GeoPair: Geometry-Preserving Cross-Layer Factorization for Training-Free Transformer Compression

Baher Mohammad[1,2*], **Ammar Ali**[1,2*], **Stamatios Lefkimmiatis**[1,2]
[1]MWS AI [2]ITMO
baher@itmo.ru

Abstract

Transformer architectures exhibit cross-layer redundancies, yet post-training compression pipelines typically optimize layers in isolation or rely on heuristic grouping strategies that disregard layer-specific activation geometries. We introduce a principled, training-free framework that sequentially optimizes cross-layer weight pairings and shared-dictionary factorizations. Rather than forcing weights of adjacent layers to share a basis or heuristically merging activation statistics, our approach identifies structurally compatible projections and learns a shared representation that better preserves each layers distinct calibration geometry. Coupled with structured sparsity, this yields highly efficient weight decompositions without sacrificing functional fidelity. Across diverse architectures, scales, and modalities, our method achieves state-of-the-art results, consistently outperforming independent structured weight decompositions and alternative pairwise weight factorizations, which operate under heuristic grouping strategies. By combining geometry-preserving dictionary updates with graph-based pairing and sparse coefficient optimization, we provide a principled framework for transformer compression across modalities.

1 Introduction

The widespread adoption of transformer-based architectures has yielded unprecedented capabilities across language [36, 42, 22, 35, 1], vision [8, 11], and generative tasks [37]. However, their substantial memory footprint and computational overhead present a critical bottleneck for deployment in resource-constrained environments. Post-training model compression has emerged as a practical alternative to retraining, with matrix factorization methods offering a compelling trade-off between parameter efficiency and functional fidelity. While conventional approaches approximate each layer's weight matrix independently, they overlook a fundamental structural property: substantial cross-layer redundancies emerge naturally across deep transformer stacks.

Exploiting such redundancies through shared-dictionary learning, (where multiple projections are represented by a common dictionary with layer-specific coefficients), promises sublinear storage scaling without sacrificing expressiveness. Yet, principled pairwise cross-layer sharing remains largely unexplored in post-training compression. The primary obstacle lies in the data-aware nature of modern factorization pipelines: accurate low-rank approximation requires a whitening transform calibrated to each layer's activation distribution. Because these distributions vary significantly across depths, their associated whitening spaces are incompatible, complicating direct dictionary sharing. Recent approaches such as [38] heuristically aggregate layer-wise covariances into global whitening transforms and restrict sharing to adjacent layers. This introduces two limitations: (i) covariance

*Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

averaging distorts layer-specific activation geometry, degrading fidelity; (ii) fixed adjacency ignores non-local alignments, leaving gains unrealized.

In this work, we introduce a principled, training-free framework that optimizes cross-layer grouping and shared-dictionary factorization under layer-specific whitening transforms via alternating minimization. Rather than relying on heuristic covariance merging or greedy adjacency rules, our method identifies structurally compatible projections and learns a shared representation that rigorously preserves each layer's calibration geometry. We formulate the dictionary update as a generalized Sylvester equation, enabling closed-form solutions that respect distinct whitening spaces. Layer pairing is cast as a maximum-weight matching problem, solved via Edmonds' Blossom algorithm [12] using a shape-agnostic column-space alignment metric, with the solution being optimal with respect to the proposed distance metric. To further enhance compression efficiency and adapt the method for dictionary learning-based decompositions, we utilize Hard Thresholding Pursuit (HTP) [14] powered by a conjugate gradient linear solver, enabling structured coefficient sparsity without heuristic budget allocation or dynamic scheduling.

Contributions: Our framework establishes a reproducible, optimization-driven alternative to heuristic compression pipelines. The main contributions are:

- **Shared-dictionary learning under distinct whitening spaces**: We introduce a closed-form generalized Sylvester solver that eliminates heuristic covariance aggregation while preserving layer-specific activation geometry.
- **Layer pairing:** We formulate cross-layer grouping as a weighted maximum matching problem, replacing fixed adjacency heuristics with a data-driven strategy that minimizes structural discrepancy across the entire architecture.
- **Sparse coefficient optimization:** We extend the framework to sparse dictionary learning via HTP, enabling adaptive, layer-specific compression that outperforms dense low-rank baselines at high compression ratios.
- **Broad empirical validation:** Extensive experiments across diverse architectures, scales, and modalities demonstrate state-of-the-art results, consistently outperforming independent factorization, heuristic merging, and existing dictionary learning approaches.

By unifying Layer grouping, closed-form dictionary updates, and sparse coding, our work provides a theoretically grounded foundation for scalable, transformers compression.

2 Related Work

Data-Aware Matrix Factorization. Post-training compression via matrix factorization has emerged as a practical strategy for reducing transformer memory footprint without fine-tuning. Truncated singular value decomposition (SVD) yields the optimal rank-r approximation of a weight matrix under the Frobenius norm, and is mathematically equivalent to performing principal component analysis (PCA) on its column space [5]. Early compression pipelines applied this decomposition directly to pretrained weights, but assumed isotropic activation statistics, ignoring the input-dependent scaling inherent to transformer forward passes. This geometric mismatch causes significant accuracy degradation at high compression ratios. Subsequent data-aware approaches [9, 43, 39, 46] aligned the factorization objective with true activation reconstruction by operating in a calibration-induced whitened space. While these methods preserve downstream performance, they treat each layer in isolation, overlooking the substantial cross-layer redundancy inherent in deep architectures.

Dictionary Learning and Sparse Decomposition. To improve compression fidelity, recent work has shifted from dense low-rank approximations to dictionary learning formulations, which decouple a shared dictionary from layer-specific coefficient matrices. Classical sparse coding algorithms such as k-SVD [2] and Method of Optimal Directions [13] have been adapted to the transformer compression setting. Methods like CoSpaDI [19], ROCKET [3], and COMPOT [20] demonstrate that structured sparsity often yields superior trade-offs compared to dense baselines, particularly when coefficient matrices are heavily constrained. However, these approaches still optimize coefficients and dictionaries per layer or rely on heuristic update schedules, leaving explicit cross-layer parameter sharing unexplored.

Cross-Layer Grouping and Shared Basis Learning. Explicitly grouping structurally similar layers to learn a shared basis offers a promising path to sublinear storage scaling. The most notable advance in this direction, Basis Sharing [38] pairs adjacent layers and learns a joint basis, outperforming independent baselines. Yet, this approach suffers from three critical limitations: (i) fixed adjacency for grouping ignoring non-local alignments; (ii) global covariance aggregation which distorts layer-specific geometry; (iii) manual, model-specific constraints limit universal applicability. (e.g., restricting sharing between certain projection types), meaning the approach cannot be applied universally to maintain numerical stability. Complementary work like Matrix PCA [46] extracts shared bases via Eigen Value Decomposition (EVD) on stacked weights but remains constrained by distributional-drift-based grouping and independent refinement stages, necessitating manual budgeting and failing to optimize grouping under distinct whitening geometries.

Positioning of Our Approach. Our framework addresses these gaps through a principled, optimization-driven pipeline that sequentially solves layer grouping and shared-dictionary learning. Rather than heuristic covariance merging, we formulate cross-layer dictionary sharing under distinct whitening transforms as a generalized Sylvester equation, yielding closed-form dictionary updates that preserve individual layer geometries. We replace fixed adjacency rules with a global maximum-weight matching strategy [12], optimally pairing layers with respect to a shape-agnostic column-space alignment metric. Finally, we integrate Hard Thresholding Pursuit (HTP) [14] with conjugate gradient to enforce structured coefficient sparsity, enabling flexible compression without heuristic budget allocation or dynamic scheduling. This eliminates manual engineering while establishing a theoretically grounded pathway for scalable, multi-modal model compression.

3 Method

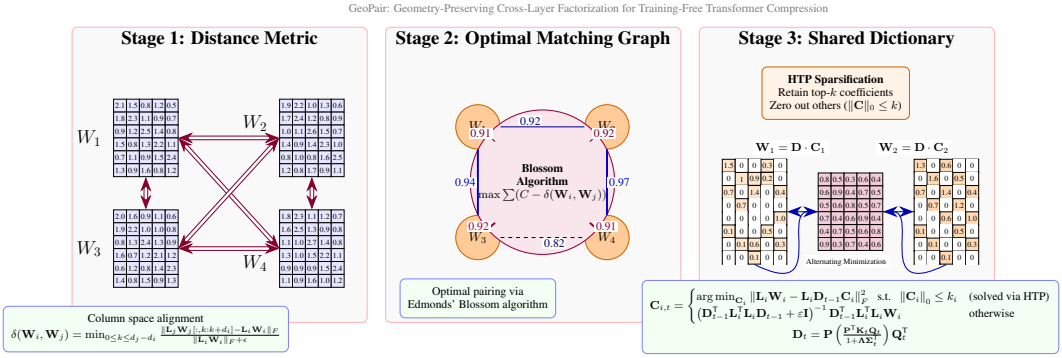

Figure 1: **GeoPair framework overview.** Stage 1 computes whitened-space structural distances between candidate weight matrices. Stage 2 solves for optimal pairings via maximum-weight graph matching with respect to the distance metric. Stage 3 learns shared dictionaries for each pair through calibration-aware alternating minimization with optional HTP-based coefficient sparsification.

3.1 Overview and Problem Setup

Similar to prior post-training compression work [39], we formulate compression as *activation reconstruction* over a small calibration set. We consider a pretrained transformer with linear projections parameterized by weight matrices $\mathbf{W} \in \mathbb{R}^{d \times d_{out}}$ and seek a structured approximation $\widehat{\mathbf{W}}$ that reduces storage and computation while preserving functional behavior, without the need for finetuning using back-propagation.

Let $\mathbf{X} \in \mathbb{R}^{N \times d}$ denote calibration activations and define the empirical Gram matrix $\mathbf{G} = \mathbf{X}^\top \mathbf{X}$. In practice, limited calibration data often yields a rank-deficient $\mathbf{G}$, rendering it singular. To guarantee a well-posed whitening transform, we enforce non-singularity by introducing a Tikhonov regularizer $\mathbf{G}_\eta = \mathbf{X}^\top \mathbf{X} + \eta \mathbf{I}$ ($\eta > 0$). This is mathematically equivalent to augmenting the reconstruction objective with a pure weight-space ℓ_2 penalty, which strictly ensures $\mathbf{G}_\eta \succ 0$ and admits a unique Cholesky factorization as a whitening transformation $\mathbf{G}_\eta = \mathbf{L}^\top \mathbf{L}$. The activation reconstruction objective is then equivalently written as

$$\widehat{\mathbf{W}} = \arg\min_{\widehat{\mathbf{W}}} \left\| \mathbf{X} \left(\mathbf{W} - \widehat{\mathbf{W}} \right) \right\|_F^2 + \eta \left\| \mathbf{W} - \widehat{\mathbf{W}} \right\|_F^2 = \arg\min_{\widehat{\mathbf{W}}} \left\| \mathbf{L} \left(\mathbf{W} - \widehat{\mathbf{W}} \right) \right\|_F^2 . \tag{1}$$

This shows that minimizing the activation reconstruction error is equivalent to minimizing the reconstruction in the whitened space induced by calibration statistics.

3.2 Cross-Layer Shared Dictionary Optimization

Unlike standard compression pipelines that optimize each projection in isolation, we aim to exploit structural redundancies that naturally arise across layers sharing the same input dimension. By coupling their factorizations, we can learn a single shared dictionary that efficiently spans both layers, yielding higher compression ratios at comparable reconstruction fidelity.

Building on this motivation, our optimization objective remains strictly tied to minimizing functional activation error. Following the equivalence established in Section 3.1, preserving the input–output behavior for two compatible projections, translates directly to minimizing their respective calibration-weighted reconstruction errors.

Cross-Layer Shared Dictionary Formulation. Consider two weight matrices $\mathbf{W}_1 \in \mathbb{R}^{d\times d_1}$ and $\mathbf{W}_2 \in \mathbb{R}^{d\times d_2}$, each with its own layer-specific Cholesky whitening transform $\mathbf{L}_1, \mathbf{L}_2 \in \mathbb{R}^{d\times d}$. We approximate them using a *shared dictionary* $\mathbf{D} \in \mathbb{R}^{d\times r}$ (where $r \ll d$) and layer-specific coefficient matrices $\mathbf{C}_1 \in \mathbb{R}^{r\times d_1}$, $\mathbf{C}_2 \in \mathbb{R}^{r\times d_2}$. We define:

$$s_i = \|\mathbf{L}_i\mathbf{W}_i\|_F, \qquad \bar{\mathbf{L}}_i = \frac{\mathbf{L}_i}{s_i}, \qquad \bar{\mathbf{G}}_i = \bar{\mathbf{L}}_i^\top\bar{\mathbf{L}}_i = \frac{\mathbf{L}_i^\top\mathbf{L}_i}{s_i^2}.$$

We assume nonzero weight matrices, so $s_i > 0$. Our objective is to minimize the sum of the two squared relative Frobenius reconstruction errors, which can be written using the normalized whitening transforms as

$$\min_{\mathbf{D},\mathbf{C}_1,\mathbf{C}_2} \quad \sum_{i=1}^{2} \left\|\bar{\mathbf{L}}_i(\mathbf{W}_i - \mathbf{D}\mathbf{C}_i)\right\|_F^2. \tag{2}$$

The shared dictionary $\mathbf{D}$ captures common directional components activated across both layers, while $\mathbf{C}_i$ projects these components onto each layer's output space. After compression, the original parameter space is recovered via $\widehat{\mathbf{W}}_i = \mathbf{D}\mathbf{C}_i$.

Alternating Minimization. The objective in Eq. (2) is bi-convex in $(\mathbf{D}, \mathbf{C})$. We optimize it via alternating minimization, which decouples the joint problem into two sequential subproblems. Because the optimization alternates between the dictionary and the coefficients, we require a distinct closed-form update rule for each block. Hereafter, we first derive the update rule for the coefficients $\mathbf{C}_1, \mathbf{C}_2$ given a fixed $\mathbf{D}_{t-1}$, and then present the update rule for $\mathbf{D}_t$ given the newly computed coefficients. These two steps are applied cyclically until convergence.

