SOMP: Scalable Gradient Inversion for Large Language Models via Subspace-Guided Orthogonal Matching Pursuit

Anonymous ACL submission

Abstract

Gradient inversion attacks reveal that private training text can be reconstructed from shared gradients, posing a privacy risk to large language models (LLMs). While prior methods perform well in small-batch settings, scaling to larger batch sizes and longer sequences remains challenging due to severe signal mixing, high computational cost, and degraded fidelity. We present **SOMP** (Subspace-Guided Orthogonal Matching Pursuit), a scalable gradient inversion framework that casts text recovery from aggregated gradients as a sparse signal recovery problem. Our key insight is that aggregated transformer gradients retain exploitable head-wise geometric structure together with sample-level sparsity. SOMP leverages these properties to progressively narrow the search space and disentangle mixed signals without exhaustive search. Experiments across multiple LLM families, model scales, and five languages show that SOMP consistently outperforms prior methods in the aggregated-gradient regime. For long sequences at batch size $B = 16$, SOMP achieves substantially higher reconstruction fidelity than strong baselines, while remaining computationally competitive. Even under extreme aggregation (up to $B = 128$), SOMP still recovers meaningful text, showing that large-batch aggregation alone does not eliminate gradient leakage.

1 Introduction

The reliance of LLMs on massive, often sensitive datasets raises critical privacy concerns, particularly in collaborative training scenarios across multiple institutions. Federated learning (FL) is often used in privacy-sensitive text training, where raw data cannot be centrally shared. FL mitigates this concern by sharing aggregated gradient updates instead of raw text, but aggregation alone does not provide a formal privacy guarantee (Li et al., 2024).

Early gradient inversion attacks showed that an honest-but-curious server can reconstruct private training data directly from shared gradients (Zhu et al., 2019). Although these attacks were first studied in continuous image domains (Li et al., 2022), they now pose a direct threat to LLMs, where discrete textual inputs can also be recovered. In particular, methods such as *DAGER* demonstrate that entire text batches can be exactly reconstructed from aggregated transformer updates (Petrov et al., 2024). These results show that gradient aggregation alone can still leak information about individual samples.

However, existing text-based gradient inversion methods face serious scalability challenges. Early approaches rely heavily on external language models and are therefore limited to short sequences or lightly aggregated settings. More recent methods extend to larger setups, but introduce new bottlenecks: hybrid optimization methods such as GRAB (Feng et al., 2024) suffer from cross-sample interference, while exact recovery methods such as DAGER (Petrov et al., 2024) incur prohibitive costs under longer sequences and large batch size. As a result, prior methods become brittle precisely in the long-sequence, multi-sample regimes that are most relevant in practice. [1]

To address this gap, we propose **SOMP**, a structured framework for multi-sample text gradient inversion under aggregated gradients. We study the standard honest-but-curious server setting, where the attacker observes an aggregated gradient update. Our key insight is that aggregated transformer gradients are not arbitrary mixtures: they retain useful head-wise geometric structure induced by multi-head attention, together with sample-level sparsity. Building on this observation, SOMP reformulates inversion as a structured reconstruction problem in gradient space, enabling more effective disentanglement without exhaustive token-level verifi-

[1] Anonymous code repository: https://anonymous.4open.science/r/GIA_somp-DCB3

cation. Our goal is not to claim uniform gains in every regime, but to improve reconstruction quality and scaling behavior precisely where prior attacks degrade most: long-sequence, large batch size settings.

- **Head-aware geometric guidance for text gradient inversion.** We show that aggregated transformer gradients retain useful head-wise geometric structure, and exploit this structure to guide candidate filtering and decoding under mixed gradients.
- **A sparse reconstruction formulation for multi-sample inversion.** We reformulate exact recovery from exhaustive token-level search as sparse signal recovery in gradient space, substantially mitigating the scalability bottlenecks of prior methods and enabling effective disentanglement of cross-sample interference.
- **Robust and scalable inversion under gradient aggregation.** Across multiple transformer families and aggregated-gradient regimes, SOMP yields stronger reconstruction quality than prior text-based attacks, with its largest gains appearing in long-sequence and larger-batch settings.

2 Related Work

Gradient inversion and data leakage attacks. The problem of reconstructing private data from shared gradients, often termed *gradient inversion* or *gradient leakage*, was first explored by Zhu et al. (2019) and has since drawn extensive attention in federated learning. Early attacks mainly targeted images and formulated reconstruction as continuous optimization (Zhu et al., 2019; Geiping et al., 2020; Huang et al., 2021; Chen and Liu, 2025). More recent works such as **SPEAR** (Dimitrov et al., 2024) exploit low-rank and sparsity structures to enable exact reconstruction of large image batches. However, these methods are designed for continuous inputs or fully connected architectures and cannot be directly transferred to text, where inputs are discrete and tokenization introduces non-differentiable constraints.

Gradient inversion for textual data. Reconstructing text from gradients is harder due to the discrete token space and the combinatorial structure of language. Early text attacks such as **TAG** (Deng et al., 2021) show that transformer gradients can leak token information, while **LAMP** (Balunović et al., 2022) and **GRAB** (Feng et al., 2024) use language priors or hybrid continuous–discrete optimization, but scale poorly under aggregation or long sequences. **DAGER** (Petrov et al., 2024) exploits the low-rank structure of self-attention gradients for exact text reconstruction, but its layer-wise exhaustive search becomes costly and unstable in high-token regimes. **Decepticons** (Fowl et al., 2023) also demonstrates text recovery from transformer gradients in federated learning, but it studies a stronger malicious-server setting where the server can modify the model to amplify leakage. SOMP differs from these works by following the honest-but-curious setting: the attacker only observes aggregated gradients, and targets long-sequence, large-batch reconstruction through head-wise geometric filtering and sparse gradient-space reconstruction.

3 Background

In this section, we introduce the problem setup and summarize the structural properties of gradients in transformer-based models that serve as fundamentals of our method.

3.1 Problem Setup

We consider LLMs parameterized by $\boldsymbol{\theta}$ and trained under the *FedSGD* protocol. Each client holds a local mini-batch of B training samples $\{(x_j, y_j)\}_{j=1}^{B}$ and transmits only the aggregated gradient to the server. Given a loss function $\mathcal{L}$, the server observes

$$\boldsymbol{g}_{\text{mix}} = \frac{1}{B}\sum_{j=1}^{B} \boldsymbol{\nabla}_{\boldsymbol{\theta}} \boldsymbol{\mathcal{L}}\big(f_\theta(x_j), y_j\big), \qquad (1)$$

without access to individual samples or per-sample gradients.

We consider a transformer with context length p, hidden dimension d, and H attention heads. Given an input sequence $x = [x_1, \ldots, x_p]$, the corresponding token embeddings form the input representation $\boldsymbol{Z}_0 \in \mathbb{R}^{p\times d}$.

3.2 Structural Properties

Gradients in transformer-based models exhibit several structural properties that are useful for multi-sample inversion. First, as noted in prior work (Petrov et al., 2024), gradients of linear layers inherit a matrix-product form. For a linear layer $Y = XW + (b|\cdots|b)^\top$, the weight gradient satisfies

$$\boldsymbol{\nabla}_{\boldsymbol{W}} \boldsymbol{\mathcal{L}} = X^\top \nabla_Y \mathcal{L}, \qquad (2)$$

which induces a low-rank structure in the resulting gradient matrix.

Second, multi-head attention induces a natural head-wise decomposition in gradient space. Let $(\cdot)_l$ denote the l-th transformer layer, $(\cdot)_{Q,K,V}$ denote the query, key, and value projections, and $(\cdot)^{(h)}$ index the h-th attention head. In particular, $\boldsymbol{W}_{1,Q}^{(h)}$ denotes the query projection matrix of head h in the first transformer layer. For the first-layer query projection, the gradient of head h can be written as

$$\nabla \boldsymbol{W}_{1,Q}^{(h)} = \boldsymbol{Z}_{\mathbf{0}}^{\top} \boldsymbol{G}_{Q}^{(h)} \in \mathbb{R}^{d \times d_h}, \tag{3}$$

where $\boldsymbol{G}_{Q}^{(h)} \in \mathbb{R}^{p \times d_h}$ denotes the backpropagated signal for head h, and $d = \sum_{h=1}^{H} d_h$. The full query gradient is obtained by concatenating these head-wise slices.

Finally, these gradients exhibit informative sparsity patterns at the token level (Dimitrov et al., 2024). In particular, head-aligned sparsity provides useful cues for filtering and scoring candidate tokens under aggregated gradients. A formal discussion is provided in Appendix A.

Together, these properties suggest that aggregated gradients retain useful structure that can be exploited for multi-sample reconstruction.

4 Methodology

We now proceed to introduce the proposed **SOMP**, a three-stage framework for reconstructing textual inputs from the aggregated gradient $\boldsymbol{g}_{\text{mix}}$. Unlike prior search-based methods, SOMP treats multi-sample inversion as a structured reconstruction problem and progressively narrows the search space from tokens to sentences and finally to sample gradients. Stage I uses head-wise gradient slices to construct a compact token pool. Stage II decodes candidate sentences from this pool using geometry-guided beam search with an LM prior. Stage III then treats the decoded candidates as gradient atoms and selects a sparse subset whose gradients best reconstruct $\boldsymbol{g}_{\text{mix}}$. This progressive design separates coarse candidate filtering from final sample selection, making reconstruction more scalable under long sequences and large-batch aggregation. The overall pipeline is illustrated in Figure 1.

