then examine what signal should guide this allocation by comparing Head-Fisher with reconstruction-only (Recon.) and Rank@90-based allocation. Recon. allocates budgets using reconstruction error alone, while Rank@90 relies on spectral compressibility. Incorporating Head-Fisher raises the five-task mean accuracy from 31.95% under Recon. to 49.98% and achieves a lower PPL than both Recon. (2146.26) and Rank@90 (58.91). These results show that adaptive layer-wise allocation is necessary, and that task sensitivity provides a more effective allocation signal than reconstruction fidelity or spectral compressibility alone (Figure 8(b)–(c)).

Calibration robustness. We independently repeat the full offline pipeline on C4, BookCorpus, and WikiText-103, each with 128 sequences of 1024 tokens. Figure 8(a) plots task-wise accuracy differences relative to the C4-calibrated configuration at a 50% budget. Across PIQA, HellaSwag, and the two ARC tasks, the absolute change is at most 0.60 points; on BoolQ it is at most 1.97 points. These small deviations indicate that refitting anchors, coefficients, and layer budgets on different calibration corpora yields similar downstream performance.

5 Conclusion

CASH demonstrates how to preserve the logical head structure of an LLM without independently evaluating every attention map. By combining real-anchor convex reconstruction, Head-Fisher-guided allocation, and offline permutation, it turns cross-head redundancy into a static MPC workload while retaining all value and output branches. Without fine-tuning, CASH achieves the highest average accuracy in 11 of 12 matched-budget settings. At a 50% evaluated-head budget and sequence length 512, it delivers 1.47–1.69× secure inference speedups and 33.6–41.9% lower communication across four models under the evaluated LAN setting. These results establish task-aware reconstruction as a complementary direction to secure-operator optimization, showing how model structure can be adapted to reduce end-to-end private-inference cost.

AI USE STATEMENT

In this work, we used generative AI tools for assisting with translation from Chinese to English in the manuscript preparation process. We have not used generative AI tools for developing research methods, designing or providing feedback on research methodology or experiments, implementing methods, formulating theoretical claims, generating hypotheses, interpreting experimental results, conducting data analysis, or any other research-related tasks requiring disclosure. Additionally, we used generative AI tools for improving English-language readability, polishing manuscript text, and assisting with LaTeX formatting. We have reviewed all AI-assisted work and verified that the final manuscript accurately reflects the authors' research, methods, experiments, and conclusions. The authors are responsible for the final content of this work, including all text, claims, experimental results, and artifacts produced with the aid of generative AI.

REFERENCES

Saurabh Agarwal, Bilge Acun, Basil Hosmer, Mostafa Elhoushi, Yejin Lee, Shivaram Venkataraman, Dimitris Papailiopoulos, and Carole-Jean Wu. CHAI: Clustered head attention for efficient LLM inference. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 291–312, 2024. URL https://proceedings.mlr.press/v235/agarwal24a.html.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4895–4901, 2023. doi: 10.18653/v1/2023.emnlp-main.298. URL https://aclanthology.org/2023.emnlp-main.298/.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7432–7439, 2020. URL https://arxiv.org/abs/1911.11641.

Chi-Chih Chang, Wei-Cheng Lin, Chien-Yu Lin, Chong-Yan Chen, Yu-Fang Hu, Pei-Shuo Wang, Ning-Chi Huang, Luis Ceze, Mohamed Abdelfattah, and Kai-Chiang Wu. Palu: KV-cache compression with low-rank projection. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/7da6e0e00702c60607a6ae05c802ef85-Abstract-Conference.html.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2924–2936, 2019. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300/.

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. URL https://arxiv.org/abs/1803.05457.

Ye Dong, Wen-jie Lu, Yancheng Zheng, Haoqi Wu, Derun Zhao, Jin Tan, Zhicong Huang, Cheng Hong, Tao Wei, and Wenguang Chen. PUMA: Secure inference of LLaMA-7b in five minutes. *Security and Safety*, 4:2025014, 2025. doi: 10.1051/sands/2025014. URL https://arxiv.org/abs/2307.12533.