Coefficient update ($\mathbf{C}_1, \mathbf{C}_2$ given $\mathbf{D}_{t-1}$). With $\mathbf{D}_{t-1}$ fixed, we compute the coefficients using ridge-regularized least squares:

$$\begin{aligned}\mathbf{C}_{i,t} &= \arg\min_{\mathbf{C}_i} \left\{ \left\|\bar{\mathbf{L}}_i(\mathbf{W}_i - \mathbf{D}_{t-1}\mathbf{C}_i)\right\|_F^2 + \varepsilon \left\|\mathbf{C}_i\right\|_F^2 \right\} \\ &= \left(\mathbf{D}_{t-1}^\top\bar{\mathbf{G}}_i\mathbf{D}_{t-1} + \varepsilon\mathbf{I}\right)^{-1}\mathbf{D}_{t-1}^\top\bar{\mathbf{G}}_i\mathbf{W}_i, \qquad i \in \{1,2\}. \end{aligned} \tag{3}$$

Here $\varepsilon = 10^{-6}$ regularizes the coefficient subproblem and ensures a unique minimizer. In practice, we solve the corresponding linear system without explicitly forming the inverse.

Dictionary update ($\mathbf{D}_t$ given $\mathbf{C}_{1,t}, \mathbf{C}_{2,t}$). With the coefficients fixed, we solve the regularized dictionary subproblem

$$J_{\mathrm{rel}}(\mathbf{D}) = \sum_{i=1}^{2} \left[\left\|\bar{\mathbf{L}}_i(\mathbf{W}_i - \mathbf{D}\mathbf{C}_{i,t})\right\|_F^2 + \epsilon \left\|\bar{\mathbf{L}}_i\mathbf{D}\right\|_F^2 \right]. \tag{4}$$

Setting the derivative of J_{rel} to zero gives the generalized Sylvester normal equation

$$\bar{\mathbf{G}}_1\mathbf{D}_t\tilde{\mathbf{B}}_{1,t} + \bar{\mathbf{G}}_2\mathbf{D}_t\tilde{\mathbf{B}}_{2,t} = \mathbf{K}_t, \tag{5}$$

where

$$\mathbf{K}_t = \sum_{i=1}^{2} \bar{\mathbf{G}}_i\mathbf{W}_i\mathbf{C}_{i,t}^\top. \tag{6}$$

Defining $\Phi(\cdot,\cdot)$ as the *generalized eigenvalue decomposition* operator, we compute the activation Gram pair decomposition *once* to obtain constant transformation matrices $(\mathbf{P}, \mathbf{\Lambda}) = \Phi(\bar{\mathbf{G}}_2, \bar{\mathbf{G}}_1)$. At each iteration t, we stabilize the coefficient Gram matrices via $\tilde{\mathbf{B}}_{i,t} = \mathbf{C}_{i,t}\mathbf{C}_{i,t}^\top + \epsilon\mathbf{I}$ and decompose the regularized pair to obtain $(\mathbf{Q}_t, \mathbf{\Sigma}_t) = \Phi(\tilde{\mathbf{B}}_{2,t}, \tilde{\mathbf{B}}_{1,t})$. Using these transformations we can decouple the Sylvester equation into independent scalar equations, yielding the closed-form solution:

$$\mathbf{D}_t = \mathbf{P}\left(\frac{\mathbf{P}^\mathsf{T}\mathbf{K}_t\mathbf{Q}_t}{\mathbf{1}_d\mathbf{1}_r^\mathsf{T} + \boldsymbol{\lambda}\boldsymbol{\sigma}_t^\mathsf{T}}\right)\mathbf{Q}_t^\mathsf{T}, \tag{7}$$

where division is elementwise, $\boldsymbol{\lambda} = \operatorname{diag}(\mathbf{\Lambda})$, $\boldsymbol{\sigma}_t = \operatorname{diag}(\mathbf{\Sigma}_t)$, and $\mathbf{1}_d \in \mathbb{R}^d$ and $\mathbf{1}_r \in \mathbb{R}^r$ are vectors of ones. For more details we refer to Appendix A. This simultaneous diagonalization approach provides a deterministic update for $\mathbf{D}_t$ without iterative optimization or step-size tuning. The dense coefficient update exactly minimizes its ridge-regularized quadratic subproblem.

3.3 Sparse Matrix Coefficients via Hard Thresholding Pursuit

Recent state-of-the-art post-training compression methods increasingly rely on dictionary learning formulations that enforce structured sparsity in the factorized representations. Motivated by these advances, we integrate a sparsity-constrained coefficient update directly into our alternating minimization pipeline as a core mechanism for maximizing compression fidelity under strict parameter budgets. Specifically, we replace the dense least-squares coefficient update step with an ℓ_0-constrained formulation:

$$\begin{aligned} \min_{\mathbf{C}_{i,t}} \quad & \left\|\bar{\mathbf{L}}_i(\mathbf{W}_i - \mathbf{D}_{t-1}\mathbf{C}_{i,t})\right\|_F^2 + \varepsilon\left\|\mathbf{C}_{i,t}\right\|_F^2 \\ \text{s.t.} \quad & \left\|\mathbf{C}_{i,t}[:,j]\right\|_0 \le k, \qquad j = 1,\dots,d_i. \end{aligned} \tag{8}$$

Here k is the fixed per-column sparsity budget, and $\|\cdot\|_0$ counts nonzero entries. We approximately solve this subproblem using Hard Thresholding Pursuit (HTP). We define $\mathrm{CR} = 1 - S_{\mathrm{comp}}/S_{\mathrm{raw}}$, where S_{raw} and S_{comp} denote the nominal uncompressed and compressed representation sizes, respectively. The ratio $\mathrm{KS} = r/k$ is the inverse nonzero fraction per coefficient column (see Appendix B).

For a fixed dictionary $\mathbf{D}_{t-1}$, we define

$$\mathbf{H}_{i,t} = \mathbf{D}_{t-1}^\mathsf{T}\bar{\mathbf{G}}_i\mathbf{D}_{t-1} + \varepsilon\mathbf{I}, \qquad \mathbf{R}_{i,t} = \mathbf{D}_{t-1}^\mathsf{T}\bar{\mathbf{G}}_i\mathbf{W}_i. \tag{9}$$

We warm-start HTP with the coefficients from the previous outer iteration. Each inner iteration comprises a gradient step, support selection, and restricted least-squares refinement.

Gradient Update: At outer iteration t, initialize the HTP coefficients as $\mathbf{C}_i^{(0)} = \mathbf{C}_{i,t-1}$. For inner iteration h, compute

$$\mathbf{C}_i^{\mathrm{tmp},(h)} = \mathbf{C}_i^{(h)} + \mathbf{R}_{i,t} - \mathbf{H}_{i,t}\mathbf{C}_i^{(h)} \tag{10}$$

Support Selection: In each column of $\mathbf{C}_i^{\mathrm{tmp},(h)}$, select the k entries with the largest absolute values. Define the binary mask $\mathbf{M}_i^{(h+1)} \in \{0,1\}^{r\times d_i}$ to be one at the selected locations and zero elsewhere.

Restricted Projection: With the support fixed, compute $\mathbf{C}_i^{(h+1)}$ by approximately solving

$$\begin{aligned} \min_{\mathbf{C}} \quad & \left\|\bar{\mathbf{L}}_i(\mathbf{W}_i - \mathbf{D}_{t-1}\mathbf{C})\right\|_F^2 + \varepsilon\left\|\mathbf{C}\right\|_F^2 \\ \text{s.t.} \quad & \mathbf{C} = \mathbf{C} \odot \mathbf{M}_i^{(h+1)}. \end{aligned} \tag{11}$$

The restricted normal equations are

$$\mathbf{M}_i^{(h+1)} \odot (\mathbf{H}_{i,t}\mathbf{C} - \mathbf{R}_{i,t}) = \mathbf{0}, \tag{12}$$

with coefficients outside the selected support fixed to zero. We approximately solve this system using batched Conjugate Gradient (CG), initialized with $\mathbf{C}_i^{(h)} \odot \mathbf{M}_i^{(h+1)}$.

After T_{HTP} inner iterations, we set $\mathbf{C}_{i,t} = \mathbf{C}_i^{(T_{\mathrm{HTP}})}$ and proceed to the dictionary update. We use two HTP iterations and at most ten CG iterations per restricted solve. When $k = r$, the sparsity constraint is inactive, and we use the dense coefficient update in Eq. (3).

The finite-step HTP+CG procedure constitutes an inexact coefficient update. Convergence results based on BSUM and Kurdyka–Łojasiewicz theory require additional conditions, including suitable decrease and control of the inexact updates. We do not establish these conditions for the implemented finite-step solver. Its practical behavior is assessed through the iteration-budget ablation. Further discussion is provided in Appendix A.2.

3.4 Cross-Layer Grouping via Graph Matching

While prior compression pipelines default to pairing adjacent layers, structural similarities in pretrained weight matrices are not strictly localized. To maximize the efficacy of cross-layer dictionary sharing, we formulate layer grouping as a global optimization problem that pairs weights with minimal structural discrepancy.

Whitened-Space Frobenius Submatrix Distance Given two projection matrices $\mathbf{W}_i \in \mathbb{R}^{d\times d_i}$ and $\mathbf{W}_j \in \mathbb{R}^{d\times d_j}$ sharing the same input dimension d but potentially differing in output dimension, we define a scale-invariant surrogate metric for the joint approximation error. Without loss of generality, assume $d_i < d_j$. The normalized Frobenius submatrix distance is computed in the calibration-induced whitened space as:

$$\delta(\mathbf{W}_i, \mathbf{W}_j) = \min_{0\le k\le d_j - d_i} \frac{\|\mathbf{L}_j \mathbf{W}_j[:, k : k + d_i] - \mathbf{L}_i \mathbf{W}_i\|_F}{\|\mathbf{L}_i \mathbf{W}_i\|_F + \epsilon}, \tag{13}$$

For the case of $d_i = d_j$, please refer to appendix B.

where $\mathbf{L}_i, \mathbf{L}_j$ are the layer-specific Cholesky whitening transforms derived from calibration activations, $\mathbf{W}_j[:, k : k + d_i]$ extracts a contiguous column window of width d_i, and $\epsilon > 0$ ensures numerical stability. This metric captures the minimal alignment cost between the two weight spaces while explicitly accounting for distinct activation geometries and is computed in $\mathcal{O}(d \cdot d_j)$ time using optimized 1D cross-correlation, avoiding explicit window allocations.

Global Maximum-Weight Matching. Crucially, our grouping strategy is not restricted to identical submodule types; attention and feed-forward weights can be paired whenever they share a common input dimension d. Let $\mathcal{V} = \{1, \dots, N\}$ index all candidate weight matrices across layers and projection types. We construct an undirected complete graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with edge weights defined as $w_{ij} = \mathcal{C} - \delta(\mathbf{W}_i, \mathbf{W}_j)$, where $\mathcal{C} > \max_{i,j} \delta(\mathbf{W}_i, \mathbf{W}_j)$ converts distance minimization into weight maximization. The pairing $\mathcal{P}^*$ is obtained by solving:

$$\mathcal{P}^* = \arg\max_{\mathcal{M}\subseteq\mathcal{E}} \sum_{\{i,j\}\in\mathcal{M}} w_{ij} \quad \text{s.t.} \quad \mathcal{M} \text{ is a valid matching}, \tag{14}$$

which simultaneously enforces maximum cardinality and minimal total structural distance. We solve this problem using the Edmonds' Blossom algorithm [12]. The resulting disjoint pairs are subsequently passed to the calibration-aware alternating minimization of Section 3.2, ensuring that dictionary sharing is restricted to structurally aligned projections rather than arbitrary adjacent layers. This data-driven grouping strategy consistently yields lower activation-weighted reconstruction error and improved downstream perplexity compared to fixed adjacent pairing.

While the Blossom algorithm optimizes this tractable surrogate metric rather than the final joint reconstruction loss directly, we empirically verify that the whitened submatrix alignment cost is predictive of the optimized pairwise reconstruction error (see Appendix E.6).

3.5 Algorithmic Summary

For clarity, we consolidate the complete alternating minimization procedure into its explicit initialization and iterative update rules:

- **Initialization ($t = 0$):** Let $[\mathbf{W}_1\ \mathbf{W}_2] = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\mathsf{T}$ be a singular value decomposition, with singular values ordered nonincreasingly. Initialize
$$\mathbf{D}_0 = \mathbf{U}_{[:,1:r]}.$$

(15)

For the sparse variant, we initialize $\mathbf{C}_{i,0}$ using Eq. (3) with $\mathbf{D}_{t-1} = \mathbf{D}_0$.

- **Coefficient Update ($t \geq 1$):** Using $\mathbf{H}_{i,t}$ and $\mathbf{R}_{i,t}$ from Eq. (9), compute

$$\mathbf{C}_{i,t} = \begin{cases} \mathrm{HTP}\left(\mathbf{H}_{i,t}, \mathbf{R}_{i,t}, \mathbf{C}_{i,t-1}; k\right), & k < r, \\ \mathbf{H}_{i,t}^{-1}\mathbf{R}_{i,t}, & k = r. \end{cases} \tag{16}$$

  Here HTP denotes the finite-budget HTP+CG procedure described in Section 3.3, initialized with $\mathbf{C}_{i,t-1}$.

