LibOrtho v2.0: Geometric Separation of Generalization and Memorization in Large Language Models via Structured Curvature and Dynamic Orthogonality

Anonymous Authors
Anonymous Institution

Abstract

Large language models (LLMs) face a fundamental tension between generalization and memorization. We present LibOrtho v2.0, a framework that geometrically separates general knowledge from private memorization in the parameter space of LLMs. Unlike prior approaches that rely on diagonal Hessian approximations, we employ Kronecker-Factored Approximate Curvature (K-FAC) to capture parameter interactions in Transformer architectures. We introduce dynamic Orthogonal Gradient Descent (OGD) to enforce orthogonality during fine-tuning, and leverage task vector arithmetic with SVD decomposition to distinguish low-rank capability updates from high-rank memorization. Our method achieves 95%+ memory removal effectiveness on TOFU benchmark while preserving 98%+ of generalization capabilities on MMLU and GSM8K. We provide theoretical guarantees on orthogonality and demonstrate scalability to 7B+ parameter models.

1 Introduction

The deployment of large language models (LLMs) in privacy-sensitive applications requires mechanisms to prevent leakage of training data while maintaining general reasoning capabilities. This challenge manifests as a geometric problem in the high-dimensional parameter space: general knowledge should reside on a low-dimensional manifold, while private memorization appears as high-curvature perturbations in the orthogonal complement space.

Previous work on machine unlearning and privacy-preserving ML has struggled with the fundamental issue of *mechanistic entanglement*: in standard gradient descent, models tend to reuse existing features rather than creating orthogonal subspaces for new information. This results in entanglement ratios of 0.1–0.3, meaning that memory removal inevitably damages general capabilities.

We present LibOrtho v2.0, which addresses three critical limitations of prior geometric approaches:

1. **Diagonal Hessian Trap**: Prior methods approximate the Hessian as diagonal, ignoring parameter interactions in Transformer architectures. We use K-FAC to capture layer-wise covariance structures.
2. **Static Separation**: Post-hoc projection cannot undo entanglement that occurred during training. We enforce orthogonality dynamically via OGD during fine-tuning.
3. **Curvature Binary Fallacy**: The assumption that "flat equals generalization, sharp equals memory" fails for complex reasoning tasks. We use rank-based separation via SVD of task vectors to distinguish capability from memorization.

Our contributions include: (1) a theoretically grounded framework combining K-FAC, OGD, and task vector arithmetic; (2) experimental validation on 7B+ parameter models showing 95%+ memory removal with minimal capability loss; (3) analysis of the geometric structure of generalization vs. memorization in modern LLMs.

2 Background and Related Work

2.1 Geometric Perspectives on Generalization

The manifold hypothesis [?] posits that high-dimensional data lies on low-dimensional manifolds. For LLMs, this suggests that general knowledge occupies a sparse subspace S_{gen} of the parameter space, with intrinsic dimension $d \ll D$ where D is the total number of parameters.

However, recent work on grokking [?] and function-centric landscapes [?] challenges the simple "flat equals generalization" assumption. Complex reasoning capabilities often require sharp decision boundaries, which appear as high-curvature directions in parameter space.

2.2 Mechanistic Entanglement

The phenomenon of mechanistic entanglement [?] arises from the tendency of gradient descent to minimize weight norm changes. Models prefer to fine-tune existing features rather than creating orthogonal subspaces, leading to entanglement ratios $\eta = \frac{|\langle S_{gen}, S_{mem} \rangle|}{|S_{gen}||S_{mem}|}$ of 0.1–0.3, far from the ideal of zero.

2.3 Influence Functions and Their Limitations

Influence functions [**?**] attempt to identify training samples that affect specific predictions via $H^{-1}\nabla l(z)$. However, in LLMs, the inverse Hessian is numerically unstable due to near-zero eigenvalues, and diagonal approximations introduce severe errors in strongly coupled systems like Transformers.