Kanav Gupta, Neha Jawalkar, Ananta Mukherjee, Nishanth Chandran, Divya Gupta, Ashish Panwar, and Rahul Sharma. SIGMA: Secure GPT inference with function secret sharing. *Proceedings on Privacy Enhancing Technologies*, 2024(4):61–79, 2024. doi: 10.56553/popets-2024-0107. URL https://petsymposium.org/popets/2024/popets-2024-0107.php.

Meng Hao, Hongwei Li, Hanxiao Chen, Pengzhi Xing, Guowen Xu, and Tianwei Zhang. Iron: Private inference on transformers. In *Advances in Neural Information Processing Systems*, volume 35, pp. 15718–15731, 2022. doi: 10.52202/068431-1143. URL https://proceedings.neurips.cc/paper/2022/hash/64e2449d74f84e5b1a5c96ba7b3d308e-Abstract.html.

Yen-Chang Hsu, Ting Hua, Sungen Chang, Qian Lou, Yilin Shen, and Hongxia Jin. Language model compression with weighted low-rank factorization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=uPv9Y3gmAI5.

Zhicong Huang, Wen-jie Lu, Cheng Hong, and Jiansheng Ding. Cheetah: Lean and fast secure two-party deep neural network inference. In *31st USENIX Security Symposium*, pp. 809–826, 2022. URL https://www.usenix.org/conference/usenixsecurity22/presentation/huang-zhicong.

Andes Y. L. Kei and Sherman S. M. Chow. SHAFT: Secure, handy, accurate and fast transformer inference. In *Network and Distributed System Security Symposium*, 2025. URL https://www.ndss-symposium.org/ndss-paper/shaft-secure-handy-accurate-and-fast-transformer-inference/.

Woosuk Kwon, Sehoon Kim, Michael W. Mahoney, Joseph Hassoun, Kurt Keutzer, and Amir Gholami. A fast post-training pruning framework for transformers. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/987bed997ab668f91c822a09bce3ea12-Abstract-Conference.html.

Dacheng Li, Rulin Shao, Hongyi Wang, Han Guo, Eric P. Xing, and Hao Zhang. MPCFormer: Fast, performant and private transformer inference with MPC. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=CWmvjOEhgH-.

Zhengyi Li, Kang Yang, Jin Tan, Wen-jie Lu, Haoqi Wu, Xiao Wang, Yu Yu, Derun Zhao, Yancheng Zheng, Minyi Guo, and Jingwen Leng. Nimbus: Secure and efficient two-party inference for transformers. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/264a9b3ce46abdf572dcfe0401141989-Abstract-Conference.html.

Bingli Liao and Danilo Vasconcellos Vargas. Beyond KV caching: Shared attention for efficient LLMs. *arXiv preprint arXiv:2407.12866*, 2024. URL https://arxiv.org/abs/2407.12866.

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. In *Network and Distributed System Security Symposium*, 2025. URL https://www.ndss-symposium.org/ndss-paper/bumblebee-secure-two-party-inference-framework-for-large-transformers/.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. LLM-Pruner: On the structural pruning of large language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/44956951349095f74492a5471128a7e0-Abstract-Conference.html.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Byj72udxe.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32, 2019.

Payman Mohassel and Peter Rindal. ABY3: A mixed protocol framework for machine learning. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pp. 35–52, 2018. doi: 10.1145/3243734.3243760. URL https://eprint.iacr.org/2018/403.

Qi Pang, Jinhao Zhu, Helen Möllering, Wenting Zheng, and Thomas Schneider. BOLT: Privacy-preserving, accurate and efficient inference for transformers. In *IEEE Symposium on Security and Privacy*, 2024. URL https://eprint.iacr.org/2023/1893.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL https://jmlr.org/papers/v21/20-074.html.

Deevashwer Rathee, Dacheng Li, Ion Stoica, Hao Zhang, and Raluca Popa. MPC-minimized secure LLM inference. *arXiv preprint arXiv:2408.03561*, 2024. URL https://arxiv.org/abs/2408.03561.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. URL https://arxiv.org/abs/2302.13971.