- **Dictionary Update ($t \geq 1$):**

$$\begin{aligned} \mathbf{D}_t &= \arg\min_{\mathbf{D}} \sum_{i=1}^{2} \left[\left\| \bar{\mathbf{L}}_i (\mathbf{W}_i - \mathbf{D}\mathbf{C}_{i,t}) \right\|_F^2 + \epsilon \left\| \bar{\mathbf{L}}_i \mathbf{D} \right\|_F^2 \right] \\ &= \mathbf{P} \left(\frac{\mathbf{P}^\mathsf{T} \mathbf{K}_t \mathbf{Q}_t}{\mathbf{1}_d \mathbf{1}_r^\mathsf{T} + \boldsymbol{\lambda} \boldsymbol{\sigma}_t^\mathsf{T}} \right) \mathbf{Q}_t^\mathsf{T}. \end{aligned} \tag{17}$$

We run at most 50 outer iterations and stop after three consecutive iterations that fail to improve the best recorded reconstruction loss by more than 10^{-5}. We return the factors attaining the lowest observed reconstruction loss.

For practical storage details and throughput analysis, please refer to the appendix C.

4 Experiments

This section systematically evaluates our approach, hereafter referred to as GeoPair, across design components and assess performance across different settings using 7 well established benchmarks (we refer to Appendix B for details). We begin with a component-wise ablation on two representative language models, comparing each configuration against the Basis Sharing baseline to isolate the impact of our proposed modules. Following this analysis, we benchmark our method against recent dictionary learning approaches to establish its effectiveness within this paradigm. We then evaluate the framework against a broad set of pruning and compression techniques, demonstrating that our training-free pipeline achieves competitive accuracy without the post-compression fine-tuning typically required by existing methods. To further assess scalability and architectural robustness, we extend the comparison against Basis Sharing across varying compression ratios and diverse model families. Finally, we apply the framework to a recent video generation model, providing qualitative evidence that high-fidelity generation is preserved without any post-compression adaptation or recovery steps.

Pairwise Weight Optimization and Coefficient Sparsification Table 1 evaluates the Sylvester update, grouping strategy, and sparse coefficient representation at $\mathrm{CR} = 0.2$. On Llama-3.2 1B, replacing global whitening with the Sylvester update increases average accuracy from 33.11% to 41.26%, and incorporating the proposed grouping increases it further to 43.03%. The corresponding results on Llama-3 8B are 53.47%, 56.40%, and 57.46%.

Moving from dense factorization to sparse dictionary factorization produces the largest accuracy increase. This comparison changes the representation family; it does not isolate the benefit of HTP itself. Within the sparse representation, the proposed grouping improves average accuracy from 53.06% to 54.09% on Llama-3.2 1B and from 67.31% to 68.17% on Llama-3 8B. Thus, the geometry-preserving dictionary update and grouping provide improvements within the evaluated representation families. Additional comparisons with Basis Sharing and CoSpaDi are provided in Appendix D.2.

Shared Dictionary Learns Better Figure 2 evaluates GeoPair against a broad set of structured weight factorization methods, encompassing both dense low-rank projections and sparse dictionary learning approaches. GeoPair consistently achieves the highest accuracy, outperforming established

Table 1: Component ablation at CR $= 0.2$. GW denotes global whitening; Sylv denotes the geometry-preserving Sylvester update; OG denotes our graph-matching strategy; and SP denotes sparse coefficients obtained using HTP. Dense and sparse configurations use our Sylvester update with either adjacent or matched pairing.

Configuration	Components				CR	Benchmarks (Accuracy)								Perplexity		Avg Acc
	GW	Sylv	OG	SP		PIQA	HellaSwag	Lambada_OA	ARC-e	ARC-c	SciQ	Race	MMLU	Wiki	Lambada	
Llama3.2 1B (baseline)	✗	✗	✗	✗	–	74.53	63.66	62.95	60.47	36.20	88.30	37.79	37.00	11.60	5.73	57.61
Basis Sharing	✓	✗	✗	✗	0.2	56.75	31.69	15.08	32.49	21.59	58.40	25.93	22.95	928.07	239.30	33.11
Dense Coeff. – adjacent	✗	✓	✗	✗	0.2	63.76	40.21	32.91	41.33	24.74	73.80	29.86	23.44	109.59	57.95	41.26
Dense Coeff. – matched	✗	✓	✓	✗	0.2	64.91	42.64	37.42	44.02	25.94	76.30	29.95	23.05	56.60	32.49	43.03
Sparse Coeff. – adjacent	✗	✓	✗	✓	0.2	71.44	57.35	56.55	54.46	32.51	87.50	34.93	29.70	16.77	7.88	53.06
Sparse Coeff. – matched	✗	✓	✓	✓	0.2	73.50	57.67	58.88	57.74	31.83	88.90	34.26	29.96	**15.92**	**6.74**	**54.09**
Llama3 8B (baseline)	✗	✗	✗	✗	–	80.69	79.13	75.57	77.69	53.5	93.9	40.29	62.15	7.26	3.09	70.36
Basis Sharing	✓	✗	✗	✗	0.2	72.52	58.71	50.2	57.2	34.04	84.9	37.13	33.04	41.26	14.84	53.47
Dense Coeff. – adjacent	✗	✓	✗	✗	0.2	74.21	61.14	57.4	59.76	36.18	86.9	37.13	38.47	26.82	9.69	56.4
Dense Coeff. – matched	✗	✓	✓	✗	0.2	73.5	59.8	60.7	65.11	37.29	90.8	37.42	35.09	24.56	6.82	57.46
Sparse Coeff. – adjacent	✗	✓	✗	✓	0.2	78.35	76.18	70.62	73.4	49.91	93.2	39.43	57.37	9.75	4.15	67.31
Sparse Coeff. – matched	✗	✓	✓	✓	0.2	78.73	75.38	74.25	76.14	49.74	93.9	40.77	56.45	**9.43**	**3.25**	**68.17**

baselines in both categories. This advantage stems from our shared dictionary formulation and grouping strategy, which more effectively preserves weight structure than conventional low-rank approximations or other dictionary learning strategies. These results establish GeoPair as the leading training-free weight factorization method for the evaluated compression regime.

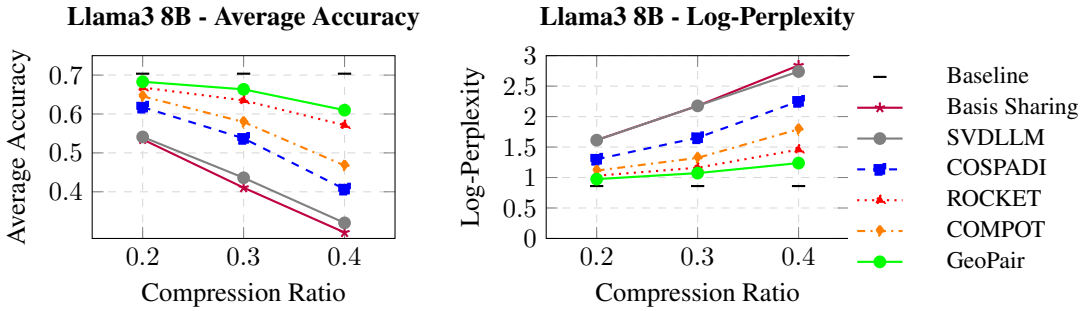

Figure 2: Comparison of different compression methods on Llama3 8B across varying compression ratios. Left: Average accuracy across benchmarks. Right: Log-Perplexity on WikiText.

Comparison with other compression methods Table 2 evaluates GeoPair against a broad range of compression and pruning strategies, including methods that operate outside the matrix factorization paradigm. Under a consistent evaluation protocol, GeoPair achieves the highest average accuracy across all benchmarks while operating entirely without post-compression fine-tuning. In contrast, competing approaches rely on extensive healing or recovery training, which introduces significant computational overhead and requires large datasets. These results establish GeoPair as the state-of-the-art training-free compression method.

Table 2: Comparison against pruning methods on Llama2 7B across standard zero-shot benchmarks at 20% compression. Training-free indicates whether a finetuning is done after compression.

Method	Training-free	Accuracy↑							Avg.
		BoolQ	PIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	
Baseline	–	76.50	79.80	76.10	70.10	72.80	47.60	57.20	68.59
LLM-Pruner	✗	66.79	77.58	68.48	64.96	64.06	37.88	39.00	59.82
LoRAPrune	✗	65.82	**79.31**	70.00	62.76	65.87	37.69	39.14	60.05
WANDA	✓	65.75	74.70	64.52	59.35	60.65	36.26	39.40	57.23
ShortGPT	✗	68.26	72.28	61.70	63.77	60.22	39.00	41.60	58.12
LoRAShear	✗	72.78	76.36	69.49	**67.63**	69.02	39.47	40.78	62.22
GeoPair	✓	**74.06**	77.2	**71.8**	67.24	**72.1**	**41.3**	**41.4**	**63.58**

Generalization Across Model Architectures To assess cross-architecture robustness, we evaluate our compression pipeline across diverse model families and scales, ranging from 1B to 32B parameters. This evaluation verifies that our framework maintains effectiveness irrespective of model capacity, architectural design, or training paradigm. Table 3 reports average zero-shot accuracy and Lambada OpenAI perplexity under increasing compression ratios. For direct comparison, we include Basis Sharing results at compression ratios 0.2, 0.3, and 0.4, enabling a consistent assessment

of performance trends across compression intensities. Additional results for Qwen3 on an updated benchmark suite are provided in Appendix D.3 to confirm consistency under alternative evaluation protocols. We observe minor improvements on small compression ratios, this aligns with recent findings that small rank truncation acts as denoising, preserving salient features [30].

Table 3: Generalization across model families and parameter scales. For each architecture, we report average zero-shot accuracy and Lambada OpenAI perplexity under increasing compression ratios. At CR=0.2, 0.3, and 0.4, we include direct comparisons against Basis Sharing.

Model	Metric	CR=0	CR=0.2		CR=0.3		CR=0.4	
			Ours	Basis Sharing	Ours	Basis Sharing	Ours	Basis Sharing
Qwen 3 8B	Avg. Accuracy	70.46	**67.7**	61.25	**64.3**	55.17	**58.88**	46.41
	Perplexity	4.60	**4.87**	7.31	**6.56**	12.94	**13.40**	45.21
Gemma 3 12B	Avg. Accuracy	72.28	**71.64**	58.96	**67.72**	50.2	**60.49**	40.22
	Perplexity	4.16	**3.62**	35.65	**4.39**	164.3	**10.98**	765.7
Phi-4 14B	Avg. Accuracy	72.09	**74.07**	70.33	**70.25**	65.22	**68.59**	57.52
	Perplexity	3.49	**3.37**	3.44	**3.80**	4.33	**4.35**	8.3
Qwen 3 32B	Avg. Accuracy	74.55	**73.27**	69.64	**72.24**	66.04	**70.31**	59.5
	Perplexity	3.74	**3.26**	3.51	**3.22**	4.09	**3.38**	6.6

4.1 Generalization across tasks

To evaluate cross-task generalization, we apply GeoPair to a video generation model. Specifically, we compress Wan2.2 5B [37] at 20% and 40% compression ratios and demonstrate that the compressed models retain high-fidelity video generation capabilities without any post-compression fine-tuning. To quantitatively assess the preservation of semantic alignment under compression, we evaluate generated videos using X-CLIP [17] with 16-frame sampling over 50 prompts drawn from the `Rapidata/awesome-text2video-prompts` dataset. The compressed models exhibit near-baseline performance: at 20% compression, the average CLIP score decreases by only 1×10^{-4} (0.2164 vs. 0.2165 baseline), while even at 40% compression the relative degradation remains marginal (0.2111, a 2.5% drop). These results confirm that our training-free compression framework preserves cross-modal alignment and generalization capacity without task-specific adaptation or recovery procedures. Compression results on audio generation models are provided in Appendix D.4.

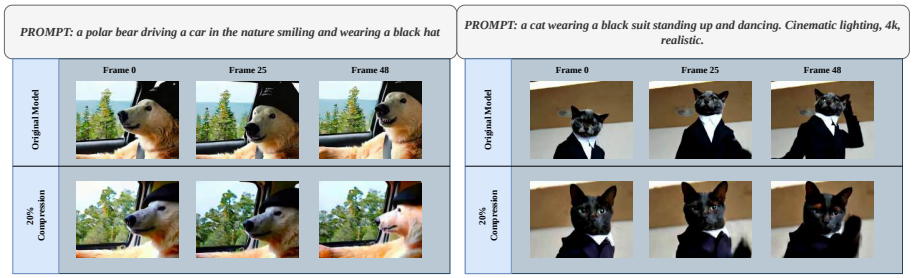

Figure 3: Video frames generated using Wan2.2 5B model and a compressed version using GeoPair.

5 Ablation Studies

This section evaluates the core components of the proposed compression framework to validate key design choices. We begin by analyzing the layer grouping strategy, comparing alternative distance metrics to determine the most effective criterion for pairing weight matrices. Next, we examine the framework's sensitivity to the KS ratio (defined as the number of atoms in the dictionary over the non-zero elements in each column of the coefficient matrices $\mathbf{C}_i$). In Appendix E we benchmark alternative sparse algorithms for enforcing target sparsity levels and perform ablations related to GeoPair's convergence and sensitivity on calibration data.

Grouping algorithm The proposed grouping strategy is optimal with respect to a chosen distance metric for each pair of weights, In Table 4, we ablate two different metrics as well as the greedy approach used in Basis Sharing [38].

Table 4: Ablation study of layer grouping strategies at CR=0.4 and KS=2.5 on Llama3.2 1B. Frobenius norm-based grouping achieves the best trade-off.

Grouping Strategy	CR	WikiText-2 (Perplexity↓)	Avg. Accuracy
Baseline (Llama3.2 1B)	–	11.60	57.6
General Frobenius Norm	0.4	**34.26**	**45.61**
Cosine Similarity	0.4	43.75	45.0
Consecutive Layers (Greedy)	0.4	37.97	43.35

Ablation on the KS ratio From Table 5 we observe that KS equal to 2.5 leads to the best results. Based on this observation, we set $k/s = 2.5$ for all the reported experiments that include sparsification of matrix coefficients.