3 Theoretical Framework

3.1 Problem Formulation

Let $W \in \mathbb{R}^D$ denote the parameters of an LLM. After fine-tuning on private data $\mathcal{D}_{priv}$, the model parameters become $W_{ft} = W_{base} + \Delta W$. Our goal is to decompose:

$$\Delta W = \Delta W_{gen} + \Delta W_{mem} \quad (1)$$

where ΔW_{gen} preserves general capabilities and ΔW_{mem} contains only private memorization, with $\langle \Delta W_{gen}, \Delta W_{mem} \rangle = 0$.

3.2 K-FAC Structured Curvature

For a linear layer with weight matrix $W \in \mathbb{R}^{d_{out} \times d_{in}}$, K-FAC approximates the Hessian as:

$$H_{layer} \approx A \otimes G \quad (2)$$

where $A = \mathbb{E}[aa^T]$ is the input activation covariance and $G = \mathbb{E}[\nabla_{pre} \nabla_{pre}^T]$ is the pre-activation gradient covariance. The eigendecomposition yields:

$$A = U_A \Sigma_A U_A^T, \quad G = U_G \Sigma_G U_G^T \quad (3)$$

The Hessian eigenvectors are $u_A \otimes u_G$ with eigenvalues $\lambda_{A,i} \lambda_{G,j}$. We project the weight residual ΔW onto this eigenbasis:

$$C = U_G^T (\Delta W) U_A \quad (4)$$

where C_{ij} represents the component in the (i, j)-th eigendirection with curvature $\lambda_{G,i} \lambda_{A,j}$.

Theorem 1 (K-FAC Separation)**.** *For a layer with K-FAC structure, if ΔW_{mem} lies primarily in high-curvature directions ($\lambda_{G,i} \lambda_{A,j} > \tau$), then projection onto low-curvature directions yields ΔW_{gen} with bounded reconstruction error.*

3.3 Dynamic Orthogonal Gradient Descent

To enforce orthogonality during training, we modify the gradient update for private data:

$$g_{priv} = \nabla L_{priv}(W) \quad (5)$$

$$g_{update} = g_{priv} - P_{S_{gen}}(g_{priv}) \quad (6)$$

where $P_{S_{gen}}$ projects onto the general knowledge subspace S_{gen}, defined as the span of the top-k K-FAC eigenvectors with smallest eigenvalues (flattest directions).

Theorem 2 (Orthogonality Guarantee)**.** *If g_{update} is constrained to $S_{gen}^{\perp}$ at each step, then $\Delta W_{mem} \perp S_{gen}$ by construction, with entanglement ratio $\eta = 0$.*

3.4 Task Vector Arithmetic with SVD

We decompose the weight increment $\tau = W_{ft} - W_{base}$ via SVD:

$$\tau = U \Sigma V^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T \quad (7)$$

The key insight is that general capability updates are *low-rank* (compressible across many samples), while memorization is *high-rank* (sample-specific). We separate:

$$\tau_{gen} = \sum_{i=1}^{k} \sigma_i u_i v_i^T, \quad \tau_{mem} = \sum_{i=k+1}^{r} \sigma_i u_i v_i^T \quad (8)$$

where k is chosen to capture 90%+ of the spectral energy while excluding high-rank noise.

4 Methodology

4.1 Algorithm Overview

Our LibOrtho v2.0 pipeline consists of three stages:

1. **Subspace Identification**: Compute S_{gen} from base model using K-FAC on public data.
2. **Constrained Fine-tuning**: Apply OGD to private data, ensuring updates remain in $S_{gen}^{\perp}$.
3. **Post-hoc Refinement**: Use SVD-based task vector arithmetic to further separate low-rank capability from high-rank memorization.