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, 2019. doi: 10.18653/v1/P19-1580. URL https://aclanthology.org/P19-1580/.

Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. SVD-LLM: Truncation-aware singular value decomposition for large language model compression. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/3104e1ab39875cf54fe1eb4473e7c5a1-Abstract-Conference.html.

Haoqi Wu, Wenjing Fang, Yancheng Zheng, Junming Ma, Jin Tan, and Lei Wang. Ditto: Quantization-aware secure inference of transformers upon MPC. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53346–53365, 2024. URL https://proceedings.mlr.press/v235/wu24d.html.

Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, Junxian Guo, Shang Yang, Haotian Tang, Yao Fu, and Song Han. DuoAttention: Efficient long-context LLM inference with retrieval and streaming heads. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/5c1ddd2e59df46fd2aa85c833b1b36ed-Abstract-Conference.html.

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. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Bo Zeng, Zhi Pang, Yuyang Zhang, Kai Zhao, Tian Wu, Geying Yang, Lina Wang, and Run Wang. PCFormer: Accelerating privacy-preserving transformer inference by partition and combination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 28076–28084, 2026. doi: 10.1609/aaai.v40i33.40033. URL https://ojs.aaai.org/index.php/AAAI/article/view/40033.

Wenxuan Zeng, Ye Dong, Jinjin Zhou, Jin Tan, Lei Wang, Tao Wei, Runsheng Wang, and Meng Li. MPCache: MPC-friendly KV cache eviction for efficient private LLM inference. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2501.06807v2. arXiv:2501.06807, version 2.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. OPT: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022. URL https://arxiv.org/abs/2205.01068.

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*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/83a14a36de4502bac5b580db36e81858-Paper-Conference.pdf.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching

movies and reading books. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 19–27, 2015. URL https://openaccess.thecvf.com/content_iccv_2015/html/Zhu_Aligning_Books_and_ICCV_2015_paper.html.

A Implementation and Reconstruction Properties

A.1 Calibration and Anchor Search

For each layer, we concatenate post-Softmax attention probabilities over calibration sequences and valid causal coordinates $\Omega_x = \{(q, k) : k \leq q\}$ to obtain A_ℓ. All head representations and gradient statistics use the same coordinate order and causal mask. We accumulate the Gram matrix in coordinate blocks:

$$G_\ell = A_\ell A_\ell^\top \tag{10}$$

Calibration uses evaluation mode, batch size one, and eager attention, with KV computed independently for each sequence block. Model parameters remain frozen; activation gradients are enabled only to compute Head Fisher. Gram accumulation and coefficient optimization use double precision.

For each layer, we factor $G = U\Lambda U^\top$ and construct the head-space embedding $Z = U\Lambda^{1/2}$. The first anchor is the head farthest from the centroid of all heads; subsequent anchors are selected iteratively as the heads farthest from the current anchor convex hull. Ties are broken by the smaller head index. This produces a nested anchor ordering whose prefixes define candidate anchor sets for different budgets. The greedy procedure constructs the candidate library without requiring a globally optimal search over all head subsets.

For a given anchor set S and non-anchor head h, reconstruction coefficients are obtained from the convex program

$$\min_{c \geq 0,\, \mathbf{1}^\top c = 1} G_{hh} - 2c^\top G_{S,h} + c^\top G_{S,S} c. \tag{11}$$

We solve this problem with SLSQP, using a tolerance of 10^{-12} and a maximum of 2000 iterations. Small negative coefficients due to numerical error are clipped, and each coefficient row is renormalized. Anchor-head coefficients are explicitly set to their corresponding one-hot vectors. Dividing the resulting squared reconstruction error by the number of calibration sequences gives the residual in Equation (4).