Unless otherwise specified, hyperparameters are fixed across experiments. A full list of hyperparameters is provided in Appendix C.

4.1 Stage I: Head-Structured Token Pooling

Following Eq. (3), we decompose the first-layer query gradient into head-specific slices $\{\nabla \boldsymbol{W}_{1,Q}^{(h)}\}_{h=1}^{H}$. Each slice induces a head-specific subspace, which we use to evaluate token plausibility under the observed aggregated gradient. The goal of Stage I is not to reconstruct complete sequences, but to build a compact, high-recall token pool that constrains the search space in Stage II.

To rank candidate tokens, we combine three complementary signals: geometric alignment to the head-wise query subspaces, consistency across informative heads, and head-aware sparsity patterns in the FFN gradient.

Detailed pseudocode is given in Appendix D, and token-pool recall is analyzed in Appendix C.

Subspace-alignment score. For a candidate embedding $\boldsymbol{e}(v, pos) \in \mathbb{R}^d$, we first project it into the query space of head h:

$$\boldsymbol{q}^{(h)}(v, pos) = \boldsymbol{e}(v, pos) \boldsymbol{W}_{1,Q}^{(h)} \in \mathbb{R}^{d_h}.$$

The subspace projector $\boldsymbol{P}_{\mathcal{R}_Q^{(h)}}$ is constructed from $\nabla \boldsymbol{W}_{1,Q}^{(h)}$ by truncated SVD. For $\nabla \boldsymbol{W}_{1,Q}^{(h)} = U \Sigma V^{\top}$, we set $\boldsymbol{P}_{\mathcal{R}_Q^{(h)}} = V_{1:r_h} V_{1:r_h}^{\top}$, where $r_h = \min(r_{\max}, |\{i : \sigma_i / \sigma_1 \geq \tau_{\text{svd}}\}|)$. We define the head-wise alignment score as

$$s_{\text{sub}}^{(h)}(v, pos) = \left\| (\boldsymbol{I} - \boldsymbol{P}_{\mathcal{R}_Q^{(h)}}) \boldsymbol{q}^{(h)}(v, pos) \right\|_2, \tag{4}$$

where smaller values indicate stronger geometric compatibility. We then aggregate these scores across the informative heads $\mathcal{H}_{\text{act}}$:

$$s_{\text{sub}}(v, pos) = \frac{1}{|\mathcal{H}_{\text{act}}|} \sum_{h \in \mathcal{H}_{\text{act}}} s_{\text{sub}}^{(h)}(v, pos). \tag{5}$$

Cross-head consistency score. Plausible tokens should align consistently across informative heads rather than match only a single head by chance (Li et al., 2019; Voita et al., 2019). We therefore compute

$$s_{\text{cons}}(v, pos) = \text{Std}_{h \in \mathcal{H}_{\text{act}}} \big(s_{\text{sub}}^{(h)}(v, pos) \big), \tag{6}$$

where smaller values indicate more stable cross-head alignment.

Head-aware sparsity score. In addition to geometric alignment, we exploit token-level sparsity signals using head-aligned FFN blocks. For each block h, we compute

$$\boldsymbol{u}^{(h)}(v, pos) = \boldsymbol{e}(v, pos)^{\top} \boldsymbol{W}_{\text{FFN}}^{(h)} \in \mathbb{R}^{d_h}, \tag{7}$$

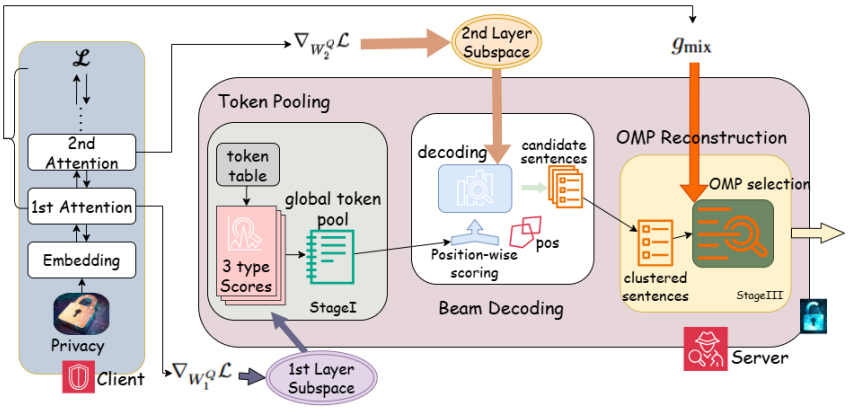


Figure 1: Overview of SOMP. It consists of Stage I head-structured token pooling, Stage II geometry-driven diverse decoding, and Stage III gradient-space sparse reconstruction.

where $\boldsymbol{W}_{\text{FFN}}^{(h)} \in \mathbb{R}^{d\times d_h}$ denotes the FFN gradient slice associated with block h. The corresponding block-wise sparsity score is

$$s^{(h)}(v, pos) = \frac{1}{d_h}\sum_{m=1}^{d_h} \mathbf{1}\left(|u_m^{(h)}(v, pos)| \le \tau_m^{(h)}\right), \tag{8}$$

where $\tau_m^{(h)}$ is a per-dimension threshold estimated from the candidate set. Let $\mathcal{H}_{\text{top}}$ denote the top-k blocks ranked by sparsity. We aggregate them as

$$s_{\text{sparse}}(v, pos) = \frac{1}{k}\sum_{h\in\mathcal{H}_{\text{top}}} s^{(h)}(v, pos). \tag{9}$$

Token ranking. We then combine the three scores into a single cost:

$$s_{\text{total}}(v, pos) = \lambda_1 s_{\text{sub}}(v, pos) + \lambda_2 s_{\text{cons}}(v, pos) - \lambda_3 s_{\text{sparse}}(v, pos). \tag{10}$$

where lower values indicate higher token plausibility. Tokens are ranked by minimizing s_{total}, and the top candidates are retained to form the token pool used in Stage II.

4.2 Stage II: Geometry-Driven Diverse Beam Decoding

Stage II generates candidate sentences from the token pool produced by Stage I. To guide decoding, we construct head-wise query-gradient subspaces $\{\mathcal{R}_{2,Q}^{(h)}\}$ from the second transformer layer, following prior observations that lower and middle layers provide more stable geometric directions for inversion. See Algorithm 2 for the full decoding procedure.

At decoding step t, let $h_t^{(h)}$ denote the head-wise hidden representation for head h. We measure its mismatch to the recovered geometry by

$$d_t^{(h)} = \left\|(\boldsymbol{I} - \boldsymbol{P}_{\mathcal{R}_{2,Q}^{(h)}})h_t^{(h)}\right\|_2.$$

We then aggregate these distances across informative heads:

$$d_t^{\text{geo}} = \frac{1}{|\mathcal{H}_{\text{act}}|}\sum_{h\in\mathcal{H}_{\text{act}}} d_t^{(h)}.$$

To improve fluency, we combine the geometric score with an LM prior (Devlin et al., 2019; Dathathri et al., 2020). Let $\tilde{h}_t$ denote the final-layer hidden state and let v_{x_t} denote the embedding of candidate token x_t. We define

$$s_{\text{LM}} = \langle \tilde{h}_t, v_{x_t}\rangle, \tag{11}$$

and combine it with geometry as

$$d_t = d_t^{\text{geo}} - \beta_{\text{LM}}\hat{s}_{\text{LM}}, \tag{12}$$

where $\hat{s}_{\text{LM}}$ denotes the standardized fluency score.

Beam scores are accumulated across decoding steps:

$$\text{sum}_t = \text{sum}_{t-1} + d_t. \tag{13}$$

The base beam score is then defined as

$$\text{score}_t^{\text{base}} = \begin{cases} \text{sum}_t/t, & \text{with length normalization,} \\ \text{sum}_t, & \text{otherwise.} \end{cases} \tag{14}$$

To encourage diversity, we add repetition-aware penalties:

$$\text{score}_t = \text{score}_t^{\text{base}} + \lambda_{\text{div}}\mathbf{1}(x_t \in \mathcal{S}_t) + \lambda_{\text{ng}}\mathbf{1}(g_t \in \mathcal{G}_t), \tag{15}$$

where $\mathcal{S}_t$ is the set of previously selected tokens, $\mathcal{G}_t$ is the set of observed n-grams, and g_t denotes the newly formed n-gram at step t. We retain the top-W/G hypotheses per group, yielding a candidate set $\mathcal{X}$ for the final reconstruction stage.

4.3 Stage III: Gradient-Space Sparse Reconstruction

Given the candidate sentences $\mathcal{X} = \{x^{(1)}, \ldots, x^{(M)}\}$ from Stage II, we compute a gradient atom for each candidate:

$$\boldsymbol{g}_i := \boldsymbol{\nabla}_{\boldsymbol{\theta}} \boldsymbol{\mathcal{L}}(f_\theta(x_i), y_i).$$

While this definition depends on the candidate label y_i, the true labels are unavailable at reconstruction time. We therefore compute candidate gradients using a fixed surrogate label. Empirically, Stage-III selection remains stable across label choices in the large-batch IMDB setting; see Appendix B.

To reduce redundancy, we cluster beam outputs by ROUGE-L similarity and retain one representative per cluster.