Table 5: Ablation study of the KS ratio at CR=0.4 on Llama3.2 1B. The KS ratio balances the sparsity level of the coefficients and the rank of the dictionary. Lower perplexity and higher accuracy indicate better preservation of model capability. Intermediate KS values (2.5–3.5) yield the most favorable trade-offs, with KS=2.5 selected as the default configuration.

Metric	Baseline	KS Ratio (CR=0.4)				
		2.0	2.5	3.0	3.5	4.0
WikiText-2 (Perplexity↓)	11.60	48.51	34.26	33.96	**33.53**	35.43
Lambada (Perplexity↓)	5.73	27.60	**21.69**	21.80	22.71	24.55
Avg. Accuracy (↑)	57.6	43.5	**45.6**	45.1	45.0	44.3

6 Conclusion & Limitations

GeoPair introduces a training-free framework for pairwise dictionary sharing under layer-specific calibration geometries. The framework combines closed-form regularized Sylvester dictionary updates, maximum-weight matching under the proposed surrogate cost, and optional HTP-based sparse coefficient optimization. Experiments demonstrate favorable accuracy/compression trade-offs across the evaluated language models and show applicability to video and audio generation, without post-compression fine-tuning.

A primary limitation of the current framework is its restriction to pairwise layer grouping. Extending the optimization to simultaneously share dictionaries across larger groups of layers ($m > 2$) is non-trivial, as the Sylvester solver relies on the simultaneous diagonalization of two matrix pencils, a property that does not analytically generalize to m-term systems. Multi-group extensions would necessitate iterative numerical approximations or higher-order tensor factorizations, introducing additional computational overhead and potential numerical instability. Developing scalable, theoretically grounded strategies for m-way layer sharing remains a key challenge and constitutes a primary direction for future work.

Acknowledgments This research was supported as part of the ITMO University Research Projects in AI Initiative.

References

[1] ABDIN, M., ANEJA, J., BEHL, H., BUBECK, S., ELDAN, R., GUNASEKAR, S., HARRISON, M., HEWETT, R. J., JAVAHERIPI, M., KAUFFMANN, P., LEE, J. R., LEE, Y. T., LI, Y., LIU, W., MENDES, C. C. T., NGUYEN, A., PRICE, E., DE ROSA, G., SAARIKIVI, O., SALIM, A., SHAH, S., WANG, X., WARD, R., WU, Y., YU, D., ZHANG, C., AND ZHANG, Y. Phi-4 technical report. *arXiv preprint arXiv: 2412.08905* (2024).

[2] AHARON, M., ELAD, M., AND BRUCKSTEIN, A. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing 54*, 11 (2006), 4311–4322.

[3] ALI, A., MOHAMMAD, B., MAKHOV, D., SHOPKHOEV, D., ZHUSSIP, M., AND LEFKIMMIATIS, S. Rocket: Rapid optimization via calibration-guided knapsack enhanced truncation for efficient model compression. *arXiv preprint arXiv: 2602.11008* (2026).

[4] BARTELS, R. H., AND STEWART, G. W. Solution of the matrix equation AX + XB = C. *Communications of the ACM 15*, 9 (1972), 820–826.

[5] BISHOP, C. *Pattern recognition and machine learning*, vol. 4. Springer, 2006.

[6] BISK, Y., ZELLERS, R., BRAS, R. L., GAO, J., AND CHOI, Y. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020* (2020), AAAI Press, pp. 7432–7439.

[7] BOLTE, J., SABACH, S., TEBOULLE, M., AND VAISBOURD, Y. First order methods beyond convexity and lipschitz gradient continuity with applications to quadratic inverse problems. *SIAM Journal on Optimization 28*, 3 (2018), 2131–2151.

[8] CARION, N., MASSA, F., SYNNAEVE, G., USUNIER, N., KIRILLOV, A., AND ZAGORUYKO, S. End-to-end object detection with transformers. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I* (2020), A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, Eds., Lecture Notes in Computer Science, Springer, pp. 213–229.

[9] CHEN, P. H., YU, H.-F., DHILLON, I. S., AND HSIEH, C.-J. Drone: data-aware low-rank compression for large nlp models. In *Proceedings of the 35th International Conference on Neural Information Processing Systems* (Red Hook, NY, USA, 2021), NIPS '21, Curran Associates Inc.

[10] CLARK, P., COWHEY, I., ETZIONI, O., KHOT, T., SABHARWAL, A., SCHOENICK, C., AND TAFJORD, O. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018.

[11] DOSOVITSKIY, A., BEYER, L., KOLESNIKOV, A., WEISSENBORN, D., ZHAI, X., UNTERTHINER, T., DEHGHANI, M., MINDERER, M., HEIGOLD, G., GELLY, S., USZKOREIT, J., AND HOULSBY, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021* (2021), OpenReview.net.

[12] EDMONDS, J. Paths, trees, and flowers. *Canadian Journal of Mathematics 17* (1965), 449–467.

[13] ENGAN, K., AASE, S. O., AND HUSOY, J. H. Method of optimal directions for frame design. In *Proceedings of the 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '99, Phoenix, Arizona, USA, March 15-19, 1999* (1999), IEEE Computer Society, pp. 2443–2446.

[14] FOUCART, S. Hard thresholding pursuit: An algorithm for compressive sensing. *SIAM J. Numer. Anal. 49*, 6 (2011), 2543–2563.

[15] HENDRYCKS, D., BURNS, C., BASART, S., ZOU, A., MAZEIKA, M., SONG, D., AND STEINHARDT, J. Measuring massive multitask language understanding, 2021.

[16] LAI, G., XIE, Q., LIU, H., YANG, Y., AND HOVY, E. Race: Large-scale reading comprehension dataset from examinations, 2017.

[17] MA, Y., XU, G., SUN, X., YAN, M., ZHANG, J., AND JI, R. X-CLIP: end-to-end multi-grained contrastive learning for video-text retrieval. In *MM '22: The 30th ACM International Conference on Multimedia, Lisboa, Portugal, October 10 - 14, 2022* (2022), J. Magalhães, A. D. Bimbo, S. Satoh, N. Sebe, X. Alameda-Pineda, Q. Jin, V. Oria, and L. Toni, Eds., ACM, pp. 638–647.

[18] MACKO, V., AND BOA, V. Macko: Sparse matrix-vector multiplication for low sparsity. *arXiv preprint arXiv: 2511.13061* (2025).

[19] MAKHOV, D., SHOPKHOEV, D., ZHUSSIP, M., ALI, A., AND LEFKIMMIATIS, S. Cospadi: Compressing llms via calibration-guided sparse dictionary learning. *arXiv preprint arXiv: 2509.22075* (2025).

[20] MAKHOV, D., SHOPKHOEV, D., ZHUSSIP, M., ALI, A., MOHAMMAD, B., AND LEFKIMMIATIS, S. Compot: Calibration-optimized matrix procrustes orthogonalization for transformers compression. *arXiv preprint arXiv: 2602.15200* (2026).

[21] MERITY, S., XIONG, C., BRADBURY, J., AND SOCHER, R. Pointer sentinel mixture models, 2016.

[22] OPENAI, :, AGARWAL, S., AHMAD, L., AI, J., ALTMAN, S., APPLEBAUM, A., ARBUS, E., ARORA, R. K., BAI, Y., BAKER, B., BAO, H., BARAK, B., BENNETT, A., BERTAO, T., BRETT, N., BREVDO, E., BROCKMAN, G., BUBECK, S., CHANG, C., CHEN, K., CHEN, M., CHEUNG, E., CLARK, A., COOK, D., DUKHAN, M., DVORAK, C., FIVES, K., FOMENKO, V., GARIPOV, T., GEORGIEV, K., GLAESE, M., GOGINENI, T., GOUCHER, A., GROSS, L., GUZMAN, K. G., HALLMAN, J., HEHIR, J., HEIDECKE, J., HELYAR, A., HU, H., HUET, R., HUH, J., JAIN, S., JOHNSON, Z., KOCH, C., KOFMAN, I., KUNDEL, D., KWON, J., KYRYLOV, V., LE, E. Y., LECLERC, G., LENNON, J. P., LESSANS, S., LEZCANO-CASADO, M., LI, Y., LI, Z., LIN, J., LISS, J., LILY, LIU, LIU, J., LU, K., LU, C., MARTINOVIC, Z., MCCALLUM, L., MCGRATH, J., MCKINNEY, S., MCLAUGHLIN, A., MEI, S., MOSTOVOY, S., MU, T., MYLES, G., NEITZ, A., NICHOL, A., PACHOCKI, J., PAINO, A., PALMIE, D., PANTULIANO, A., PARASCANDOLO, G., PARK, J., PATHAK, L., PAZ, C., PERAN, L., PIMENOV, D., POKRASS, M., PROEHL, E., QIU, H., RAILA, G., RASO, F., REN, H., RICHARDSON, K., ROBINSON, D., ROTSTED, B., SALMAN, H., SANJEEV, S., SCHWARZER, M., SCULLEY, D., SIKCHI, H., SIMON, K., SINGHAL, K., SONG, Y., STUCKEY, D., SUN, Z., TILLET, P., TOIZER, S., TSIMPOURLAS, F., VYAS, N., WALLACE, E., WANG, X., WANG, M., WATKINS, O., WEIL, K., WENDLING, A., WHINNERY, K., WHITNEY, C., WONG, H., YANG, L., YANG, Y., YASUNAGA, M., YING, K., ZAREMBA, W., ZHAN, W., ZHANG, C., ZHANG, B., ZHANG, E., AND ZHAO, S. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv: 2508.10925* (2025).

[23] PAPERNO, D., KRUSZEWSKI, G., LAZARIDOU, A., PHAM, Q. N., BERNARDI, R., PEZZELLE, S., BARONI, M., BOLEDA, G., AND FERNÁNDEZ, R. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers* (2016), The Association for Computer Linguistics.

[24] PENEDO, G., KYDLÍČEK, H., ALLAL, L. B., LOZHKOV, A., MITCHELL, M., RAFFEL, C., WERRA, L. V., AND WOLF, T. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track* (2024).

[25] PENG, Z., YU, J., WANG, W., CHANG, Y., SUN, Y., DONG, L., ZHU, Y., XU, W., BAO, H., WANG, Z., HUANG, S., XIA, Y., AND WEI, F. Vibevoice technical report. *arXiv preprint arXiv: 2508.19205* (2025).

[26] PRATAP, V., XU, Q., SRIRAM, A., SYNNAEVE, G., AND COLLOBERT, R. Mls: A large-scale multilingual dataset for speech research. *ArXiv abs/2012.03411* (2020).

[27] RAZAVIYAYN, M., HONG, M., AND LUO, Z. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM J. Optim. 23*, 2 (2013), 1126–1153.

[28] REIN, D., HOU, B. L., STICKLAND, A. C., PETTY, J., PANG, R. Y., DIRANI, J., MICHAEL, J., AND BOWMAN, S. R. GPQA: A graduate-level google-proof q&a benchmark. *CoRR abs/2311.12022* (2023).

[29] SAEKI, T., XIN, D., NAKATA, W., KORIYAMA, T., TAKAMICHI, S., AND SARUWATARI, H. Utmos: Utokyo-sarulab system for voicemos challenge 2022, 2022.

[30] SHARMA, P., ASH, J. T., AND MISRA, D. The truth is in there: Improving reasoning in language models with layer-selective rank reduction. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024* (2024), OpenReview.net.

[31] SPRAGUE, Z., YE, X., BOSTROM, K., CHAUDHURI, S., AND DURRETT, G. Musr: Testing the limits of chain-of-thought with multistep soft reasoning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024* (2024), OpenReview.net.

[32] SUZGUN, M., SCALES, N., SCHÄRLI, N., GEHRMANN, S., TAY, Y., CHUNG, H. W., CHOWDHERY, A., LE, Q. V., CHI, E. H., ZHOU, D., AND WEI, J. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023* (2023), A. Rogers, J. L. Boyd-Graber, and N. Okazaki, Eds., Association for Computational Linguistics, pp. 13003–13051.

[33] SYLVESTER, J. J. Sur l'équation en matrices $px = xq$. *Comptes Rendus Acad. Sci. Paris 99* (1884), 67–71, 115–116.