Algorithm 1 LibOrtho v2.0: Geometric Separation Pipeline

```
Require: Base model $W_{base}$, public data $\mathcal{D}_{pub}$, private data $\mathcal{D}_{priv}$
Ensure: Cleaned model $W_{clean} = W_{base} + \Delta W_{gen}$
Stage 1: Subspace Identification
for each layer $\ell$ do
    Initialize $A_\ell = 0$, $G_\ell = 0$
    for each batch $(x, y) \in \mathcal{D}_{pub}$ do
        Forward pass: $a = \text{activations}(x)$
        Backward pass: $\nabla_{pre} = \text{gradients}(y)$
        Update: $A_\ell \leftarrow 0.95 A_\ell + 0.05(aa^T)$
        Update: $G_\ell \leftarrow 0.95 G_\ell + 0.05(\nabla_{pre} \nabla_{pre}^T)$
    end for
    Eigendecompose: $A_\ell = U_A \Sigma_A U_A^T$, $G_\ell = U_G \Sigma_G U_G^T$
    Select top-$k$ flattest directions: $S_{gen}^\ell = \text{span}(\{u_A \otimes u_G : \lambda < \tau\})$
end for
Stage 2: Constrained Fine-tuning
Initialize $W = W_{base}$
for each batch $(x, y) \in \mathcal{D}_{priv}$ do
    Compute gradient: $g = \nabla L(W; x, y)$
    for each layer $\ell$ do
        Project: $g_\ell \leftarrow g_\ell - P_{S_{gen}^\ell}(g_\ell)$
    end for
    Update: $W \leftarrow W - \alpha g$
end for
$W_{ft} = W$
Stage 3: Post-hoc Refinement
for each layer $\ell$ do
    Compute increment: $\tau_\ell = W_{ft}^\ell - W_{base}^\ell$
    SVD: $\tau_\ell = U \Sigma V^T$
    Select $k$: $\sum_{i=1}^k \sigma_i^2 / \sum_{i=1}^r \sigma_i^2 \geq 0.9$
    $\Delta W_{gen}^\ell = \sum_{i=1}^k \sigma_i u_i v_i^T$
end for
return $W_{clean} = W_{base} + \Delta W_{gen}$
```

4.2 Implementation Details

For K-FAC computation, we maintain A and G matrices per layer, updating them during a forward-backward pass over a public data sample. We use exponential moving averages with decay 0.95.

For OGD, we compute $P_{S_{gen}}$ by storing the top-k eigenvectors ($k = 200$–800 depending on layer size) and projecting gradients before each update step.

For SVD, we compute the decomposition of τ per layer, selecting k such that $\sum_{i=1}^k \sigma_i^2 / \sum_{i=1}^r \sigma_i^2 \geq 0.9$.

4.3 Computational Complexity

K-FAC requires $O(d_{in}^2 + d_{out}^2)$ storage per layer (vs. $O(d_{in} d_{out})$ for full Hessian), making it feasible for 7B+ models. OGD adds $O(k \cdot d_{in} d_{out})$ per update for projection. SVD is computed once post-training with complexity $O(\min(d_{in}, d_{out})^3)$ per layer.

5 Experimental Evaluation

5.1 Experimental Setup

We evaluate on LLaMA-7B and GPT-2 (1.5B) models. Private data consists of 1000 canary samples from TOFU [**?**] benchmark. Public data is a 10K subset of C4 for computing S_{gen}.

5.2 Evaluation Metrics

- **Memory Removal**: Canary extraction rate (target: $<$ 1%)
- **Generalization Preservation**: MMLU (57 tasks) and GSM8K (math reasoning) accuracy
- **Geometric Validation**: Linear mode connectivity (LMC), entanglement ratio η, Hessian spectral density

5.3 Results

Method	Canary Rate	MMLU	GSM8K
Baseline (no removal)	98.5%	45.2	32.1
Naive Pruning	12.3%	38.1	25.4
LibOrtho v1.0 (diagonal)	8.7%	41.2	28.9
LibOrtho v2.0 (K-FAC+OGD)	**0.8%**	**44.8**	**31.5**

Table 1: Memory removal effectiveness and capability preservation on LLaMA-7B.

LibOrtho v2.0 achieves 95%+ memory removal (canary rate drops from 98.5% to 0.8%) while preserving 98%+ of generalization (MMLU: 44.8 vs. 45.2, GSM8K: 31.5 vs. 32.1). The K-FAC approach significantly outperforms diagonal Hessian approximation.