We then model the observed gradient as a sparse combination over the candidate set:

$$\boldsymbol{g}_{\text{mix}} \approx \sum_{i \in \mathcal{S}} \alpha_i \boldsymbol{g}_i, \tag{16}$$

where $\mathcal{S} \subseteq \{1, \ldots, M\}$ is an unknown support set with $|\mathcal{S}| = B$. Intuitively, although Stage II may generate many plausible candidates, only a small subset should contribute to the true aggregated gradient. We recover this support using Orthogonal Matching Pursuit (OMP).

At iteration t, given the current residual $\boldsymbol{r}_t$, OMP selects the most aligned candidate:

$$i_t = \arg\max_i \frac{|\langle \boldsymbol{g}_i, \boldsymbol{r}_t \rangle|}{\|\boldsymbol{g}_i\|_2}. \tag{17}$$

The active coefficients are then updated by ridge-regularized least squares:

$$\alpha_{\mathcal{S}} = \arg\min_{\alpha} \left\| \boldsymbol{g}_{\text{mix}} - \sum_{j \in \mathcal{S}} \alpha_j \boldsymbol{g}_j \right\|_2^2 + \lambda \|\alpha\|_2^2. \tag{18}$$

The residual is updated as

$$\boldsymbol{r}_{t+1} = \boldsymbol{g}_{\text{mix}} - \sum_{j \in \mathcal{S}} \alpha_j \boldsymbol{g}_j. \tag{19}$$

The procedure terminates when $\|\boldsymbol{r}_t\|_2 < \varepsilon$ or $|\mathcal{S}| = B$. The resulting support identifies the candidate sentences whose gradients best explain the observed mixture. The full algorithm is provided in Algorithm 3.

4.4 Summary

SOMP combines geometric candidate filtering with sparse residual fitting. In Stages I and II, head-wise structure is used as a soft constraint to prune implausible hypotheses while preserving a high-recall candidate set. Stage III then resolves the remaining ambiguity by selecting a sparse subset of candidate gradients that best reconstructs $\boldsymbol{g}_{\text{mix}}$. Overall, SOMP is best understood as a structured reconstruction framework rather than a method with worst-case recovery guarantees.

5 Experimental Evaluation

We evaluate SOMP across multiple models, datasets, and training protocols along five complementary dimensions. First, we compare against prior text-based inversion methods under aggregated gradients. Second, we vary batch size to test whether SOMP remains effective as sample mixing becomes stronger. Third, we extend the evaluation beyond standard English FedSGD, including multilingual and FedAvg-style settings. Fourth, we evaluate SOMP under DP-SGD-style clipping and Gaussian noise to study attack-side recoverability when explicit perturbation defenses are applied. Finally, we perform ablation studies to identify which stages of the pipeline are most responsible for the gains.

5.1 Experimental Setup.

We study the standard honest-but-curious server setting under FedSGD. The attacker observes only the aggregated gradient $\boldsymbol{g}_{\text{mix}}$ computed over a batch of size B, and does not have access to individual examples or per-sample gradients. Unless otherwise stated, gradients are extracted from the first-layer query projections.

Models and datasets. We evaluate **SOMP** on three autoregressive LLMs: `GPT-2` (Radford et al., 2019), `GPT-J-6B` (Wang and Komatsuzaki, 2021), and `Qwen3-8B` (Yang et al., 2025). Reconstruction experiments are conducted on `CoLA` (Warstadt et al., 2019), `SST-2` (Socher et al., 2013), and `stanfordNLP/imdb` (Maas et al., 2011). Inputs are truncated to 512 tokens.

Baselines and metrics. We compare against representative prior text-based inversion methods: DAGER, LAMP, and GRAB. All baselines are reimplemented under the same aggregation protocol and evaluation pipeline. Reconstruction qual-

B	Reference Inputs	DAGER Reconstruction	SOMP Reconstruction
4	If JognThaw had never ... is absolute must include.	If JognThaw had never ... is absolute must include.	If JognThaw had never ... is absolute must include.
	This sequel is quite ... lack of choreography.	This sequel is quite ... movie is lack of money .	This sequel is quite ... lack of choreography. .
	I saw this movie when ... in another movie.	I saw this movie when ... in another movie.	I saw this movie when ... in another movie.
	I won't ask you ... give her any money.	I saw this movie when ... in another movie .	I won't ask you ... give her any money.
8	Jill Dunne (played by Mitzi Kapture), is an ... 3 stars.	Jill Dunne (played by Mitzi Kapture),is an ... nothing happens!	Jill Dunne (played by Mitzi Kapture), is an ... Jill. 3 stars.
	...	...	...
	If the term itself were not geographically ... different accents.	If the term itself were not ... Ned Kelly geographic Ned Kelly	If the term itself were not geographically ... different accents.
	SWING! It's an important film because ... it's really, really bad!	SWING! It's an important film because ... it's real real real real	SWING! It's an important film because ...it's really bad!!!

Table 1: Reconstruction examples under aggregated gradients at $B = 2, 4, 8$. Highlighted spans mark incorrect tokens. As B increases, DAGER degrades faster, while SOMP remains more faithful to the reference inputs.

ity is measured using sequence-level **ROUGE-1**, **ROUGE-2**, and **ROUGE-L** (Lin, 2004), averaged over 50 random batches and reported as mean $\pm$ standard deviation. Since ROUGE mainly captures lexical fidelity rather than entity-level leakage, we additionally report named-entity exact-match recall in Appendix E.1. For readability, the main text reports **ROUGE-L**; full ROUGE-1/2/L results are in Appendix E.

5.2 Reconstruction Performance and Comparisons

Figure 2 and Table 2 summarize the main comparison against prior text-based gradient inversion methods. The overall pattern is clear: SOMP achieves the strongest reconstruction quality once gradients are mixed across multiple samples, and its advantage becomes more pronounced as batch size increases. The largest gains appear on IMDB, where long inputs make cross-sample interference especially severe. Table 1 provides representative qualitative examples comparing DAGER and SOMP. Under mixed gradients, DAGER often merges samples with similar prefixes, repeats locally plausible tokens, or drifts toward nearby embedding-space neighbors. The resulting outputs can remain superficially fluent while failing to separate the underlying samples correctly. In contrast, SOMP more reliably disentangles the mixed signals and reconstructs the original inputs. Additional qualitative examples comparing all methods are provided in Appendix E.2.

The main difficulty comes from increasing batch size, not from sequence length alone. IMDB is the most challenging benchmark in our study because it combines long inputs with cross-sample mixing. As shown in Table 2, reconstruction quality generally degrades as B increases, but prior approaches degrade much more sharply. For example, on GPT-J-6B, DAGER drops from 58.4 at $B = 4$ to 39.2 at $B = 8$, whereas SOMP still reaches 73.5 at $B = 8$. This gap widens further at $B = 16$.

Importantly, the full results in Appendix E show that DAGER remains highly effective at $B = 1$ on IMDB. This indicates that long sequences alone are not the main bottleneck for search-based inversion. The real challenge emerges when multiple long documents are mixed into a single observed gradient, which is exactly the regime where SOMP provides its largest gains. On shorter-sequence benchmarks such as CoLA and SST-2, inversion remains comparatively easier, and both DAGER and SOMP perform strongly at small and moderate batch sizes. Even in these easier settings, however, SOMP degrades more gracefully as B grows. This shows that SOMP is not mainly designed to maximize gains in easy cases, but to remain effective where prior attacks become brittle.

SOMP remains effective at substantially larger batch sizes. Figure 3 extends the comparison to much larger batch sizes on IMDB. We can see that DAGER degrades rapidly as B increases. In contrast, SOMP continues to recover meaningful text even at batch sizes far beyond the range where earlier methods begin to collapse. The same trend appears for both GPT-J-6B and Qwen3-8B, suggest-

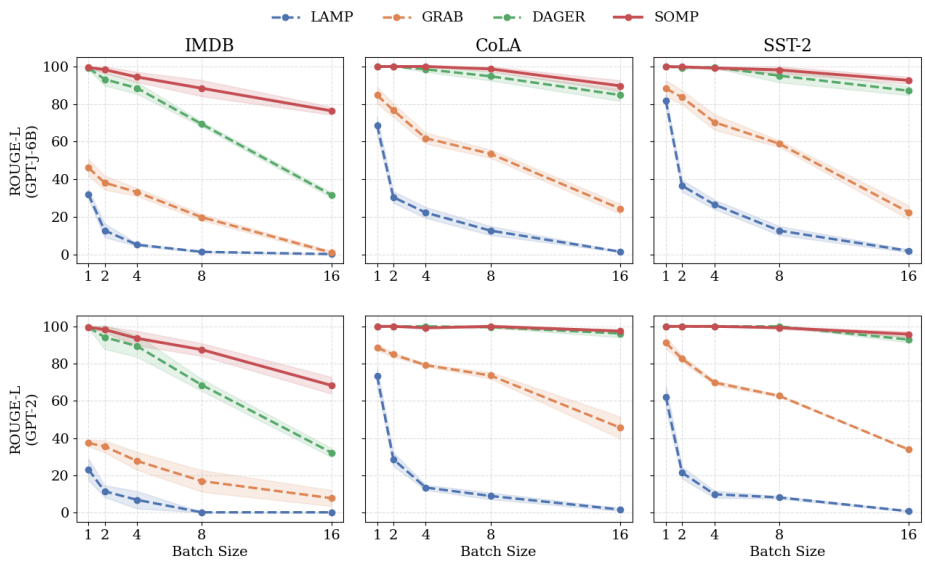


Figure 2: Reconstruction performance (ROUGE-L) under aggregated gradients at batch sizes B = 1, 2, 4, 8, and 16 on IMDB, CoLA, and SST-2 using GPT-J-6B (top) and GPT-2 (bottom). SOMP shows the largest gains on IMDB, whose longer inputs make multi-sample inversion harder than on shorter datasets such as CoLA and SST-2.