[34] TAORI, R., GULRAJANI, I., ZHANG, T., DUBOIS, Y., LI, X., GUESTRIN, C., LIANG, P., AND HASHIMOTO, T. B. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

[35] TEAM, G., KAMATH, A., FERRET, J., PATHAK, S., VIEILLARD, N., MERHEJ, R., PERRIN, S., MATEJOVICOVA, T., RAMÉ, A., RIVIÈRE, M., ROUILLARD, L., MESNARD, T., CIDERON, G., BASTIEN GRILL, J., RAMOS, S., YVINEC, E., CASBON, M., POT, E., PENCHEV, I., LIU, G., VISIN, F., KENEALY, K., BEYER, L., ZHAI, X., TSITSULIN, A., BUSA-FEKETE, R., FENG, A., SACHDEVA, N., COLEMAN, B., GAO, Y., MUSTAFA, B., BARR, I., PARISOTTO, E., TIAN, D., EYAL, M., CHERRY, C., PETER, J.-T., SINOPALNIKOV, D., BHUPATIRAJU, S., AGARWAL, R., KAZEMI, M., MALKIN, D., KUMAR, R., VILAR, D., BRUSILOVSKY, I., LUO, J., STEINER, A., FRIESEN, A., SHARMA, A., SHARMA, A., GILADY, A. M., GOEDECKEMEYER, A., SAADE, A., FENG, A., KOLESNIKOV, A., BENDEBURY, A., ABDAGIC, A., VADI, A., GYÖRGY, A., PINTO, A. S., DAS, A., BAPNA, A., MIECH, A., YANG, A., PATERSON, A., SHENOY, A., CHAKRABARTI, A., PIOT, B., WU, B., SHAHRIARI, B., PETRINI, B., CHEN, C., LAN, C. L., CHOQUETTE-CHOO, C. A., CAREY, C., BRICK, C., DEUTSCH, D., EISENBUD, D., CATTLE, D., CHENG, D., PAPARAS, D., SREEPATHIHALLI, D. S., REID, D., TRAN, D., ZELLE, D., NOLAND, E., HUIZENGA, E., KHARITONOV, E., LIU, F., AMIRKHANYAN, G., CAMERON, G., HASHEMI, H., KLIMCZAK-PLUCISKA, H., SINGH, H., MEHTA, H., LEHRI, H. T., HAZIMEH, H., BALLANTYNE, I., SZPEKTOR, I., NARDINI, I., POUGET-ABADIE, J., CHAN, J., STANTON, J., WIETING, J., LAI, J., ORBAY, J., FERNANDEZ, J., NEWLAN, J., YEONG JI, J., SINGH, J., BLACK, K., YU, K., HUI, K., VODRAHALLI, K., GREFF, K., QIU, L., VALENTINE, M., COELHO, M., RITTER, M., HOFFMAN, M., WATSON, M., CHATURVEDI, M., MOYNIHAN, M., MA, M., BABAR, N., NOY, N., BYRD, N., ROY, N., MOMCHEV, N., CHAUHAN, N., SACHDEVA, N., BUNYAN, O., BOTARDA, P., CARON, P., RUBENSTEIN, P. K., CULLITON, P., SCHMID, P., SESSA, P. G., XU, P., STANCZYK, P., TAFTI, P., SHIVANNA, R., WU, R., PAN, R., ROKNI, R., WILLOUGHBY, R., VALLU, R., MULLINS, R., JEROME, S.,

SMOOT, S., GIRGIN, S., IQBAL, S., REDDY, S., SHETH, S., PÕDER, S., BHATNAGAR, S., PANYAM, S. R., EIGER, S., ZHANG, S., LIU, T., YACOVONE, T., LIECHTY, T., KALRA, U., EVCI, U., MISRA, V., ROSEBERRY, V., FEINBERG, V., KOLESNIKOV, V., HAN, W., KWON, W., CHEN, X., CHOW, Y., ZHU, Y., WEI, Z., EGYED, Z., COTRUTA, V., GIANG, M., KIRK, P., RAO, A., BLACK, K., BABAR, N., LO, J., MOREIRA, E., MARTINS, L. G., SANSEVIERO, O., GONZALEZ, L., GLEICHER, Z., WARKENTIN, T., MIRROKNI, V., SENTER, E., COLLINS, E., BARRAL, J., GHAHRAMANI, Z., HADSELL, R., MATIAS, Y., SCULLEY, D., PETROV, S., FIEDEL, N., SHAZEER, N., VINYALS, O., DEAN, J., HASSABIS, D., KAVUKCUOGLU, K., FARABET, C., BUCHATSKAYA, E., ALAYRAC, J.-B., ANIL, R., DMITRY, LEPIKHIN, BORGEAUD, S., BACHEM, O., JOULIN, A., ANDREEV, A., HARDIN, C., DADASHI, R., AND HUSSENOT, L. Gemma 3 technical report. *arXiv preprint arXiv: 2503.19786* (2025).

[36] TOUVRON, H., LAVRIL, T., IZACARD, G., MARTINET, X., LACHAUX, M.-A., LACROIX, T., ROZIÈRE, B., GOYAL, N., HAMBRO, E., AZHAR, F., RODRIGUEZ, A., JOULIN, A., GRAVE, E., AND LAMPLE, G. Llama: Open and efficient foundation language models. *arXiv preprint arXiv: 2302.13971* (2023).

[37] WAN, T., WANG, A., AI, B., WEN, B., MAO, C., XIE, C.-W., CHEN, D., YU, F., ZHAO, H., YANG, J., ZENG, J., WANG, J., ZHANG, J., ZHOU, J., WANG, J., CHEN, J., ZHU, K., ZHAO, K., YAN, K., HUANG, L., FENG, M., ZHANG, N., LI, P., WU, P., CHU, R., FENG, R., ZHANG, S., SUN, S., FANG, T., WANG, T., GUI, T., WENG, T., SHEN, T., LIN, W., WANG, W., WANG, W., ZHOU, W., WANG, W., SHEN, W., YU, W., SHI, X., HUANG, X., XU, X., KOU, Y., LV, Y., LI, Y., LIU, Y., WANG, Y., ZHANG, Y., HUANG, Y., LI, Y., WU, Y., LIU, Y., PAN, Y., ZHENG, Y., HONG, Y., SHI, Y., FENG, Y., JIANG, Z., HAN, Z., WU, Z.-F., AND LIU, Z. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv: 2503.20314* (2025).

[38] WANG, J., CHEN, Y., LIN, I., LI, B., AND ZHANG, G. L. Basis sharing: Cross-layer parameter sharing for large language model compression. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025* (2025), OpenReview.net.

[39] WANG, X., ZHENG, Y., WAN, Z., AND ZHANG, M. SVD-LLM: Truncation-aware singular value decomposition for large language model compression. In *International Conference on Learning Representations* (2025).

[40] WANG, Y., MA, X., ZHANG, G., NI, Y., CHANDRA, A., GUO, S., REN, W., ARULRAJ, A., HE, X., JIANG, Z., LI, T., KU, M., WANG, K., ZHUANG, A., FAN, R., YUE, X., AND CHEN, W. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024* (2024), A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, Eds.

[41] WELBL, J., LIU, N. F., AND GARDNER, M. Crowdsourcing multiple choice science questions, 2017.

[42] YANG, A., LI, A., YANG, B., ZHANG, B., HUI, B., ZHENG, B., YU, B., GAO, C., HUANG, C., LV, C., ZHENG, C., LIU, D., ZHOU, F., HUANG, F., HU, F., GE, H., WEI, H., LIN, H., TANG, J., YANG, J., TU, J., ZHANG, J., YANG, J., YANG, J., ZHOU, J., ZHOU, J., LIN, J., DANG, K., BAO, K., YANG, K., YU, L., DENG, L., LI, M., XUE, M., LI, M., ZHANG, P., WANG, P., ZHU, Q., MEN, R., GAO, R., LIU, S., LUO, S., LI, T., TANG, T., YIN, W., REN, X., WANG, X., ZHANG, X., REN, X., FAN, Y., SU, Y., ZHANG, Y., ZHANG, Y., WAN, Y., LIU, Y., WANG, Z., CUI, Z., ZHANG, Z., ZHOU, Z., AND QIU, Z. Qwen3 technical report. *arXiv preprint arXiv: 2505.09388* (2025).

[43] YUAN, Z., SHANG, Y., SONG, Y., YANG, D., WU, Q., YAN, Y., AND SUN, G. ASVD: Activation-aware singular value decomposition for compressing large language models. In *International Conference on Learning Representations* (2024).

[44] ZELLERS, R., HOLTZMAN, A., BISK, Y., FARHADI, A., AND CHOI, Y. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers* (2019), A. Korhonen, D. R. Traum, and L. Màrquez, Eds., Association for Computational Linguistics, pp. 4791–4800.

[45] ZHOU, J., LU, T., MISHRA, S., BRAHMA, S., BASU, S., LUAN, Y., ZHOU, D., AND HOU, L. Instruction-following evaluation for large language models. *CoRR abs/2311.07911* (2023).

[46] ZHUSSIP, M., SHOPKHOEV, D., ALI, A., AND LEFKIMMIATIS, S. Share your attention: Transformer weight sharing via matrix-based dictionary learning. In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026* (2026), S. Koenig, C. Jenkins, and M. E. Taylor, Eds., AAAI Press, pp. 29260–29268.

A Theoretical background

A.1 Background and recap

Sylvester Equation. The classical Sylvester equation is a linear matrix equation of the form

$$\mathbf{AX} + \mathbf{XB} = \mathbf{C}, \tag{18}$$

where $\mathbf{A} \in \mathbb{R}^{m\times m}$, $\mathbf{B} \in \mathbb{R}^{n\times n}$, and $\mathbf{C} \in \mathbb{R}^{m\times n}$ are given matrices, and $\mathbf{X} \in \mathbb{R}^{m\times n}$ is the unknown. This equation arises frequently in control theory, model order reduction, and structured matrix factorization. A unique solution exists if and only if the spectra of $\mathbf{A}$ and $-\mathbf{B}$ are disjoint, i.e., $\lambda_i(\mathbf{A}) + \lambda_j(\mathbf{B}) \neq 0$ for all eigenvalue pairs $(\lambda_i(\mathbf{A}), \lambda_j(\mathbf{B}))$ [33]. While the system can be vectorized using Kronecker products as $(\mathbf{I}_n \otimes \mathbf{A} + \mathbf{B}^\mathsf{T} \otimes \mathbf{I}_m)\mathrm{vec}(\mathbf{X}) = \mathrm{vec}(\mathbf{C})$, direct inversion scales poorly ($\mathcal{O}(m^3n^3)$) and is numerically unstable for large dimensions. Instead, the Bartels-Stewart algorithm [4] exploits Schur decompositions to solve the system efficiently in $\mathcal{O}(m^3 + n^3)$ time with guaranteed numerical stability.

Generalized Sylvester Equation. In our dictionary update step, we encounter a *two-term generalized Sylvester equation* of the form

$$\mathbf{A}_1\mathbf{X}\mathbf{B}_1 + \mathbf{A}_2\mathbf{X}\mathbf{B}_2 = \mathbf{C}. \tag{19}$$

More generally, an m-term generalized Sylvester equation takes the form $\sum_{k=1}^{m} \mathbf{A}_k\mathbf{X}\mathbf{B}_k = \mathbf{C}$. Unlike the classical case, no universal closed-form solution exists, and the feasibility of the solution depends on the spectral properties of the matrix pencils $(\mathbf{A}_1, \mathbf{A}_2)$ and $(\mathbf{B}_1, \mathbf{B}_2)$. However, when $\mathbf{A}_1$ and $\mathbf{B}_1$ are symmetric positive definite and $\mathbf{A}_2, \mathbf{B}_2$ are symmetric positive semi-definite, the system can be decoupled via simultaneous diagonalization. In our case, strict positive definiteness of $\mathbf{B}_1$ is enforced via Tikhonov regularization ($\tilde{\mathbf{B}}_1 = \mathbf{B}_1 + \epsilon\mathbf{I}$) to ensure the generalized eigenvalue decomposition is well-posed. Specifically, we compute the generalized eigenvalue decompositions (GEVD) $(\mathbf{P}, \mathbf{\Lambda}) = \Phi(\mathbf{A}_2, \mathbf{A}_1)$ and $(\mathbf{Q}, \mathbf{\Sigma}) = \Phi(\mathbf{B}_2, \mathbf{B}_1)$, which satisfy

$$\mathbf{P}^\mathsf{T}\mathbf{A}_1\mathbf{P} = \mathbf{I}, \quad \mathbf{P}^\mathsf{T}\mathbf{A}_2\mathbf{P} = \mathbf{\Lambda}, \qquad \mathbf{Q}^\mathsf{T}\mathbf{B}_1\mathbf{Q} = \mathbf{I}, \quad \mathbf{Q}^\mathsf{T}\mathbf{B}_2\mathbf{Q} = \mathbf{\Sigma}. \tag{20}$$

Substituting $\mathbf{X} = \mathbf{P}\mathbf{Y}\mathbf{Q}^T$ into Eq.(13) transforms the coupled matrix system into a set of independent scalar equations:

$$(1 + \lambda_i\sigma_j)y_{ij} = (\mathbf{P}^T\mathbf{C}\mathbf{Q})_{ij}, \quad \forall i, j, \tag{21}$$

where λ_i and σ_j denote the diagonal entries of $\mathbf{\Lambda}$ and $\mathbf{\Sigma}$, respectively. Provided that $1 + \lambda_i\sigma_j \neq 0$ for all i, j a condition naturally satisfied in our setting since all eigenvalues are non-negative the solution admits a closed-form expression via element-wise division:

$$\mathbf{Y} = \frac{\mathbf{P}^T\mathbf{C}\mathbf{Q}}{\mathbf{1}\mathbf{1}^T + \boldsymbol{\lambda}\boldsymbol{\sigma}^T}, \tag{22}$$

where $\boldsymbol{\lambda} = \mathrm{diag}(\mathbf{\Lambda})$ and $\boldsymbol{\sigma} = \mathrm{diag}(\mathbf{\Sigma})$ are eigenvalue vectors, and the denominator represents the outer product structure for proper broadcasting. The final solution is recovered as $\mathbf{X} = \mathbf{P}\mathbf{Y}\mathbf{Q}^\mathsf{T}$. This simultaneous diagonalization strategy avoids iterative solvers, eliminates step-size tuning, and guarantees deterministic convergence with $\mathcal{O}(d^3)$ complexity dominated by the initial GEVD computation.

For the dictionary update in Section 3.2, identify

$$\mathbf{A}_i = \bar{\mathbf{G}}_i, \qquad \mathbf{B}_i = \tilde{\mathbf{B}}_{i,t} = \mathbf{C}_{i,t}\mathbf{C}_{i,t}^\mathsf{T} + \epsilon\mathbf{I}.$$

The normalized activation Gram matrices $\bar{\mathbf{G}}_i$ are positive definite for $\eta > 0$ and $s_i > 0$, while $\tilde{\mathbf{B}}_{i,t}$ are positive definite for $\epsilon > 0$. The generalized eigenvalues are therefore positive, and $1 + \lambda_a\sigma_b > 0$ for all a, b. The resulting solution is the unique minimizer of the regularized dictionary subproblem.