A.2 Head-Fisher Normalization and Budget Solver

The calibration loss is the sum of next-token cross-entropies within each sequence. Gradient squares are collected at the interface between Softmax output and PV computation and divided by the total number of valid causal coordinates M; reconstruction residuals are divided by the sequence count N. All candidates use the same normalization, without gradient clipping or additional layer-wise sensitivity scaling.

Head Fisher is a block-level sensitivity estimate at the attention-map interface, used to assess the effect of reconstruction errors on task loss. It requires neither a full Fisher matrix nor model-parameter updates. Residual-only allocation uses the same candidate library and replaces the cost with $\sum_h e_{\ell,h}(r)$.

The dynamic program initializes $F_0(0) = 0$ and sets all other unreachable states to $+\infty$. It obtains the allocation by backtracking from $F_L(B)$. When all budget counts are available, feasibility requires $L \leq B \leq LH$. The algorithm uses $O(LBH)$ time, $O(B)$ rolling-state memory, and $O(LB)$ backpointer memory. Its optimality applies to the constructed finite candidate set and additive distortion objective.

A.3 Probability Preservation and Hard Sharing

For any valid query row, the simplex constraints on reconstruction coefficients imply

$$\sum_k \widehat{P}_{h,qk} = \sum_j c_{hj} \sum_k P_{a_j,qk} = 1. \tag{12}$$

Convex reconstruction therefore preserves attention-map nonnegativity, row normalization, and common causal support. A fixed one-hot sharing matrix is feasible within the convex constraint set, so the optimal convex fit has no greater error than hard sharing for the same anchor set.

B SECURE COMPILATION

Let $\pi = (a_1, \ldots, a_r, \text{remaining heads})$ denote the offline head permutation. The model owner consistently reorders the Q/K/V parameter blocks, output-projection input blocks, and reconstruction coefficients, preserving the correspondence with the original model's attention computation. For $T \in \{Q, K, V, O\}$,

$$W_i^{T\prime} = W_{\pi(i)}^T, \qquad C'_{ij} = C_{\pi(i),j}. \tag{13}$$

The first r coefficient rows are the unit vectors corresponding to anchor heads. This offline permutation avoids input-dependent online head access while preserving the correspondence between value branches and the output projection. The current implementation targets standard multi-head attention; grouped-query attention additionally requires permutations that respect shared K/V groups.

CASH uses the standard semi-honest two-party MPC setting adopted by existing private Transformer inference systems. Model weights, inputs, activations, and reconstruction coefficients remain private online; model architecture, tensor shapes, the mask, and the budget schedule are public. CASH is implemented using secure-computation primitives supplied by the underlying MPC backend and introduces no new cryptographic protocol. The PUMA experiments inherit the security model and preprocessing assumptions of its three-party backend; the trust setting of the two-party experiments remains unchanged.

With m query–key positions per head, reducing H fully evaluated heads to r anchor heads removes $(H-r)md_h$ score-product multiplications and introduces $(H-r)rm$ secure reconstruction multiplications. All Hmd_h value-product multiplications, value projections, and output projections remain unchanged. Appendix D further specifies the scope of the secure-efficiency measurements.

C SUPPORTING STRUCTURAL EVIDENCE

Spectral statistic. Rank@q is the smallest r satisfying

$$\frac{\sum_{i \le r} \lambda_i(G_\ell)}{\operatorname{tr}(G_\ell)} \ge q.$$

It measures the concentration of attention-map energy in head space, rather than algebraic rank or the number of real anchors needed to achieve the same error. Table 4 supplements the cross-head analysis in Figure 2. Each corpus uses 128 independent sequences of 1,024 tokens.

Table 4: Cross-head energy statistics on LLaMA-1 7B.