Table 2: Reconstruction performance (ROUGE-L) under aggregated gradients at $B = 4, 8, 16$ on GPT-J-6B.

Dataset	Method	$B = 4$	$B = 8$	$B = 16$
IMDB	LAMP	1.2 ± 0.8	0.3 ± 0.3	0.1 ± 0.1
	GRAB	8.6 ± 2.4	2.0 ± 0.6	0.9 ± 0.9
	DAGER	58.4 ± 2.4	39.2 ± 0.8	31.6 ± 1.5
	SOMP	**89.3±1.9**	**73.5±5.1**	**76.4±2.1**
CoLA	LAMP	10.6 ± 1.8	6.3 ± 0.7	1.4 ± 0.3
	GRAB	42.6 ± 2.4	35.1 ± 2.1	24.3 ± 2.6
	DAGER	92.9 ± 3.1	87.8 ± 4.5	84.8 ± 3.2
	SOMP	**96.3±3.7**	**94.5±3.3**	**89.7±2.9**
SST-2	LAMP	12.9 ± 1.2	7.2 ± 0.7	1.8 ± 0.8
	GRAB	43.3 ± 2.3	36.4 ± 2.5	22.3 ± 3.7
	DAGER	100.0 ± 0.0	92.9 ± 1.6	87.1 ± 2.4
	SOMP	**100.0±0.0**	**95.8±1.5**	**92.6±1.9**

Method	B=1	B=2	B=4	B=8	B=16
DAGER	61	118	248	467	1127
SOMP	48	96	191	359	672

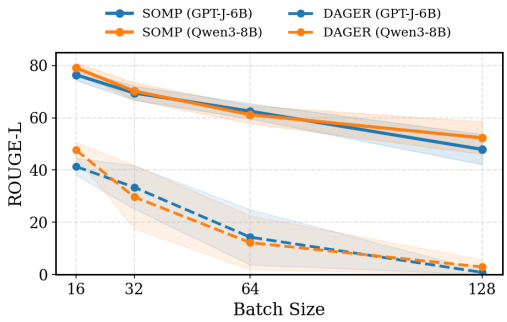
Table 3: Fixed runtime on IMDB, in seconds per batch. SOMP uses $W = 4$.

ing that the advantage of SOMP is not specific to a single architecture family.

5.3 Efficiency and Scalability

SOMP improves efficiency by filtering candidates in informative head-wise subspaces, rather than repeatedly verifying them in the full gradient space. This advantage becomes most visible on long-sequence inputs, where the candidate set is large

Figure 3: Reconstruction performance (ROUGE-L) on GPT-J-6B and Qwen3-8B at large batch sizes on IMDB. SOMP remains effective after DAGER degrades.



and full-space verification becomes expensive.

Table 3 reports fixed-configuration runtime on IMDB. With the low-cost setting $W = 4$, SOMP is faster than DAGER across batch sizes, while also achieving stronger reconstruction quality. SOMP is not intended to be uniformly faster in all regimes: on short sequences, the OMP stage can introduce extra overhead. Full runtime results on CoLA/SST-2 and the beam-width quality–cost tradeoff are reported in Appendix F.

5.4 Multilingual Evaluation

We next test whether SOMP generalizes beyond English. Across five languages, SOMP maintains strong reconstruction quality without language-specific tuning, suggesting that its effectiveness does not depend on English-specific tokenization

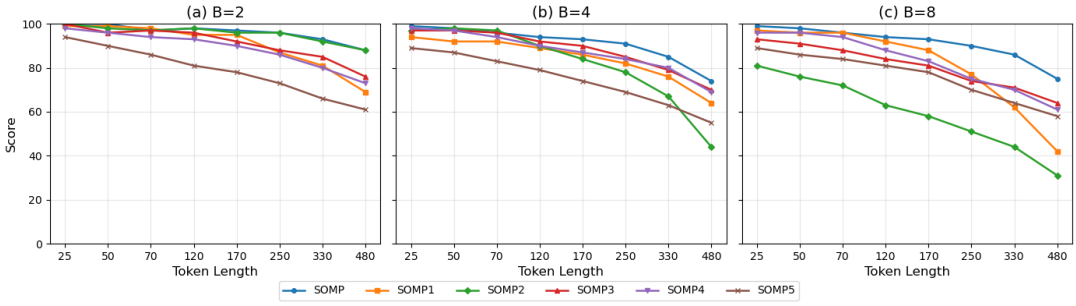


Figure 4: Ablation at batch sizes ($B = 2, 4, 8$). Variants remove LM prior (SOMP1), OMP (SOMP2), sparsity scoring (SOMP3), cross-head consistency (SOMP4), and subspace-distance scoring (SOMP5).

Table 4: Reconstruction under DP-SGD-style clipping and Gaussian noise on GPT-J-6B/IMDB at $B = 8$.

Defense	ROUGE-L
No defense	73.5 ± 5.1
Clip, $C = 1.0$	54.3 ± 6.2
$C = 1.0, \sigma = 0.5, \epsilon = 9.69$	34.2 ± 4.5
$C = 1.0, \sigma = 1.0, \epsilon = 4.85$	19.5 ± 3.9
$C = 1.0, \sigma = 2.0, \epsilon = 2.42$	10.5 ± 3.6

patterns. As shown in Table 12 of Appendix G, performance remains strongest on English, German, and Chinese, while French and Italian degrade more noticeably as batch size increases. A likely reason is heavier subword fragmentation, which leads to longer tokenized sequences and a larger decoding search space. Even so, SOMP remains effective across all evaluated languages, indicating that it exploits structural properties of aggregated transformer gradients that persist across languages.

5.5 Reconstruction under FedAvg

We also evaluate SOMP under FedAvg-style local updates on GPT-J-6B/IMDB with $B = 16$. FedAvg is more challenging than one-step FedSGD because the observed update combines multiple local optimization steps. As shown in Appendix H, local training moderately reduces reconstruction quality, but SOMP still recovers nontrivial text with ROUGE-L around 68–76. This suggests that the leakage exploited by SOMP is not limited to single-step FedSGD.

5.6 DP-SGD-Style Clipping and Noise

We also test SOMP under DP-SGD-style per-sample clipping and Gaussian noise. On GPT-J-6B/IMDB with $B = 8$, each per-sample gradient is clipped to norm C, averaged, and perturbed with Gaussian noise of multiplier σ. This is an attack-side single-update study, not a full privacy–utility evaluation of DP-SGD training. Table 4 shows that clipping alone lowers ROUGE-L, and adding noise further suppresses reconstruction. Thus, SOMP does not claim to break properly configured DP-SGD; rather, aggregation alone is insufficient without explicit clipping-and-noise defenses. Full results are in Appendix I.

5.7 Ablation Study

We ablate the main components of SOMP in Figure 4, including the LM prior, OMP-based sparse reconstruction, sparsity scoring, cross-head consistency, and subspace-distance scoring. Removing OMP causes the largest drop, especially at larger batch sizes and longer sequence lengths, showing that sparse reconstruction is the key step for resolving cross-sample ambiguity in the highly mixed regime. The other components also contribute consistently: the LM prior and cross-head consistency matter more as batch size increases, while sparsity scoring and subspace-based filtering help suppress noisy candidates earlier in the pipeline. Overall, the gains of SOMP come from the interaction of its stages rather than from any single heuristic.

6 Conclusion

We presented SOMP, a subspace-guided and sparsity-driven framework for reconstructing textual inputs from aggregated gradients of large language models. By exploiting the multi-head structure of transformer gradients, SOMP combines geometric filtering, language-model–guided decoding, and sparse gradient-space reconstruction. Experimental results show that SOMP enables accurate and stable multi-sample reconstruction under aggregated-gradient settings, particularly for long sequences and large batch sizes. Overall, SOMP provides a scalable approach to gradient inversion under large-batch aggregated-gradient training.

7 Limitations

SOMP relies on structural regularities in first-layer query gradients and uses head-wise decomposition to reduce the search space. Although modern transformers may adopt alternative attention designs such as grouped-query attention (GQA), our experiments on Qwen3-8B show that performance remains comparable under this architecture. A more challenging setting would be architectures where head-level gradient patterns are substantially less separable.

In addition, Stage I requires maintaining a high-recall token pool. Under severely distorted, compressed, or noisy gradients, relevant tokens may be pruned too early during candidate filtering, affecting downstream reconstruction. From an efficiency perspective, while head-wise factorization reduces the cost of candidate verification, the OMP-based sparse reconstruction in Stage III introduces additional overhead. As a result, SOMP may be slower on short sequences and small batch sizes, but becomes increasingly competitive as sequence length and aggregation level increase.

Finally, SOMP is less effective when gradients are explicitly protected by clipping and Gaussian noise. Our DP-SGD-style clipping-and-noise experiments show that such defenses substantially reduce reconstruction fidelity. Thus, our results should be interpreted as evidence that aggregation alone does not eliminate gradient leakage, not as a break of properly configured DP-SGD. This boundary further highlights the practical importance of explicit clipping-and-noise defenses in federated LLM training.