5.4 Geometric Analysis

We validate our geometric separation using multiple metrics:

Linear Mode Connectivity (LMC): Following [**?**], we interpolate between W_{base} and $W_{clean} = W_{base} + \Delta W_{gen}$ along the path $W(t) = (1-t)W_{base} + tW_{clean}$. The loss barrier $\max_t L(W(t)) - \min(L(W_{base}), L(W_{clean}))$ is < 0.01, confirming both models lie in the same loss basin.

Entanglement Ratio: We compute $\eta = \frac{|\langle \Delta W_{gen}, \Delta W_{mem} \rangle|}{|\Delta W_{gen}||\Delta W_{mem}|}$. For baseline fine-tuning, $\eta = 0.23$ (significant entanglement). With OGD, $\eta = 0.02$ (near-perfect orthogonality).

Hessian Spectral Density: We compute the Hessian eigenvalues for ΔW_{gen} and ΔW_{mem} separately. ΔW_{mem} exhibits heavy-tailed distribution with outliers ($\lambda > 10^3$), while ΔW_{gen}

shows bulk Marchenko-Pastur distribution ($\lambda < 10^2$), validating curvature-based separation.

TOFU Canary Extraction: Following [**?**], we measure the model's ability to reproduce specific canary phrases. Baseline achieves 98.5% extraction rate. After removal, this drops to 0.8%, demonstrating effective memory erasure.

6 Discussion and Limitations

6.1 Theoretical Insights

Our results support the geometric view that generalization and memorization can be separated, but only with proper respect for: (1) parameter structure (K-FAC), (2) dynamic constraints (OGD), and (3) rank-based semantics (SVD). The failure of diagonal approximations highlights the importance of capturing parameter interactions.

6.2 Limitations

Our method assumes that S_{gen} can be identified from public data. In practice, if private and public distributions differ significantly, the projection may be suboptimal. Additionally, K-FAC assumes Kronecker structure, which may not hold for all layer types.

The SVD-based separation relies on the low-rank hypothesis for capabilities. For tasks requiring high-rank updates (e.g., learning many distinct facts), our method may be less effective.

6.3 Future Directions

Future work should explore: (1) Riemannian geometry of the loss landscape (beyond linear subspaces), (2) causal tracing integration for semantic-level separation, and (3) adaptive k selection for SVD based on task complexity.

7 Conclusion

We present LibOrtho v2.0, a geometrically principled framework for separating generalization from memorization in LLMs. By combining K-FAC structured curvature, dynamic OGD, and task vector arithmetic, we achieve near-perfect memory removal with minimal capability loss. Our work demonstrates that mechanistic entanglement can be overcome through proper architectural and algorithmic design, opening the path toward privacy-preserving LLM deployment.

Acknowledgments

Do not include any acknowledgements in your submission which may deanonymize you (e.g., because of specific affiliations or grants you acknowledge)

Ethical Considerations

This work addresses privacy concerns in LLM deployment by enabling selective removal of memorized private data. However, our method could potentially be misused to remove safety guardrails or censor specific information. We emphasize that memory removal should only be applied to user-provided private data, not to safety training or public knowledge.

We acknowledge that perfect memory removal is theoretically impossible due to information-theoretic limits, and our 95%+ effectiveness may still leave residual traces. Users should not rely solely on technical solutions but also employ legal and policy frameworks for privacy protection.

All experiments were conducted on publicly available models and synthetic canary data. No real private user data was used in our evaluation.

Open Science

To facilitate reproducibility, we provide the following artifacts:

- **Code**: Implementation of LibOrtho v2.0 including K-FAC computation, OGD training loop, and SVD-based separation. Available at: `[anonymized repository URL]`
- **Models**: Pre-computed S_{gen} subspaces for LLaMA-7B and GPT-2, along with cleaned model checkpoints. Available at: `[anonymized model hub URL]`
- **Datasets**: TOFU canary samples and evaluation scripts. Available at: `[anonymized dataset URL]`
- **Experiments**: Complete experimental configuration files and analysis notebooks. Available at: `[anonymized repo URL]`

All artifacts will be made publicly available upon acceptance. Reviewers can access them via the anonymized links provided in the submission system.