Corpus	Mean R90	Median R90	Max R90	Mean R95
C4	4.03	2.0	16	9.75
WikiText-2	4.06	2.5	15	9.63
BookCorpus	4.28	3.0	14	9.53

The gap between reconstruction error and task impact. The allocation motivation in Section 3.2 comes from an additional diagnostic experiment. We compare allocation strategies in a fixed ARC-500 evaluation setting and compute the overall relative attention-map error:

$$\varepsilon = \sqrt{\frac{\sum_\ell \|A_\ell - \widehat{A}_\ell\|_F^2}{\sum_\ell \|A_\ell\|_F^2}}.$$

The Rank@90-matched configuration with $B = 258$ selects budgets using spectral statistics and uses the corresponding real anchors and reconstruction coefficients. The experiment shows that configurations with similar reconstruction error can have different task performance: allocation based solely on attention-map fidelity does not fully capture model behavior. We therefore introduce Head Fisher as a task-sensitivity weight.

Table 5: ARC-500 diagnostic results supporting budget allocation.

Configuration	B	Rel. error	ARC-500 (%)	Wiki PPL
Dense	1024	0.0000	38.6	6.78
CHAI hard sharing	376	0.4261	25.6	26.91
Convex, CHAI counts	376	0.2883	32.2	16.89
Convex, Rank@90-error matched	258	0.2894	23.2	50.51

D Experimental Protocol and Supporting Results

Baseline implementations. For each model, all baselines use the same pretrained checkpoint with frozen weights and no additional fine-tuning or knowledge distillation. Calibration-based configuration uses the same C4 samples, and the main comparison aligns the total budget of independently evaluated attention heads across methods. For SA, we retain direct cross-layer attention sharing and introduce budget-constrained selection of multiple contiguous sharing intervals based on attention similarity measured on C4. For HM, we retain MARILL's head-merging mechanism, grouping heads according to their similarity on the calibration data and merging each group into a wider head. This HM baseline keeps the original model weights frozen and omits MARILL's fine-tuning, distillation, and other complementary components. For CHAI, we adopt the dynamic variant based on the substantial quality degradation observed with the evaluated static configuration. At the 50% budget, with identical per-layer head counts determined from the same C4 calibration samples, static and dynamic CHAI on LLaMA-1 7B achieve five-task averages of 33.79% and 49.78%, respectively, with corresponding WikiText-2 perplexities of 2104.15 and 32.84 (Table 6). We hypothesize that the poor performance of the evaluated static configuration may stem in part from its dependence on the calibration data. Head groups and representatives determined offline may not adequately capture input-dependent head similarities on downstream evaluation inputs. Dynamic CHAI instead determines its head groups from each input's prefix, allowing the sharing assignments to adapt to the current input. The observed quality gap therefore motivates our choice of dynamic CHAI for the main comparison.

Table 6: Static and dynamic CHAI on LLaMA-1 7B at the 50% budget

Method	PIQA ↑	HellaSwag ↑	ARC-C ↑	ARC-E ↑	BoolQ ↑	PPL ↓
Static CHAI	50.27	25.99	26.79	27.69	38.23	2104.15
Dynamic CHAI	67.63	43.77	26.19	51.26	60.06	32.84

Metric definitions. Table 1 reports Avg. and PPL for the four models. Figure 6 uses task-level normalized results to show changes across capabilities: normalized accuracy for PIQA, HellaSwag, ARC-Challenge, and ARC-Easy, and accuracy for BoolQ.

Model-level Avg. is used to calculate quality retention. For method m on model k,

$$q_{m,k} = \frac{\mathrm{Avg}_{m,k}}{\mathrm{Avg}_{\mathrm{Dense},k}}.$$

Table 2 reports

$$100\bar{q}_m = 25 \sum_{k=1}^{4} q_{m,k}.$$

Thus, 92.0% denotes mean quality retention relative to Dense, rather than task accuracy. PPL is computed under the same evaluation protocol for the methods being compared.

Fixed-budget allocation ablation. The uniform, residual-only, and Head-Fisher strategies in Figure 8(b) use the same anchor candidates and reconstruction coefficients, differing only in budget allocation. The total budget is fixed at $B = 256$.

Uniform allocation distributes anchors evenly across layers; residual-only allocation uses $\sum_h e_{\ell,h}(r)$ as its objective; Head-Fisher allocation uses the weighted distortion in Equation (6).