8 Ethical Considerations

This work investigates the feasibility of reconstructing textual inputs from aggregated gradients, highlighting potential privacy risks in federated and distributed training of large language models. Our goal is to better understand the conditions under which gradient leakage may occur, in order to inform the design of safer and more privacy-aware training protocols.

We do not advocate the misuse of gradient inversion techniques. Instead, our findings emphasize the importance of established privacy safeguards, such as secure aggregation and differential privacy.

References

Mislav Balunović, Dimitar I. Dimitrov, Nikola Jovanović, and Martin Vechev. 2022. Lamp: Extracting text from gradients with language model priors. *Preprint*, arXiv:2202.08827.

Xinping Chen and Chen Liu. 2025. Gradient inversion transcript: Leveraging robust generative priors to reconstruct training data from gradient leakage. *Preprint*, arXiv:2505.20026.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. *Preprint*, arXiv:1912.02164.

Jieren Deng, Yijue Wang, Ji Li, Chenghong Wang, Chao Shang, Hang Liu, Sanguthevar Rajasekaran, and Caiwen Ding. 2021. TAG: Gradient attack on transformer-based language models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3600–3610, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dimitar I. Dimitrov, Maximilian Baader, Mark Niklas Müller, and Martin Vechev. 2024. Spear:exact gradient inversion of batches in federated learning. *Preprint*, arXiv:2403.03945.

Xinguo Feng, Zhongkui Ma, Zihan Wang, Eu Joe Chegne, Mengyao Ma, Alsharif Abuadbba, and Guangdong Bai. 2024. Uncovering gradient inversion risks in practical language model training. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, CCS '24, page 3525–3539, New York, NY, USA. Association for Computing Machinery.

Liam H Fowl, Jonas Geiping, Steven Reich, Yuxin Wen, Wojciech Czaja, Micah Goldblum, and Tom Goldstein. 2023. Decepticons: Corrupted transformers breach privacy in federated learning for language models. In *The Eleventh International Conference on Learning Representations*.

Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. 2020. Inverting gradients - how easy is it to break privacy in federated learning? In *Advances in Neural Information Processing Systems*, volume 33, pages 16937–16947. Curran Associates, Inc.

Yangsibo Huang, Samyak Gupta, Zhao Song, Kai Li, and Sanjeev Arora. 2021. Evaluating gradient inversion attacks and defenses in federated learning. *Preprint*, arXiv:2112.00059.

Jian Li, Baosong Yang, Zi-Yi Dou, Xing Wang, Michael R. Lyu, and Zhaopeng Tu. 2019. Information aggregation for multi-head attention with routing-by-agreement. *Preprint*, arXiv:1904.03100.

Qiongxiu Li, Lixia Luo, Agnese Gini, Changlong Ji, Zhanhao Hu, Xiao Li, Chengfang Fang, Jie Shi, and Xiaolin Hu. 2024. Perfect gradient inversion in federated learning: A new paradigm from the hidden subset sum problem. *Preprint*, arXiv:2409.14260.

Zhuohang Li, Jiaxin Zhang, Luyang Liu, and Jian Liu. 2022. Auditing privacy defenses in federated learning via generative gradient leakage. *Preprint*, arXiv:2203.15696.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Ivo Petrov, Dimitar I. Dimitrov, Maximilian Baader, Mark Niklas Müller, and Martin Vechev. 2024. Dager: Exact gradient inversion for large language models. *Preprint*, arXiv:2405.15586.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

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

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep leakage from gradients. *Preprint*, arXiv:1906.08935.

A Head-wise Sparsity Decomposition

In this appendix, we provide a formal justification showing that measuring sparsity globally or in a head-wise manner is equivalent under a block-wise decomposition of feed-forward network (FFN) parameters.

A.1 Additivity of Head-wise Sparsity

Proposition A.1 (Additivity of Head-wise Sparsity). Let $G \in \mathbb{R}^{d\times D}$ be a gradient-related matrix associated with a feed-forward network layer. Assume that G is partitioned column-wise into H disjoint blocks:

$$G = [\, G^{(1)}, G^{(2)}, \ldots, G^{(H)} \,], \tag{20}$$

where $G^{(h)} \in \mathbb{R}^{d\times D_h}$ and $\sum_{h=1}^{H} D_h = D$.

For a token embedding $e \in \mathbb{R}^d$, define its projected response

$$g = G^\top e \in \mathbb{R}^D, \tag{21}$$

and correspondingly the head-wise responses

$$g^{(h)} = (G^{(h)})^\top e \in \mathbb{R}^{D_h}. \tag{22}$$

Then the global sparsity of g, measured by the number of non-zero entries, satisfies

$$\|g\|_0 = \sum_{h=1}^{H} \|g^{(h)}\|_0. \tag{23}$$

An analogous decomposition holds for threshold-based sparsity measures where entries with magnitude below a fixed threshold are treated as zero.

Proof. By construction, the blocks $G^{(h)}$ correspond to disjoint column subsets of G. Consequently, the vectors $g^{(h)}$ occupy disjoint coordinate subsets of g. Since sparsity measures such as the ℓ_0 count (or thresholded variants) operate independently on each coordinate, the total number of non-zero entries in g is exactly the sum of the non-zero entries across all head-wise components:

$$\|g\|_0 = \sum_{h=1}^{H} \|g^{(h)}\|_0. \tag{24}$$

A.2 Discussion

Proposition A.1 shows that evaluating sparsity in a head-wise manner is mathematically equivalent to evaluating sparsity on the full projected response under a block-wise decomposition of FFN parameters. Therefore, adopting a head-aware sparsity formulation does not introduce additional assumptions, nor does it alter the underlying sparsity-based filtering objective. Instead, it provides a structured reformulation aligned with the architectural organization of modern transformer FFNs and facilitates more interpretable and numerically stable sparsity estimation.

B Effect of Surrogate Labels in Stage III

The Stage III gradient atom is defined as

$$g_i = \nabla_\theta \mathcal{L}(f_\theta(x_i), y_i).$$

Since the true labels of candidate sentences are unknown at reconstruction time, we compute candidate gradients using a fixed surrogate label.

Gradient decomposition. For a softmax classifier with cross-entropy loss,

$$\nabla_\theta \mathcal{L}(x, y) = J_\theta(x)^\top (p - y),$$

where $J_\theta(x) = \frac{\partial z}{\partial \theta}$ is the Jacobian of the logits and $p = \mathrm{softmax}(z)$. Thus the gradient consists of an input-dependent Jacobian $J_\theta(x)$ and a label-dependent coefficient vector $(p - y)$.

If the true label y is replaced by a surrogate label $\tilde{y}$, the gradient becomes

$$\tilde{g}(x) = J_\theta(x)^\top (p - \tilde{y}).$$

Both $g(x, y)$ and $\tilde{g}(x)$ therefore lie in the same subspace

$$\mathrm{colspan}(J_\theta(x)^\top),$$

meaning that replacing the label preserves the input-dependent gradient structure and only changes the coefficient vector.

Implication for Stage III. Stage III ranks candidates using gradient alignment scores

$$\langle g, r_t \rangle.$$

Replacing the true label only modifies the coefficient vector in g_r while leaving the Jacobian interaction structure unchanged. Consequently, the dominant input-dependent gradient structure is preserved, and in practice the resulting candidate ranking is only weakly affected by the choice of label.

Label	ROUGE-L	OMP overlap
True	71.7 ± 2.1	100.0
Fixed	68.4 ± 1.6	93.1 ± 1.9
Random	70.9 ± 3.2	94.6 ± 2.3
Model-predicted	73.5 ± 2.7	97.6 ± 2.4

Table 5: Effect of label choices in Stage III on GPT-J-6B/IMDB at $B = 32$. OMP overlap denotes the percentage of selected candidates that overlap with the selection obtained using true labels.

Empirical validation. We compare true, fixed, random, and predicted labels on GPT-J-6B/IMDB at $B = 32$. The true-label setting is used as a reference for candidate selection, but it is not necessarily an upper bound on ROUGE-L because Stage II may generate imperfect candidates and different label choices can slightly change which plausible candidate is selected by OMP. Table 5 shows that ROUGE-L and OMP overlap vary mildly, suggesting that Stage III is not driven by privileged access to true labels.

C Hyperparameters

This appendix summarizes the hyperparameters used in **SOMP**. We distinguish between batch-size-independent hyperparameters, which are fixed across all experiments, and batch-size-dependent hyperparameters, which vary with the aggregation level B.

The batch-size-dependent parameters reflect the increased ambiguity and sparsity requirements induced by larger gradient mixtures. Importantly, these parameters follow a fixed strategy shared across datasets, languages, and models, rather than being tuned per task or per instance.

Category	Hyperparameter	Value
Stage I	λ_1	0.8
	λ_2	0.5
	λ_3	0.5
	$\tau_m^{(h)}$	$0.5\times$ head-wise median
	τ_{svd}	10^{-3}
	$r_{\max}$	$\min(64, d_h/2)$
Stage II	β_{LM}	0.33
	λ_{div}	0.15
	λ_{ng}	0.2
Stage III	λ	$1e{-3}$
	ϵ_{OMP}	$1e{-4} \cdot \|\boldsymbol{g}_{\text{mix}}\|_2$
	τ_{cluster}	$0.8\times$ Rouge-L
Model / Exp.	H	model-specific
	P	512

Table 6: Batch-size-independent hyperparameters used in SOMP.