Hard Thresholding Pursuit. Hard Thresholding Pursuit (HTP) [14] is a greedy iterative algorithm designed to solve ℓ_0-constrained least squares problems, widely adopted in compressive sensing and sparse dictionary learning. Unlike basic thresholding schemes that merely zero out non-dominant coefficients, HTP enhances reconstruction accuracy by coupling a gradient-based support identification step with a least-squares projection onto the selected subspace. Given an overcomplete

system $\mathbf{y} = \mathbf{\Phi c} + \boldsymbol{\epsilon}$, HTP seeks a k-sparse solution through the following canonical iteration at step j:

$$\mathbf{c}^{\text{tmp}} = \mathbf{c}^{(j)} + \mu \mathbf{\Phi}^{\mathsf{T}} \left(\mathbf{y} - \mathbf{\Phi c}^{(j)} \right), \quad \mathbf{S}^{(j+1)} = \operatorname{supp}\big(\mathcal{H}_k(\mathbf{c}^{\text{tmp}})\big), \tag{23}$$

$$\mathbf{c}^{(j+1)} = \underset{\mathbf{c}\,:\,\operatorname{supp}(\mathbf{c}) \subseteq \mathbf{S}^{(j+1)}}{\arg\min} \|\mathbf{y} - \mathbf{\Phi c}\|_2^2, \tag{24}$$

where $\mathcal{H}_k(\cdot)$ retains the k entries of largest magnitude, $\operatorname{supp}(\cdot)$ extracts the corresponding index set, and $\mu > 0$ is a step size. This restricted projection ensures that the active coefficients are optimally fitted to the measurements, yielding superior convergence and fidelity over pure iterative thresholding.

For a fixed dictionary $\mathbf{D}_{t-1}$, the coefficient subproblem is

$$\begin{aligned} \min_{\mathbf{C}} \quad & \left\| \bar{\mathbf{L}}_i (\mathbf{W}_i - \mathbf{D}_{t-1}\mathbf{C}) \right\|_F^2 + \varepsilon \|\mathbf{C}\|_F^2 \\ \text{s.t.} \quad & \|\mathbf{C}[:, j]\|_0 \le k, \qquad j = 1, \ldots, d_i. \end{aligned} \tag{25}$$

Define

$$\mathbf{H}_{i,t} = \mathbf{D}_{t-1}^{\mathsf{T}} \bar{\mathbf{G}}_i \mathbf{D}_{t-1} + \varepsilon \mathbf{I}, \qquad \mathbf{R}_{i,t} = \mathbf{D}_{t-1}^{\mathsf{T}} \bar{\mathbf{G}}_i \mathbf{W}_i.$$

Initialize $\mathbf{C}_i^{(0)} = \mathbf{C}_{i,t-1}$. Each HTP inner iteration consists of:

1. **Gradient Update:**

$$\mathbf{C}_i^{\text{tmp},(h)} = \mathbf{C}_i^{(h)} + \mu(\mathbf{R}_{i,t} - \mathbf{H}_{i,t}\mathbf{C}_i^{(h)}), \qquad \mu = 1.$$

2. **Support Selection:** Select the k largest-magnitude entries in each column of $\mathbf{C}_i^{\text{tmp},(h)}$ to form $\mathbf{M}_i^{(h+1)} \in \{0,1\}^{r \times d_i}$.
3. **Restricted Projection:** Approximately solve

$$\mathbf{M}_i^{(h+1)} \odot (\mathbf{H}_{i,t}\mathbf{C} - \mathbf{R}_{i,t}) = \mathbf{0}, \qquad \mathbf{C} = \mathbf{C} \odot \mathbf{M}_i^{(h+1)},$$

using batched CG initialized with $\mathbf{C}_i^{(h)} \odot \mathbf{M}_i^{(h+1)}$. The resulting coefficients define $\mathbf{C}_i^{(h+1)}$.

After T_{HTP} iterations, set $\mathbf{C}_{i,t} = \mathbf{C}_i^{(T_{\text{HTP}})}$. We use two HTP iterations and at most ten CG iterations per restricted solve. The scope of the convergence claims is discussed in Appendix A.2.

A.2 Shared Dictionary Optimization

Optimization Objective. With the coefficients fixed, consider

$$J(\mathbf{D}) = \sum_{i=1}^{2} \left[\left\| \bar{\mathbf{L}}_i (\mathbf{W}_i - \mathbf{D}\mathbf{C}_{i,t}) \right\|_F^2 + \epsilon \left\| \bar{\mathbf{L}}_i \mathbf{D} \right\|_F^2 \right]. \tag{26}$$

Define

$$\bar{\mathbf{G}}_i = \bar{\mathbf{L}}_i^{\mathsf{T}} \bar{\mathbf{L}}_i, \qquad \tilde{\mathbf{B}}_{i,t} = \mathbf{C}_{i,t}\mathbf{C}_{i,t}^{\mathsf{T}} + \epsilon \mathbf{I}.$$

Differentiating gives

$$\nabla_{\mathbf{D}} J = 2 \sum_{i=1}^{2} \bar{\mathbf{G}}_i \left(\mathbf{D}\tilde{\mathbf{B}}_{i,t} - \mathbf{W}_i \mathbf{C}_{i,t}^{\mathsf{T}} \right). \tag{27}$$

Setting the gradient to zero yields

$$\bar{\mathbf{G}}_1 \mathbf{D}_t \tilde{\mathbf{B}}_{1,t} + \bar{\mathbf{G}}_2 \mathbf{D}_t \tilde{\mathbf{B}}_{2,t} = \mathbf{K}_t, \tag{28}$$

where

$$\mathbf{K}_t = \sum_{i=1}^{2} \bar{\mathbf{G}}_i \mathbf{W}_i \mathbf{C}_{i,t}^{\mathsf{T}}.$$

This is Eq. (5) of the main text.

Generalized Sylvester Equation. Compute

$$(\mathbf{P}, \mathbf{\Lambda}) = \Phi(\bar{\mathbf{G}}_2, \bar{\mathbf{G}}_1), \qquad (\mathbf{Q}_t, \mathbf{\Sigma}_t) = \Phi(\tilde{\mathbf{B}}_{2,t}, \tilde{\mathbf{B}}_{1,t}).$$

Writing $\mathbf{D}_t = \mathbf{P}\mathbf{Y}_t\mathbf{Q}_t^\mathsf{T}$ decouples the system into

$$(1 + \lambda_a \sigma_{t,b})(Y_t)_{ab} = (\mathbf{P}^\mathsf{T}\mathbf{K}_t\mathbf{Q}_t)_{ab}, \tag{29}$$

where $\boldsymbol{\lambda} = \operatorname{diag}(\mathbf{\Lambda})$ and $\boldsymbol{\sigma}_t = \operatorname{diag}(\mathbf{\Sigma}_t)$. Hence

$$\mathbf{D}_t = \mathbf{P}\left(\frac{\mathbf{P}^\mathsf{T}\mathbf{K}_t\mathbf{Q}_t}{\mathbf{1}_d\mathbf{1}_r^\mathsf{T} + \boldsymbol{\lambda}\boldsymbol{\sigma}_t^\mathsf{T}}\right)\mathbf{Q}_t^\mathsf{T}. \tag{30}$$

The normalized activation Gram matrices are positive definite for $\eta > 0$ and $s_i > 0$. The coefficient Gram matrices $\tilde{\mathbf{B}}_{i,t}$ are positive definite for $\epsilon > 0$. Their generalized eigenvalues are therefore positive, so every denominator is nonzero. The resulting update is the unique minimizer of the regularized dictionary subproblem.

Convergence Scope and Practical Behavior. The alternating procedure combines a dictionary update with dense or sparse coefficient updates. The dense coefficient update exactly solves its ridge-regularized quadratic subproblem. The sparse coefficient update uses a finite number of HTP iterations with approximate restricted solves via CG. Exact solutions of individual quadratic subproblems do not imply global optimality of the joint factorization.

Conditions for Convergence. BSUM and Kurdyka–Łojasiewicz analyses provide convergence results under suitable assumptions on the objective and update sequence [27, 7]. Applying such results requires verifying conditions including bounded iterates, sufficient decrease, and appropriate control of inexact updates.

For fixed calibration statistics, the reconstruction term and coefficient penalties are polynomial functions of the factors, and the column-wise sparsity constraints define a semi-algebraic feasible set. This structure alone does not establish convergence of the implemented algorithm. In particular, an ℓ_0 constraint does not bound coefficient magnitudes, and normalization of the calibration operators does not establish boundedness of the factor iterates.

Finite-Step Sparse Updates. A fixed HTP/CG iteration budget does not by itself guarantee sufficient decrease or the inexactness conditions required by convergence analyses. We do not establish these conditions for the implemented finite-step solver. We also do not verify RIP-type recovery assumptions for the evolving operators $\bar{\mathbf{L}}_i\mathbf{D}_{t-1}$. Accordingly, standalone HTP recovery guarantees [14] do not establish convergence of the complete alternating procedure.

Empirical Evaluation. Table 12 evaluates the runtime and downstream performance obtained with different HTP and CG iteration budgets. We select two HTP iterations and at most ten CG iterations per restricted solve as a practical runtime–accuracy trade-off. These measurements support the selected computational budget; they do not constitute a proof of convergence or optimality.

B Implementation Details

Budget Allocation Consider a pair of weight matrices $W_1 \in \mathbb{R}^{d\times d_1}$ and $W_2 \in \mathbb{R}^{d\times d_2}$ sharing the input dimension d. Let $D \in \mathbb{R}^{d\times r}$ be the shared dictionary and $C_i \in \mathbb{R}^{r\times d_i}$ the coefficient matrix of W_i, with k non-zero entries per coefficient column. We define

$$\mathrm{CR} = 1 - \frac{S_{\mathrm{comp}}}{S_{\mathrm{raw}}}, \qquad \mathrm{KS} = \frac{r}{k}. \tag{31}$$

The raw and compressed parameter counts are

$$S_{\mathrm{raw}} = d(d_1 + d_2), \qquad S_{\mathrm{comp}} = dr + k(d_1 + d_2). \tag{32}$$

Using $k = \lfloor r/\mathrm{KS} \rfloor$, the rank r is chosen to satisfy the storage budget

$$dr + \left\lfloor \frac{r}{\mathrm{KS}} \right\rfloor (d_1 + d_2) \le (1 - \mathrm{CR})\, d(d_1 + d_2). \tag{33}$$

Equivalently, using the continuous relaxation $k \approx r/\mathrm{KS}$, we set

$$r = \left\lfloor \frac{(1 - \mathrm{CR})\, d(d_1 + d_2)}{d + (d_1 + d_2)/\mathrm{KS}} \right\rfloor, \qquad k = \left\lfloor \frac{r}{\mathrm{KS}} \right\rfloor. \tag{34}$$

Benchmarks In our experiments, we mainly evaluate our method in a zero-shot setting on the following benchmarks: PIQA [6], HellaSwag [44], OpenAI LAMBADA [23], ARC-Easy and ARC-Challenge [10], SciQ [41], RACE [16], and MMLU [15] following the evaluation protocols established in prior work to ensure a fair comparison. While the results reported in Table 2 used slightly different benchmarks to follow the experiments of the corresponding pruning papers. Additionally, we evaluate Qwen3-8B on a more recent benchmark suite in Table 10.

Efficient Computation of the Normalized Frobenius Distance. The grouping metric in Eq. (6) requires evaluating $\|\mathbf{W}_j^{[:,k:k+d_i]} - \mathbf{W}_i\|_F$ for all valid column shifts $k \in [0, d_j - d_i]$. A naive implementation materializes each submatrix $\mathbf{W}_j^{[:,k:k+d_i]}$, incurring $\mathcal{O}(d \cdot d_i \cdot (d_j - d_i))$ memory overhead and redundant tensor allocations. To eliminate this bottleneck, we expand the squared Frobenius norm algebraically:

$$\|\mathbf{W}_j^{[:,k:k+d_i]} - \mathbf{W}_i\|_F^2 = \|\mathbf{W}_i\|_F^2 + \|\mathbf{W}_j^{[:,k:k+d_i]}\|_F^2 - 2\langle \mathbf{W}_i, \mathbf{W}_j^{[:,k:k+d_i]} \rangle_F. \tag{35}$$

The first term is constant across all shifts. The second term corresponds to the sum of squared column norms over a sliding window of width d_i. We compute this in $\mathcal{O}(d_j)$ time by precomputing a prefix sum (cumulative sum) of the column-wise squared ℓ_2 norms of $\mathbf{W}_j$. The third term is a sliding-window inner product, which is mathematically equivalent to a 1D cross-correlation operation. Specifically, for each row $r \in \{1, \ldots, d\}$, we compute the cross-correlation between the r-th row of $\mathbf{W}_i$ and the r-th row of $\mathbf{W}_j$, then sum the correlation outputs across all rows. This reduces the entire distance computation to a sequence of vectorized prefix sums and 1D convolutions. The cross-correlation for each row requires $\mathcal{O}(d_i \cdot (d_j - d_i))$ operations, yielding a total time complexity of $\mathcal{O}(d \cdot d_i \cdot (d_j - d_i))$ with $\mathcal{O}(1)$ auxiliary memory beyond the input and output tensors. In the worst case where $d_i \approx d_j$, this simplifies to $\mathcal{O}(d \cdot d_i^2)$, though in practice the sliding-window computation is highly optimized on modern GPU hardware.

Symmetry Handling. To construct a valid undirected graph for Edmonds' Blossom algorithm, we enforce symmetry by computing $\delta(\mathbf{W}_i, \mathbf{W}_j)$ once and assigning $w_{ij} = w_{ji} = C - \delta(\mathbf{W}_i, \mathbf{W}_j)$. When $d_i = d_j$, we normalize by the smaller Frobenius norm $\min(\|\mathbf{L}_i \mathbf{W}_i\|_F, \|\mathbf{L}_j \mathbf{W}_j\|_F) + \epsilon$ to ensure a conservative error estimate and symmetric treatment of equal-dimensional matrices.