Rank@90 matching uses the budget configuration derived from its spectral statistics and retrieves the corresponding historical anchors and coefficients. Because it determines budgets from spectral structure, it is separate from the control experiment that changes only the allocator.

Secure-inference measurement. We evaluate batch-one secure inference under the LAN setting in Section 4.1. Consistent with CipherPrune (Zhang et al., 2025), our main efficiency results report runtime and communication for the prefilling phase, excluding subsequent autoregressive decoding. Measurements cover model input processing, Transformer blocks, final normalization, the LM head, and output reveal.

Our main experiments use SHAFT (Kei & Chow, 2025) in the 2PC setting. To evaluate whether CASH transfers across secure-inference frameworks, we additionally integrate it with BumbleBee (Lu et al., 2025) under 2PC and PUMA (Dong et al., 2025) under 3PC. CASH does not modify the underlying cryptographic protocols. Anchor selection, coefficient fitting, budget allocation, and head permutation are performed offline, and the resulting configuration is compiled into a static computation graph before secure inference. As shown in Table 3, CASH reduces both latency and communication on PUMA, demonstrating that its benefits extend beyond the 2PC setting.

For method m, let T_m and C_m denote end-to-end prefill latency and communication, respectively. Speedup and communication-reduction factor relative to Dense are defined as

$$\mathrm{Speedup}_m = \frac{T_{\mathrm{Dense}}}{T_m}, \qquad \mathrm{CommFactor}_m = \frac{C_{\mathrm{Dense}}}{C_m}. \tag{14}$$

Communication reduction as a percentage is

$$\mathrm{CommReduction}_m = 100 \left(1 - \frac{C_m}{C_{\mathrm{Dense}}}\right) \%. \tag{15}$$

Diagnostic experiments. The calibration-robustness experiments vary the calibration corpus and sampling strategy and rerun the complete offline pipeline, including anchor search, coefficient fitting, and budget allocation. They assess the effect of calibration choices on the resulting configuration.

The single-layer ranking experiment in Figure 5 holds other layers fixed and compares task-loss changes across candidates for one layer. The full-model budget-allocation experiment uses a separate evaluation of the complete model.

E Related Work

Private Transformer inference. Existing private Transformer inference methods primarily reduce MPC costs through efficient secure operators (Hao et al., 2022; Pang et al., 2024; Li et al., 2024; Gupta et al., 2024; Lu et al., 2025; Kei & Chow, 2025), MPC-aware approximation and quantization (Li et al., 2023; Wu et al., 2024; Zeng et al., 2026), and computation reduction. CipherPrune (Zhang et al., 2025) combines token pruning with adaptive polynomial approximation, while MPCache (Zeng et al., 2025) exploits KV-cache redundancy through eviction and private selection. CASH targets a complementary direction: reducing the number of attention maps that must be generated independently. It preserves the original value and output parameterization, reconstructs non-anchor attention maps from real anchor heads selected offline, and compiles this process into a fixed MPC execution path.

Attention-head pruning, sharing, and merging. Attention-head redundancy has been widely studied in plaintext Transformers. Head pruning removes low-impact heads or coupled structures (Michel et al., 2019; Voita et al., 2019; Kwon et al., 2022; Ma et al., 2023). CHAI (Agarwal et al., 2024) shares attention through runtime head clustering, SA (Liao & Vargas, 2024) reuses attention across layers, and MARILL (Rathee et al., 2024) merges heads into fewer, wider ones during fine-tuning. GQA, DuoAttention, and Palu reduce KV-cache costs through head grouping, head-specific cache policies, and low-rank representations, respectively (Ainslie et al., 2023; Xiao et al., 2025; Chang et al., 2025). Calibration-based methods also preserve model behavior through weighted or data-aware low-rank weight reconstruction (Hsu et al., 2022; Wang et al., 2025). Head pruning and sharing exploit plaintext execution, where dynamic selection and routing can use inexpensive control flow. CASH adapts cross-head compression to MPC through fixed anchors and convex reconstruction, avoiding data-dependent online control while preserving the original value/output structure of the frozen pretrained model.