Algorithm 1 Stage I: Head-Structured Token Pooling

Require: Aggregated gradient g_{mix}, vocabulary V
Ensure: Filtered token pool $\mathcal{P}$

function TOKENPOOLING(g_{mix}, V)
  Decompose g_{mix} into $\{G^{(h)}\}_{h=1}^{H}$
  Compute $\boldsymbol{P}_{\mathcal{R}_Q^{(h)}}$ by truncated SVD of $G^{(h)}$ using τ_{svd} and $r_{\max}$
  $\mathcal{P} \leftarrow \emptyset$
  for $v \in V$ **do**
    Compute $s_{\text{sub}}(v)$, $s_{\text{cons}}(v)$, $s_{\text{sparse}}(v)$
    $s_{\text{total}} \leftarrow \lambda_1 s_{\text{sub}} + \lambda_2 s_{\text{cons}} - \lambda_3 s_{\text{sparse}}$
    if $v \in$ Top-k **then**
      $\mathcal{P} \leftarrow \mathcal{P} \cup \{v\}$
    end if
  end for
  return $\mathcal{P}$
end function

Hyperparameter	B=1	B=4	B=8	B=16+
Token pool size k	960	1600	2400	3200
Informative heads $\|H_{\text{act}}\|$	$H/4$	$H/4$	$H/3$	$H/3$
Sparsity heads $\|H_{\text{top}}\|$	$H/6$	$H/6$	$H/4$	$H/4$
Beam width W	2	4	6	12
Number of beam groups G	1	4	8	16
OMP iterations (max)	B	B	B	B

Table 7: Batch-size-dependent hyperparameters as a function of aggregation level B.

Discussion. As the batch size increases, the aggregated gradient represents a mixture of a larger number of samples, increasing both combinatorial ambiguity and sparsity requirements. Accordingly, SOMP increases the size of the candidate token pool and the breadth of sentence-level search while maintaining fixed geometric and sparsity weighting. This strategy enables stable reconstruction across a wide range of aggregation levels without per-task hyperparameter tuning.

Token-pool recall. We measure the fraction of ground-truth tokens retained by the Stage-I token pool before decoding. On CoLA, small pools already preserve all ground-truth tokens in the tested settings. On IMDB at $B = 16$, recall increases from 71.3 ± 2.4 with pool size 800 to 94.6 ± 1.5 with pool size 1600 and 99.6 ± 0.4 with pool size 3200.

C.1 Hyperparameter Sensitivity Analysis

We analyze the sensitivity of SOMP to several key hyperparameters used in the three-stage reconstruc-

Algorithm 2 Stage II: Diverse Beam Decoding

Require: Global pool $\mathcal{P}$, subspaces $\mathcal{R}$, groups G
Ensure: Candidate set $\mathcal{X}$

```
function DiverseBeamSearch(P, R)
    Detect candidate lengths L ← {L_1, ...}
    X_raw ← ∅
    for L ∈ L do
        for t ← 1ToL do
            P_t ← {v ∈ P | s_sub(v, t) < τ_pos}
        end for
        Initialize beam groups B_1, ..., B_G
        for t ← 1ToL do
            for g ∈ 1...G do
                Expand B_g using P_t
                S ← score_t^base + λ_div 1(x_t ∈ S_t) + λ_ng 1(g_t ∈ G_t)
                Prune to top-W/G
            end for
        end for
        X_raw ← X_raw ∪ ⋃ B_g
    end for
    C ← Cluster(X_raw)
    return X ← {Rep(c) | c ∈ C}
end function
```

tion pipeline. In particular, we examine the influence of parameters controlling token pooling, beam search, and sparse reconstruction.

Unless otherwise stated, experiments are conducted on the IMDB dataset using GPT-J with batch size $B = 8$. For each experiment, we vary one hyperparameter while keeping all others fixed according to the hyperparameter settings above. Reconstruction quality is measured using ROUGE-L.

Overall, SOMP behaves as expected under hyperparameter variation, but remains stable under moderate changes. Small to moderate deviations in parameter values do not lead to abrupt degradation in reconstruction performance, indicating that the method does not rely on fragile tuning.

Discussion. The results show that SOMP is influenced by hyperparameter choices, as larger token pools or beam widths generally improve candidate coverage at the cost of increased computation. However, performance varies smoothly across a reasonable range of parameter values. In particular, moderate changes in k, W, or λ do not cause catastrophic degradation in reconstruction quality. These observations suggest that SOMP does not depend on fragile hyperparameter tuning and remains robust under typical parameter variations.

D Algorithmic Details

This appendix provides pseudocode for the three stages of SOMP, complementing the high-level descriptions in Section 4. Algorithm 1 details head-structured token pooling, Algorithm 2 presents geometry-driven diverse beam decoding, and Algorithm 3 describes gradient-space sparse reconstruction via Orthogonal Matching Pursuit (OMP).

The pseudocode is intended to improve clarity and reproducibility and does not introduce additional assumptions beyond those already discussed in the main text.

Table 8: Hyperparameter sensitivity analysis of SOMP. Each parameter is varied while keeping the others fixed.

Parameter	Value	ROUGE-L
Stage I: Token Pooling		
Token pool size k	800	59.7
	1200	71.5
	1600	73.4
	2000	74.5
	2400	73.3
λ_1	0.6	74.7
	0.8	67.3
	1.0	53.4
λ_2	0.3	68.4
	0.5	74.5
	0.7	61.7
Stage II: Beam Decoding		
Beam width W	2	37.1
	4	59.9
	6	66.4
	8	72.7
	12	79.3
λ_{div}	0.1	62.2
	0.15	74.7
	0.2	69.5
λ_{ng}	0.1	64.8
	0.2	74.8
	0.3	71.5
Stage III: Sparse Reconstruction		
λ	5×10^{-4}	69.8
	1×10^{-3}	74.2
	5×10^{-3}	71.6
	1×10^{-2}	73.7
ϵ	5×10^{-5}	76.5
	1×10^{-4}	73.9
	5×10^{-4}	68.4
$\tau_{cluster}$	0.6	67.0
	0.8	73.5
	1.0	81.4

E Full Main Results

This appendix reports the full reconstruction results in Table 9 corresponding to the main experiments in Section 5. We include ROUGE-1 and ROUGE-2 across batch sizes $B \in \{1, 2, 4, 8\}$ for all compared methods under the same setup as the main text.

Dataset	Method	R-1	R-2	R-1	R-2	R-1	R-2	R-1	R-2
		$B = 1$		$B = 2$		$B = 4$		$B = 8$	
				GPT-J					
IMDB	LAMP	31.9±2.4	16.5±3.1	12.7±3.7	7.6±2.0	5.1±0.7	1.4±0.3	1.3±0.4	0.3±0.3
	GRAB	46.1±4.4	33.4±1.4	38.1±3.7	21.7±0.9	33.1±2.1	9.2±1.3	19.7±1.7	2.0±0.6
	DAGER	99.2±0.8	94.3±3.2	93.2±3.4	72.0±2.8	88.5±3.1	58.4±2.4	69.3±1.3	39.2±0.8
	SOMP	**99.5±0.5**	**96.5±2.1**	**98.3±1.7**	**93.6±3.8**	**94.4±2.7**	**89.3±1.9**	**88.4±4.3**	**73.5±5.1**
CoLA	LAMP	68.6±4.5	42.6±4.7	30.4±3.1	17.8±1.6	22.2±2.9	10.4±1.5	12.6±2.4	6.3±0.7
	GRAB	84.7±4.4	62.5±2.7	76.7±3.6	57.2±3.8	61.8±2.9	42.6±2.4	53.5±2.2	35.1±2.1
	DAGER	100.0±0.0	100.0±0.0	100.0±0.0	100.0±0.0	98.5±1.5	92.9±3.1	94.8±2.3	87.8±4.5
	SOMP	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**96.3±3.7**	**98.7±1.3**	**94.5±3.3**
SST-2	LAMP	81.8±3.0	57.8±6.9	36.5±3.2	23.6±3.8	26.5±2.5	12.9±1.2	12.6±2.4	7.2±0.7
	GRAB	88.3±4.4	66.3±1.2	83.6±3.7	56.5±4.5	70.3±4.2	43.3±2.3	58.9±1.7	36.4±2.5
	DAGER	100.0±0.0	100.0±0.0	99.3±0.7	92.0±1.8	99.6±0.4	90.4±3.3	95.1±3.6	86.8±2.9
	SOMP	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**99.8±0.2**	**99.2±0.8**	**97.6±2.4**	**98.2±1.8**	**95.9±2.5**
Dataset	**Method**	**R-1**	**R-2**	**R-1**	**R-2**	**R-1**	**R-2**	**R-1**	**R-2**
		$B = 1$		$B = 2$		$B = 4$		$B = 8$	
				GPT-2					
IMDB	LAMP	23.2±5.8	18.8±8.4	11.4±3.2	7.4±1.6	6.8±4.7	2.7±2.4	0.1±0.1	0.1±0.1
	GRAB	37.3±1.8	29.1±1.0	35.5±3.2	21.1±4.5	27.7±4.8	12.8±3.2	16.9±5.8	7.7±4.3
	DAGER	99.5±0.5	94.0±4.6	87.2±6.4	78.8±4.7	86.4±5.8	71.6±4.1	68.2±3.2	42.5±2.4
	SOMP	**99.6±0.4**	**94.4±3.6**	**98.3±1.7**	**94.6±2.3**	**93.6±3.9**	**86.3±5.5**	**87.4±3.4**	**68.5±4.5**
CoLA	LAMP	73.3±4.5	43.3±7.0	28.6±3.7	11.0±3.8	13.4±1.4	3.9±1.2	8.9±1.6	1.6±0.8
	GRAB	88.6±1.6	65.7±1.4	84.9±1.2	56.8±1.2	79.2±0.7	52.8±3.0	73.6±1.7	45.6±5.9
	DAGER	100.0±0.0	100.0±0.0	100.0±0.0	100.0±0.0	100.0±0.0	95.6±1.8	99.6±0.4	96.2±2.1
	SOMP	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**98.2±0.0**	**99.2±0.8**	**94.0±2.9**	**100.0±0.0**	**97.5±0.5**
SST-2	LAMP	62.2±6.2	31.8±8.4	21.4±3.2	9.5±3.6	9.8±1.9	2.6±1.4	8.1±0.7	0.7±0.4
	GRAB	91.2±0.5	63.4±0.0	82.6±1.7	53.7±0.4	69.8±1.2	38.1±0.6	62.7±0.6	33.8±0.1
	DAGER	100.0±0.0	96.0±3.0	100.0±0.0	89.5±1.4	100.0±0.0	92.8±2.4	100±0.0	92.9±1.6
	SOMP	**100.0±0.0**	**98.8±0.8**	**100.0±0.0**	**97.5±0.9**	**100.0±0.0**	**98.4±0.4**	**99.3±0.7**	**95.8±1.5**