HTP and Conjugate Gradient Integration. We approximately solve the regularized, support-restricted subproblem in Eq. (11) using batched Conjugate Gradient (CG). For the current support mask $\mathbf{M}_i^{(h+1)}$, the restricted normal equations are

$$\mathbf{M}_i^{(h+1)} \odot (\mathbf{H}_{i,t}\mathbf{C} - \mathbf{R}_{i,t}) = \mathbf{0}, \qquad \mathbf{C} = \mathbf{C} \odot \mathbf{M}_i^{(h+1)},$$

where $\mathbf{H}_{i,t}$ and $\mathbf{R}_{i,t}$ are defined in Eq. (9). We initialize CG with $\mathbf{C}_i^{(h)} \odot \mathbf{M}_i^{(h+1)}$, set its relative tolerance parameter to 10^{-3}, and allow at most ten iterations per restricted solve. The solves are batched across coefficient columns.

Blossom Matching Complexity. We use Edmonds' Blossom algorithm [12] via NetworkX to solve the maximum-weight matching formulation of layer grouping. With worst-case $\mathcal{O}(N^3)$ complexity for N candidate weights, this step is negligible in practice: for an 80-layer model with seven projections per layer ($N = 560$), matching completes in < 2 seconds on CPU.

C Sparse Storage Format and Serving Efficiency

C.1 MACKO-style sparse coefficient encoding

For a paired compression of two weight matrices $W_1 \in \mathbb{R}^{d \times d_1}$ and $W_2 \in \mathbb{R}^{d \times d_2}$, GeoPair stores a shared dictionary $D \in \mathbb{R}^{d \times r}$ and sparse coefficient matrices $C_i \in \mathbb{R}^{r \times d_i}$, with k non-zero entries per coefficient column. The Cholesky whitening factors L_i are used only during compression and are not stored in the final checkpoint. Pair-assignment metadata is negligible.

Following the CoSpaDi-compatible sparse deployment protocol [19], we store the sparse coefficient matrices using a MACKO-style delta-encoded format [18]. For each nonzero coefficient, we store both its quantized value and its delta-encoded index. We use

$$b_{\text{val}} = 14 \quad \text{value bits}, \qquad b_\Delta = 2 \quad \text{index-delta bits},$$

so each stored sparse record costs

$$b_{\text{val}} + b_{\Delta} = 16 \text{ bits}.$$

Importantly, this format does not store a separate r-bit mask per coefficient column. The 2-bit delta field is part of the same sparse record and is not used to pay for an additional bitmask.

For a pair of weights $W_1 \in \mathbb{R}^{d\times d_1}$ and $W_2 \in \mathbb{R}^{d\times d_2}$, the nominal pair-level raw and compressed parameter counts are

$$S_{\text{raw}}^{\text{pair}} = d(d_1 + d_2), \qquad S_{\text{comp}}^{\text{pair}} = dr + k(d_1 + d_2),$$

with

$$k = \left\lfloor \frac{r}{\text{KS}} \right\rfloor.$$

The corresponding compression ratio is therefore

$$\text{CR} = 1 - \frac{S_{\text{comp}}^{\text{pair}}}{S_{\text{raw}}^{\text{pair}}}.$$

Equivalently, solving for r under the target budget gives

$$r = \left\lfloor \frac{(1 - \text{CR})\, d(d_1 + d_2)}{d + (d_1 + d_2)/\text{KS}} \right\rfloor, \qquad k = \left\lfloor \frac{r}{\text{KS}} \right\rfloor.$$

In bits, the nominal compressed size of the pair is

$$B_{\text{nom}}^{\text{pair}} = 16\,(dr + k(d_1 + d_2)).$$

The actual serialized checkpoint size additionally includes sparse-format padding and row pointers:

$$B_{\text{actual}} = 16\, S_{\text{uncompressed}} + 16\, dr + \sum_{i=1}^{2} (16\, \text{nnz}_{\text{pad}}(C_i) + B_{\text{ptr},i}),$$

where $S_{\text{uncompressed}}$ denotes parameters that are left uncompressed, $\text{nnz}_{\text{pad}}(C_i)$ is the padded number of stored nonzeros in C_i, and $B_{\text{ptr},i}$ is the small pointer overhead of the sparse layout.

The MACKO-style delta encoding keeps the index overhead bounded. In the worst case, the number of stored entries per coefficient column is at most

$$k + \frac{r - k}{2^{b_\Delta}}.$$

For our default $\text{KS} = 2.5$ and $b_\Delta = 2$, this gives $k = 0.4r$ and an additional padding budget of at most $0.15r$, so the worst-case number of stored entries per column is at most $0.55r$, which remains well below dense storage.

Table 6 reports the measured serialized checkpoint sizes for Llama-3-8B. The sizes include the dictionary, sparse coefficient values, MACKO-style deltas/padding, and sparse pointer metadata.

Table 6: Measured serialized checkpoint sizes for Llama-3-8B.

Paired projections	Dense size	GeoPair size
up/gate + q/o	160 MB	111.5 MB
up/gate + k/v	136 MB	95.5 MB
q + o	64 MB	42 MB
q/o + k/v	40 MB	26 MB
down + down	256 MB	180 MB
Total model	14.9 GB	10.4 GB

C.2 Serving throughput and latency

We evaluate serving efficiency using the same asynchronous serving protocol as CoSpaDi [19]. For direct comparison with CoSpaDi's ratio-2 deployment protocol, we evaluate the selective up/gate configuration, where the up/gate projections are compressed with the CoSpaDi-compatible sparse layout. The main quality tables use the default GeoPair configuration unless stated otherwise.

Table 7 reports asynchronous serving throughput at prompt length 1, relative to the dense uncompressed baseline. Values greater than $1.0\times$ indicate faster serving than the dense model. GeoPair achieves 1.08–$1.30\times$ throughput relative to the dense baseline and matches CoSpaDi within measurement noise.

Table 7: Asynchronous serving throughput at prompt length 1, relative to the dense baseline. Higher is better.

Method	1 user	4 users	8 users	16 users
SVD-LLM	$1.29\times$	$1.33\times$	$1.35\times$	$0.99\times$
CoSpaDi	$1.24\times$	$1.30\times$	$1.29\times$	$1.13\times$
GeoPair	$1.30\times$	$1.26\times$	$1.26\times$	$1.14\times$

Table 8 reports GeoPair throughput across prompt lengths and concurrency levels. GeoPair remains faster than the dense baseline in all tested configurations. Since autoregressive serving latency is inversely related to token throughput at a fixed concurrency level, these results also indicate a corresponding reduction in average generation latency.

Table 8: GeoPair asynchronous serving throughput across prompt lengths and concurrency levels, relative to the dense baseline. Higher is better.

Users	Prompt 1	Prompt 16	Prompt 32	Prompt 64	Prompt 128
1	$1.30\times$	$1.14\times$	$1.14\times$	$1.14\times$	$1.09\times$
4	$1.26\times$	$1.11\times$	$1.13\times$	$1.08\times$	$1.11\times$
8	$1.26\times$	$1.14\times$	$1.17\times$	$1.16\times$	$1.08\times$
16	$1.14\times$	$1.15\times$	$1.15\times$	$1.13\times$	$1.16\times$

Finally, we measure peak serving memory on Llama-3-8B under the same protocol. The dense model requires $15.06\,\text{GB}$ peak memory, while GeoPair requires approximately $11.70\,\text{GB}$ and CoSpaDi requires $11.42\,\text{GB}$. The small gap between GeoPair and CoSpaDi comes from differences in the learned dictionary rank and coefficient sparsity, while both methods use the same sparse serving pattern and therefore have comparable runtime behavior.

Overall, these results show that GeoPair's gains are obtained at the compression-quality level: it learns better shared dictionaries and coefficients under distinct whitening spaces, while inheriting the practical deployment characteristics of the host sparse serving method.

D Additional Results

D.1 Statistical Significance

To assess the robustness of our compression method, we evaluated the Llama1B model at a 40% compression ratio across five independent random seeds with varied data sampling. The average accuracy was $45.85\% \pm 0.19\%$ (mean $\pm$ standard deviation, $N = 5$), and the average Lambada perplexity was 20.74 ± 0.76 . This indicates that performance is stable across seeds. All results are reproducible with fixed seeds and reported library versions in supplementary.

D.2 Comparison with Basis sharing and Cospadi

Table 9 compares GeoPair with two compression methods, Basis Sharing and CoSpaDi. For a fair comparison, we align with their experimental setup by employing consecutive grouping and omitting weight sharing for down- and out-projection layers. We evaluate performance at compression ratios

of 0.2, 0.3, and 0.4 across multiple benchmarks. Results are reported for both our dense variant (GeoPair*) and the full pipeline (GeoPair). As shown, GeoPair consistently surpasses the baselines, achieving higher average accuracy and lower perplexity across all compression levels.

Table 9: Performance comparison of different compression methods. We denote GeoPair* as our method with dense factorization (no sparsification) using basis-sharing grouping, and GeoPair as our full pipeline with both sparsification and grouping.

Model	Method	Accuracy↑									Perplexity↓	
		PIQA	Hella Swag	LAMBADA	ARC-e	ARC-c	SciQ	Race	MMLU	Avg.	Wiki Text	LAMBADA
Llama2 7B	–	78.9	76.1	73.8	74.2	45.8	91.4	39.7	40.8	65.1	8.7	3.4
0.2	Basis Sharing	71.1	59.9	62.8	60.2	37.8	85	34.7	25	54.6	15.17	7.03
	GeoPair*	71.5	60.6	63.5	62.6	35.2	86.5	35.8	24.9	**55.1**	**14.87**	**6.67**
	CoSpaDi (grouped)	75.5	66.5	71.1	68.5	38.9	88.7	38.5	26.5	59.3	11.7	4.4
	GeoPair	77.4	71.7	72.0	72.3	41.3	90.4	39.9	34.9	**62.5**	**10.09**	**4.05**
0.3	Basis Sharing	66.5	50.3	53.6	54.2	29.3	81.4	32.4	23.3	48.9	22.2	13.2
	GeoPair*	67.0	51.4	54.2	54.6	30.1	81.8	33.1	23.2	**49.5**	**21.29**	**12.48**
	CoSpaDi (grouped)	70.7	58.4	64.5	63.6	35.7	87.2	36.1	23.7	55.0	15.4	**6.5**
	GeoPair	73.2	62.8	63.5	66.9	36.4	92.5	38.0	27.8	**57.7**	**14.1**	6.89
0.4	Basis Sharing	60.7	41.5	41.0	44.6	26.5	75.4	30.1	23.2	42.9	39.6	36.5
	GeoPair*	61.2	42.5	40.8	46.3	27.0	78.4	31.0	22.9	**43.8**	**37.8**	**34.54**
	CoSpaDi (grouped)	64.6	48.1	52.0	51.9	28.9	80.5	32.7	23.3	47.8	25	14.7
	GeoPair	70.9	59.7	61.3	65.3	35.2	91.2	37.2	25.7	**55.8**	**15.64**	**7.64**

D.3 Evaluations on another set of benchmarks

To assess the robustness of our method beyond conventional evaluation suites, we report results on a set of advanced, challenging benchmarks that reflect the evolving demands placed on modern large language models. These include IFEval [45] for instruction-following fidelity, BBH [32] for complex reasoning across diverse tasks, GPQA [28] for graduate-level scientific understanding, MuSR [31] for multi-hop and long-context reasoning, and MMLU-Pro [40] for refined expert-knowledge assessment with reduced ambiguity. While many recent compression works continue to report only legacy benchmarks, we include these advanced evaluations to provide a more comprehensive view of capability preservation under sparsity. As shown in Table 10, our method maintains competitive performance across all tasks even at higher compression ratios, with graceful degradation that prioritizes reasoning-intensive benchmarks (e.g., MuSR) over surface-level accuracy.

Table 10: Evaluation on advanced benchmarks for Qwen3-8B and GeoPair at varying compression ratios (CR). Higher values indicate better performance.

Model	CR	IFEval	BBH	GPQA	MuSR	MMLU-Pro	Average
Qwen3-8B	–	39.4	55.7	36.7	43.3	46.3	44.3
GeoPair	0.2	32.6	49.5	32.0	46.4	36.0	39.3
	0.3	28.4	46.8	28.7	43.8	31.5	35.8
	0.4	25.9	37.8	25.8	44.1	25.2	31.8

D.4 Evaluations on Other Modalities

In the main paper, we demonstrated that our proposed method generalizes across diverse generative modalities, ranging from text to video. To further illustrate its versatility, we extend our evaluation to the audio generation domain. Specifically, we selected VibeVoice 1.5B [25] as a representative target model; smaller-scale models are typically more sensitive to compression, making them a challenging and informative test case.

For calibration, we used the first 256 samples from the dataset introduced in [26]. We then applied our compression method and evaluated the resulting model on a held-out subset of 100 samples drawn from a disjoint partition of the same dataset. Evaluation was conducted along two complementary dimensions: (i) *linguistic accuracy*, measured via Word Error Rate (WER) between the transcripts produced by Whisper-3 Large and the reference texts, after standard text normalization; and (ii) *perceptual quality*, assessed using UTMOS [29], an automatic predictor of mean opinion score.

As summarized in Table 11, the compressed model maintains strong performance at 20% compression, despite the absence of any fine-tuning or post-hoc recovery steps. While a modest increase in WER and a slight decrease in UTMOS are observed, the results confirm that core functionality is preserved. We anticipate that larger TTS architectures, such as VibeVoice 9B, would exhibit even greater compressibility due to higher redundancy in their parameter space. It is also worth noting that our attempt to apply Basis Sharing to VibeVoice resulted in a complete failure, with the compressed model yielding a WER over 100% and a UTMOS of around 1.4.

We present these findings primarily as a proof of concept, underscoring the applicability of our method across varying model scales, architectural designs, and generative modalities.

Table 11: Audio generation evaluation: Word Error Rate (WER) and UTMOS scores for the original and compressed VibeVoice 1.5B model at 20% compression.