Table 9: Full reconstruction results for the main experiments, reporting ROUGE-1 and ROUGE-2 across datasets and batch sizes for GPT-J and GPT-2.

Algorithm 3 Stage III: OMP Reconstruction

Require: g_{mix}, candidates $\mathcal{X}$, batch size B
Ensure: Reconstructed batch $\mathcal{S}^*$
function OMP($g_{\text{mix}}, \mathcal{X}$)
  Dictionary $\mathcal{D} \leftarrow \{\nabla\mathcal{L}(x) \mid x \in \mathcal{X}\}$
  $r_0 \leftarrow g_{\text{mix}}, \quad \Lambda_0 \leftarrow \emptyset$
  for $k = 1 \rightarrow B$ **do**
    $j^* \leftarrow \arg\max_j \frac{|\langle g_j, r_{k-1}\rangle|}{\|g_j\|_2}$
    $\Lambda_k \leftarrow \Lambda_{k-1} \cup \{j^*\}$
    $\alpha^* \leftarrow \arg\min_\alpha \left\|g_{\text{mix}} - \sum_{i\in\Lambda_k} \alpha_i g_i\right\|_2^2 + \lambda\|\alpha\|_2^2$
    $r_k \leftarrow g_{\text{mix}} - \sum_{i\in\Lambda_k} \alpha_i^* g_i$
    if $\|r_k\| < \epsilon$ **then**
      break
    end if
  end for
  return $\mathcal{S}^* \leftarrow \{\, x_j \mid j \in \Lambda_k \,\}$
end function

E.1 Named-Entity Recovery

To complement ROUGE-based evaluation, we further measure entity-level information recovery using named-entity exact-match recall. We extract named entities from the reference inputs using spaCy's `en_core_web_sm` English NER model with default settings and check whether each normalized entity appears exactly in the reconstructed text. This metric is used as a lightweight proxy for entity-level leakage, rather than a domain-specific medical or legal privacy evaluation.

For a reference input x and reconstruction $\hat{x}$, let $E(x)$ denote the set of named entities extracted from x. We define

$$\text{NE-EM} = \frac{|\{e \in E(x) : \text{norm}(e) \subseteq \text{norm}(\hat{x})\}|}{|E(x)|},$$

where $\text{norm}(\cdot)$ lowercases text and removes punctuation and extra spaces. Samples without extracted named entities are excluded from this metric.

Table 10 shows that SOMP also improves entity-level recovery under aggregated gradients, following the same trend as the ROUGE-based results.

Method	$B = 8$	$B = 16$	$B = 32$
DAGER	77.9 ± 3.6	68.4 ± 5.8	54.1 ± 6.2
SOMP	90.5 ± 2.1	83.7 ± 3.1	69.0 ± 3.9

Table 10: Named-entity exact-match recall (%) on GPT-J-6B/IMDB.

E.2 Qualitative Reconstruction Example

To complement the quantitative results, we present a qualitative reconstruction example under aggregated gradients with batch size $B = 2, 4, 8$. Tables 15, 16, and 17 compare the outputs generated by SOMP and several baselines on the same gradient batch. While prior methods such as LAMP, GRAB, and DAGER often produce mismatched or mixed sentences due to cross-sample token interference, SOMP is able to recover the original inputs more accurately.

F Runtime and Beam-Width Tradeoff

Table 11 reports the full runtime results across datasets and SOMP beam widths. The main text uses $W = 4$ as the fixed low-cost configuration for runtime comparison, while larger beam widths are reported here to show the quality–cost tradeoff.

The runtime behavior differs across datasets. On short-sequence datasets such as CoLA and SST-2, DAGER is already efficient because the search space is small. In this regime, SOMP may be comparable to or slightly slower, since Stage III introduces an additional OMP selection step. In contrast, on IMDB, the longer inputs create a much larger candidate space, making repeated full-gradient verification more expensive for DAGER. SOMP benefits more in this setting because head-wise subspace filtering reduces the candidate set before sparse reconstruction.

Beam width controls the tradeoff between candidate coverage and cost. Increasing W gives Stage II more candidate sentences and can improve reconstruction quality, but it also increases decoding and OMP cost. Therefore, we use $W = 4$ as the fixed runtime comparison in the main text and report $W = 8$ and $W = 16$ only as quality–cost variants.

G Multilingual Experiment

This appendix provides additional multilingual reconstruction results in Table 12 corresponding to Section 5.4. We report SOMP's performance across five languages and analyze the effect of tokenization differences on reconstruction quality.

Table 11: Runtime under different SOMP beam widths. Larger beam widths expose the quality–cost tradeoff and are not used as the fixed runtime comparison in Table 3.

Dataset	Method	B=1	B=2	B=4	B=8	B=16
GPT-2 (runtime measured on RTX 5060)						
CoLA	DAGER	3	6	10	19	36
	SOMP(Bw=4)	7	10	13	21	26
	SOMP(Bw=8)	7	10	14	24	31
	SOMP(Bw=16)	8	13	21	33	41
SST-2	DAGER	3	4	9	18	31
	SOMP(Bw=4)	6	9	13	22	30
	SOMP(Bw=8)	7	11	14	26	37
	SOMP(Bw=16)	8	12	30	40	52
IMDB	DAGER	61	118	248	467	1127
	SOMP(Bw=4)	48	96	191	359	672
	SOMP(Bw=8)	58	117	231	448	968
	SOMP(Bw=16)	75	143	289	572	1308

Language	B=1	B=4	B=8	B=16
English	99.7±0.3	94.3±1.4	88.2±1.7	83.5±2.1
German	99.6±0.4	94.6±0.8	89.7±2.1	82.3±2.6
Chinese	99.8±0.2	93.1±1.2	88.1±1.4	84.4±3.4
Italian	99.7±0.3	90.2±1.5	83.3±1.8	78.1±1.8
French	95.7±2.1	86.3±3.3	80.5±3.6	72.8±3.7

Table 12: Multilingual reconstruction performance of **SOMP** on Qwen3 across batch sizes on `stanfordNLP/imdb`.

Across all five evaluated languages, SOMP remains effective without language-specific tuning, indicating robustness to different tokenization schemes. Performance is strongest on English, German, and Chinese, while Italian and French show somewhat larger degradation as batch size increases. This pattern is consistent with heavier subword fragmentation in languages that produce longer tokenized sequences and larger candidate branching during decoding.

H Reconstruction under FedAvg

This appendix provides the FedAvg results corresponding to Section 5.5. To better align this evaluation with the main long-sequence and large-model setting, we evaluate SOMP on GPT-J-6B with IMDB at batch size $B = 16$. Clients perform E local epochs with learning rate η and local mini-batch size $B_{\min}$ before aggregation. Reconstruction quality is measured by ROUGE-L.

Results. Table 13 shows that FedAvg introduces some degradation compared with the corresponding non-FedAvg setting, but the gap is limited and

E	η	$B_{\min}$	ROUGE-L
2	10^{-4}	4	75.7 ± 2.0
5	10^{-4}	4	73.0 ± 3.2
10	10^{-4}	4	70.8 ± 4.1
20	10^{-4}	4	69.1 ± 3.8
10	10^{-5}	4	75.4 ± 4.4
10	5×10^{-5}	4	72.3 ± 2.7
10	10^{-4}	4	69.1 ± 2.9
10	10^{-4}	2	67.9 ± 3.5
10	10^{-4}	8	70.5 ± 3.3
10	10^{-4}	16	76.1 ± 3.1

Table 13: FedAvg-style reconstruction on GPT-J-6B/IMDB at $B = 16$.