Model	WER (↓)	UTMOS (↑)
VibeVoice 1.5B	6.6	4.17
Compressed (20%)	14.5	3.98

E Ablations

E.1 Convergence ablation

Table 12 ablates HTP and CG iteration counts at CR=0.4 on Llama 3.2 1B. With the Sylvester solver using adaptive convergence (termination at fifth-digit residual stabilization), 2 HTP + 10 CG iterations yields the optimal efficiency-accuracy balance: competitive perplexity and accuracy at 850s runtime. This configuration is used throughout our sparse compression experiments.

Table 12: Convergence ablation for HTP and CG iterations in the Sylvester refinement step (CR=0.4, Llama 3.2 1B). The Sylvester solver uses an adaptive convergence check (termination at 5th floating-point error stabilization). The configuration with 2 HTP and 10 CG iterations (first row) is selected as the default due to its optimal balance between runtime and performance. Lower perplexity and higher accuracy indicate better preservation of model capability.

Model	HTP Iters.	CG Iters.	Time (s)	CR	WikiText-2 (PPL↓)	Lambada (PPL↓)	Avg. Accuracy
Llama 3.2 1B	2	10	850	0.4	48.51	27.55	43.52
	5	10	1800	0.4	**46.16**	**25.76**	**43.69**
	2	20	1400	0.4	47.85	26.83	43.52
	2	5	680	0.4	48.51	28.68	43.27

E.2 Ablation on the Sparsification algorithm

To enforce the target sparsity level on the coefficient matrices, our framework supports multiple sparse recovery algorithms. In Table 13, we compare Hard Thresholding Pursuit (HTP) against Iterative Hard Thresholding (IHT). The results demonstrate that HTP achieves superior reconstruction fidelity with only two inner iterations, outperforming IHT configured with ten iterations while requiring approximately four times less computational time.

E.3 Ablation on Calibration Data Selection

To illustrate the sensitivity of our method to the choice of calibration data, we compare three distinct sources in Table 14: WikiText [21], Alpaca instructions [34], and fineweb [24] . All variants use 256 calibration samples at a 40% compression ratio. While Alpaca yields a slightly higher average accuracy (45.9% vs. 45.6%), fineweb achieves substantially better perplexity on the Lambada benchmark (21.69 vs. 38.86–39.02), indicating superior preservation of generative language modeling quality. This finding aligns with recent compression methods like CoSpaDi and ROCKET, which similarly

Table 13: Comparison of sparsification algorithms (HTP vs. IHT) at KS=2.5 on Llama3.2 1B. HTP achieves superior accuracy and perplexity with fewer iterations and significantly lower runtime. Lower perplexity and higher accuracy indicate better preservation of model capability. compression ratios are 20% , 40% respectively.

Model	Method	Iters.	Time (s)	CR	WikiText-2 (PPL↓)	Lambada (PPL↓)	Avg. Accuracy
Llama3.2 1B	Baseline	–	–	–	11.60	5.73	57.6
KS=2.5	HTP	2	850	0.2	**15.92**	**6.74**	**54.1**
	IHT	10	3473	0.2	17.45	7.95	52.7
KS=2.5	HTP	2	850	0.4	**34.26**	**21.69**	**45.6**
	IHT	10	2974	0.4	45.09	38.24	43.1

leverage large web corpora for calibration. Based on these results, particularly the lowest perplexity and consistency with established practices, we adopt fineweb as the default calibration dataset for all experiments.

Table 14: Ablation on calibration data selection at CR=40% on Llama3.2 1B. FineWeb achieves the lowest Lambada perplexity and competitive average accuracy, motivating its selection for main experiments. Lower perplexity values indicate better language modeling capability preservation; accuracy values are in %.

Model	Calibration Data	Samples	CR	WikiText-2 (PPL↓)	Lambada (PPL↓)	Avg. Accuracy
Llama3.2 1B	Baseline	–	–	11.60	5.73	57.6
GeoPair	WikiText-2	256	0.4	**30.16**	39.02	44.0
	Alpaca	256	0.4	53.23	38.86	**45.9**
	FineWeb	256	0.4	34.26	**21.69**	45.6

E.4 Ablation on Calibration Sequence Count

To examine the sensitivity of our method to the number of calibration samples, we evaluate GeoPair at CR=40% using FineWeb with sequence lengths ranging from 32 to 1024 samples. As shown in Table 15, performance generally improves as the calibration set grows, with notable gains up to 256 samples. Beyond this point, improvements become marginal: increasing from 256 to 1024 samples yields only a +0.9% gain in average accuracy and a modest reduction in Lambada perplexity, while requiring 4x calibration cost. Notably, the configuration with 256 samples already achieves strong preservation of both perplexity (21.69 on Lambada) and average accuracy (45.6%), closely matching the performance of larger calibration sets. This choice also aligns with established practices in recent compression methods such as CoSpaDi and ROCKET, which similarly adopt 256 calibration samples as an effective trade-off between efficiency and capability retention. Based on these observations, we fix the calibration size to 256 samples for all main experiments.

E.5 Ablation Study: Dictionary Initialization Strategies

The quality of the initial dictionary $\mathbf{D}_0$ plays a critical role in the convergence behavior and final reconstruction fidelity of alternating minimization schemes. In this section, we investigate the impact of three distinct dictionary initialization strategies on the performance of GeoPair, holding all other components constant (grouping via Blossom matching, Sylvester-based dictionary updates, and HTP sparsification with $K_S = 2.5$).

Initialization Methods We compare the following strategies for initializing the shared dictionary $\mathbf{D}_0 \in \mathbb{R}^{d \times r}$:

Proposed: Original-Space Concatenation + SVD (Ours). We concatenate the weight matrices of the paired layers in the original parameter space and compute the top-r left singular vectors via

Table 15: Ablation on calibration sequence length at CR=40% on Llama3.2 1B using fineweb. Higher average accuracy and lower perplexity indicate better capability preservation. We select 256 samples as the default, balancing performance and efficiency, consistent with CoSpaDi and ROCKET. Accuracy values are in %.

Model	CR	Data	Samples	WikiText-2 (PPL↓)	Lambada (PPL↓)	Avg. Accuracy
Llama3.2 1B	–	–	–	11.60	5.73	57.6
GeoPair	0.4	fineweb	32	44.40	41.90	42.1
			64	36.55	24.38	44.1
			128	35.40	22.76	45.1
			256	**34.26**	21.69	45.6
			512	34.26	22.34	45.3
			1024	34.17	**18.87**	**46.5**

truncated SVD:

$$\mathbf{D}_0 = \mathrm{SVD}_r\left([\mathbf{W}_1\ \mathbf{W}_2]\right). \tag{36}$$

This initialization preserves the intrinsic geometry of the pretrained weights before any calibration-induced transformation is applied. The individual layer-specific whitening transforms $\{\mathbf{L}_i\}$ are subsequently used during the alternating minimization stages.

Whitened-Space Concatenation + Global SVD. We first apply a *global* whitening transform $\mathbf{L}_{\text{global}}$, obtained by averaging the calibration Gram matrices of the paired layers, to the concatenated weights:

$$\widetilde{\mathbf{W}}_{\text{cat}} = \mathbf{L}_{\text{global}} \cdot [\mathbf{W}_1\ \mathbf{W}_2], \quad \mathbf{D}_0 = \mathbf{L}_{\text{global}}^{-1} \cdot \mathrm{SVD}_r\left(\widetilde{\mathbf{W}}_{\text{cat}}\right). \tag{37}$$

This strategy aligns with the covariance-averaging heuristic used in Basis Sharing [38].

Basis-Sharing Style Initialization. "We initialize using a hybrid protocol: concatenate weights after applying the individual whitening transforms, compute the shared basis via SVD, and then map the resulting dictionary back to the original space using the inverse of the global transform:

$$\mathbf{D}_0 = \mathbf{L}_{\text{global}}^{-1} \cdot \mathrm{SVD}_r\left([\mathbf{L}_1\mathbf{W}_1\ \mathbf{L}_2\mathbf{W}_2]\right). \tag{38}$$

This serves as a direct ablation of the initialization component, isolating its effect from the rest of the pipeline.

Experimental Setup We evaluate all initialization strategies on **Llama-3 1B** at a compression ratio of CR $= 0.4$, using the same calibration set, grouping pairs, and hyperparameters (r, KS, HTP iterations). No post-compression fine-tuning is applied. Results are reported across standard zero-shot benchmarks and perplexity metrics.

Table 16: Ablation of dictionary initialization methods on Llama-3 1B at CR=0.4. All methods use grouping, Sylvester-based dictionary updates, and HTP sparsification. Higher accuracy and lower perplexity indicate better preservation of model capability.

Method	PIQA	HellaSwag	Lambada_OA	ARC-e	ARC-c	SciQ	Race	MMLU	Lambada PPL↓	Avg. Acc.↑
Baseline (uncompressed)	74.53	63.66	62.95	60.47	36.20	88.30	37.79	37.00	5.73	57.61
Init: Whitened-Space + Global SVD	64.09	40.09	34.81	40.07	23.12	73.40	30.05	22.92	40.39	41.07
Init: Individual whiten then Global	63.82	39.29	35.57	40.70	24.91	74.20	30.62	23.12	39.66	41.53
Ours: Original-Space + SVD	**68.34**	**45.01**	**40.99**	**45.79**	**26.02**	**80.10**	**32.82**	**25.83**	**21.69**	**45.61**

Analysis. The results in Table 16 demonstrate that initializing the dictionary in the original parameter space yields marginally better average accuracy compared to whitened-space alternatives. While the differences appear modest at the aggregate level, we observe that original-space initialization provides more stable convergence during the alternating minimization phase, particularly for layers with highly divergent activation statistics.

E.6 Effect of Grouping on Relative Reconstruction Error

A natural question is whether the grouping under the proposed whitened-space surrogate metric translates to a lower final compression loss. Exact optimality for the true compression objective would require enumerating all valid pairings and running the full GeoPair optimization for each, which is combinatorially infeasible. Instead, we use the whitened submatrix alignment cost as a tractable structural-similarity proxy.

We compare grouping and dictionary-update strategies on Llama-3.2 1B at $\mathrm{CR} = 0.2$ using relative activation-weighted reconstruction error. For each compressed projection, we compute

$$e_i = \frac{\|L_i(W_i - \widehat{W}_i)\|_F}{\|L_i W_i\|_F}, \tag{39}$$

where L_i is the layer-specific Cholesky whitening factor and $\widehat{W}_i$ is the compressed weight matrix. For M compressed projections, the table reports the arithmetic mean

$$\overline{e} = \frac{1}{M} \sum_{i=1}^{M} e_i. \tag{40}$$

Each projection contributes equally to this average. The reported errors are unsquared Frobenius-norm ratios.

Table 17: Arithmetic mean of per-projection relative activation-weighted reconstruction errors for different grouping and dictionary-update strategies. Errors are unsquared Frobenius-norm ratios. Lower is better.

Grouping Strategy	Dictionary / Basis Update	Mean Relative Error ↓
Adjacent	Basis Sharing	0.161
Adjacent	Sylvester (ours)	0.141
Blossom (ours)	Sylvester (ours)	**0.129**
Blossom (ours)	Basis Sharing	0.183

Table 17 reveals two important effects:

1. **The surrogate is predictive for geometry-preserving updates:** Replacing adjacent grouping with Blossom grouping reduces the reconstruction error for our Sylvester update from 0.141 to 0.129. This confirms that the surrogate successfully identifies pairs that are genuinely more compatible to represent with a shared dictionary under distinct whitening spaces.
2. **The surrogate exposes the limitations of heuristic merging:** Applying Blossom grouping to Basis Sharing actually *increases* the reconstruction error from 0.161 to 0.183. This occurs because Basis Sharing relies on a global covariance-averaging assumption, which is violated when non-local pairs with distinct calibration geometries are forced into a single averaged whitening space.

With the Sylvester update, replacing adjacent pairing with the proposed matching reduces the mean relative reconstruction error from 0.141 to 0.129. For Basis Sharing, the corresponding error increases from 0.161 to 0.183. These aggregate comparisons support the benefit of the proposed grouping when combined with the geometry-preserving dictionary update.

E.7 Running Time and Environmental Impact

We tracked energy consumption and CO_2 emissions using CodeCarbon during compression. All experiments ran on a server with 256-core AMD EPYC 7742 CPU and 4× NVIDIA A100-SXM4-40GB GPUs.

Runtime scales approximately linearly with parameter count. Despite longer execution, larger models show better per-parameter energy efficiency (Qwen-32B: 0.099 kWh/B vs. Llama-1B: 0.147 kWh/B). Total emissions remain modest ($\leq$1.4 kg CO_2eq). All metrics recorded with CodeCarbon v3.2.6.

Table 18: Compression-only runtime and environmental metrics (CR=0.4, 256 calibration samples).

Model	Runtime (s)	Energy (kWh)	CO_2eq (kg)
Llama-3.2 1B	511.3	0.147	0.065
Llama-3 8B	3,010.0	1.523	0.672
Qwen-3 32B	19,905.9	3.165	1.396

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: main contributions of the paper are clearly stated in the Abstract and Introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The proposed algorithm works to group a pair of layers, for grouping more than two layers numerical approximations are required and it is outside the scope of this work.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provide a proper flow of the math for the proposed method and a follow-up detailed explanation is added to the appendix for the readers that are not familiar with this field.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Implementation details and summary of the algorithm are provided and we added the implementation of the method to the supplementary materials.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All data used on this work are publicly available and the implementation of the method is provided in the supplementary materials.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: No training is done in the scope of this paper, testing is done using open publicly available benchmarks and libraries.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: added to the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the running time for some experiments as well environmental impact study in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We believe that it does.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: no societal impact

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: the paper poses no such risks

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All experiments are done on publicly available models, models compressed and released using the proposed method inherits the license of the original model.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: the paper does not release new assets

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: the paper does not involve crowdsourcing

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

Justification: the paper does not involve crowdsourcing

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The LLM is used to polish the paper and correct grammar issues only.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.