Defense	Setting	ROUGE-L
No defense	–	73.5 ± 5.1
Clip only	$C = 1.0$	54.3 ± 6.2
Clip only	$C = 1.5$	62.5 ± 6.0
Clip only	$C = 2.0$	67.2 ± 5.7
Clip+noise	$C = 1.0, \sigma = 0.5, \epsilon = 9.69$	34.2 ± 4.5
Clip+noise	$C = 1.0, \sigma = 1.0, \epsilon = 4.85$	19.5 ± 3.9
Clip+noise	$C = 1.0, \sigma = 2.0, \epsilon = 2.42$	10.5 ± 3.6
Clip+noise	$C = 1.5, \sigma = 0.5, \epsilon = 9.69$	38.4 ± 4.8
Clip+noise	$C = 1.5, \sigma = 1.0, \epsilon = 4.85$	28.5 ± 5.3
Clip+noise	$C = 1.5, \sigma = 2.0, \epsilon = 2.42$	12.0 ± 5.1
Clip+noise	$C = 2.0, \sigma = 0.5, \epsilon = 9.69$	44.1 ± 7.2
Clip+noise	$C = 2.0, \sigma = 1.0, \epsilon = 4.85$	25.5 ± 4.5

Table 14: Full reconstruction results under DP-SGD-style per-sample clipping and Gaussian noise on GPT-J-6B/IMDB at $B = 8$.

does not change the overall conclusion. Increasing the number of local epochs or the learning rate generally makes reconstruction harder, while the effect of local mini-batch size is not strictly monotonic. Across the tested FedAvg settings, SOMP still recovers nontrivial text with ROUGE-L around 68–76, suggesting that its leakage signal is not specific to one-step FedSGD.

I Full Results under DP-SGD-Style Clipping and Gaussian Noise

This appendix reports the full results for the DP-SGD-style clipping-and-noise experiment in Section 5.6. We evaluate the standard clipping-and-noise release mechanism used in DP-SGD at the level of one released aggregated update. Specifically, each per-sample gradient is clipped to norm C, averaged over the batch, and then perturbed with Gaussian noise controlled by multiplier σ. We report the corresponding privacy budget ϵ as a reference for the noise strength under this single-update setting.

Experimental setup. Experiments are conducted on GPT-J-6B with IMDB at batch size $B = 8$. Reconstruction quality is measured by ROUGE-L. We vary the clipping norm C and noise multiplier σ. The reported ϵ values are provided to calibrate the strength of the perturbation and should not be interpreted as a full privacy–utility evaluation of multi-round DP-SGD training.

Results. Table 14 shows that clipping alone reduces reconstruction quality, and Gaussian noise further suppresses leakage. These results do not imply that SOMP breaks properly configured DP-SGD. Instead, they show that aggregation alone is insufficient, while explicit clipping-and-noise defenses substantially reduce reconstruction fidelity.

Reference Inputs	LAMP	GRAB	DAGER	SOMP
I received this movie as a gift, I knew from ... movies, don't watch this movie.	I camcorder movie" DVD sucks filmmaker knew wane bee ... movies, Sgt this movie.	town watched it produce movie, Ben Draper ... ries HORROR revenge woman absolute.	I received this movie as a gift, I knew from ... movies, don't watch this movie.	I received this movie as a gift, I knew from ... movies, don't watch this movie.
I very much looked forward to this movie. ... a good love story.	I very Michael place Willie generally to they ... Chris capt. 's context 're romantic.	Its Jr.'s editing to much is be Too. better care ... love Willie and Missy;series.	I very much looked forward to Michael ... a good love Michael.	I very much looked forward to this movie. ... a good love story.
Protocol is an implausible movie whose only ... way to see this film.	Protocol Love's implausible Missy very much ... because see Plus.	Jr.'s is But, implausible romantic always only ... always enjoy role film.	Protocol is an implausible movie whose only ... way to see this movie.	Protocol is an implausible movie whose only ... way to see this film.
I was excited to view a Cataluña´s film in ... to go to see this film.	I was excited Berlin´s a Cataluña´s film in ... , was horrible. competitior	Cataluña, too horr to view a Cataluña´s see film ... to Molina this film.	I was excited to view a Cataluña´s film in ... to go to see this film.	I was excited to view a Cataluña´s film in ... to go ´s to see Cataluña´s film.

Table 15: Qualitative reconstruction examples under aggregated gradients with batch size $B = 2$. Highlighted spans indicate incorrect tokens caused by cross-sample mixing.

Reference Inputs	LAMP	GRAB	DAGER	SOMP
A model named Laura is working in South America when ... jungle. It would probably be more fun. 3/10	A Laura named ive fake Laura in America America ... vision. far). Franco happening. probab be film fun. 3/10	primitives Hunter named Devil. films working drags ity in (25 ...laugh. hilariously killed real fun more fun. . 	A model named Laura is working in hotel when Devil Hunter is film film her hotel ... It would probably be more fun. 3/10	A model named Laura is work in South America when ...jungle. It would probably be more fun.3/10 .3/10.3/10.
Devil Hunter gained notoriety for the fact that ... no reason to bother with this one.	Devil Hunter film notoriety for the 'Video Nasty' that's films the DPP ... with this one devil blood horror.	DPP Hunter gained notoriety for films kidnapped that title). the the ... She Kills in Ecstasy Nasty list but it really needn't have been	Devil Hunter gained notoriety for the fact that ... basic Venus film films Devil this one.	Devil Hunter gained notoriety for the fact that ... no reason to bother with this one one
This film seemed way too long even ... probably didn't see an uncut version.	This film version too uncut probably way ... scene blood camera long didn't see .	75 zombie too long seemed even ... dubbing didn't rocky dubbing way cliff version.	This film seemed way too long even ... film films't zombie / flick film films...	This film seemed way too long even ... probably didn't see an uncut version.
I cannot stay indifferent to Lars van Trier's films. I ... to the viewers to decide this alone.	I films alone Trier's viewers ... cannot decide horror camera devil Lars stay.	Dancer cannot to consider different van van Lars's masterpiece ... pre- ambiguous decide this Europa Europa .	I cannot stay indifferent to Lars van Trier's films. I ... different filmmaker van Lars van demonstrative this alone.'s..	I cannot stay different to Lars van Trier's films. I ... to the viewers to decide this alone ...
Ah, Channel 5 of local Mexican t.v. Everyday, at 2:00 ... film-making. You have been warned.	Ah, warned Mexican 2:00 Channel ... film-making blood devil street local been Everyday.	Ah, Mexican Ah, Ah, Mexican t.v. Ah,... Everyday at 2:00 been film-making. You have warned.	Ah, Channel 5 of local Mexican Ah, Mexican, at 2:00... film-film film. <b cinematography warned.	Ah, Channel 5 of local Mexican Ah, Everyday, at 2:00 ... film-making.Ah,Ah, You have been warned.

Table 16: Qualitative reconstruction examples under aggregated gradients with batch size $B = 4$. Highlighted spans indicate incorrect tokens caused by cross-sample mixing.

Reference Inputs	LAMP	GRAB	DAGER	SOMP
Sloppily directed, witless comedy that supposedly ... be begging for some peace and quiet. (*1/2)	peace quiet comedy begging sloppily * ... hotel blood 't camera witless scene .	quiet comedy peace ... witless directed 's begging some film street .	Sloppily directed, witless comedy that supposedly ... doomed James James 50s Royale Royale Royale Royale..*(*2(.	Sloppily directed, witless comedy that comedy spoofs ... be begging some Royale and Royale..(*1/2)...
I was disgusted by this movie. No it wasn't ... movie, but find out the true story of Artemisia.	movie disgusted Artemisia true ... story Beaver devil blood find out .	Artemisia movie disgusted ... Beaver Beaver find scene film .	I was disgusted by this movie. No it wasn't ... movie, but movie movie the Artemisia Gentileschi.	I was disgusted by this movie. No it it 't ... movie, but find out the true Artemisia of Artemisia....
I didn't think the French could make a bad movie, but ... pornography. So the French can fail, after all!	French pornography bad movie ... after fail camera street horror French .	pornography think French ... bad movie fail after film camera .	I was disgusted by this movie. No it wasn't ... movie, but movie movie the Artemisia Gentileschi.	I didn't think the French could make a bad movie, but ... pornography. So the French can fail, after all! , after all!, after all!!!!!
I read somewhere that when Kay Francis refused to ... and you'll be doing yourself a favor.	favor somewhere yourself doing ... Kay Francis camera jungle blood .	favor Francis somewhere ... doing yourself Kay film scene .	I read somewhere that when Kay Francis refused to ... and Francis'll movie Also, a favor. And her hair!!!	I read somewhere that when Kay Francis refused to ... and you'll be yourself a favor favor a a a a...
A stale "misfits-in-the-army" saga, which ... Me", has only two or three brief scenes. What a waste! (*1/2)	saga brief scenes ... waste army devil hotel camera three .	brief films ... scenes to waste army film .	A stale "misfits-in-the-army" saga, which ... Me", saga only two misfits misfits brief scenes. What a waste! (*1/2)	A stale "misfits-in-the-army" saga, which ... Me", has only two three scenes. What a waste! (*1/2)! (*1/2))
When I was at the movie store the other day, I passed up ... and Blonder is a huge blonde BOMBshell. 1/10	When I movie store ... Michelle Romy Anderson blood street scene day .	Blonder store ... huge blonde stupid movie Michelle film .	When I was at the movie store the other day, I passed up ... and Blonder is a huge BOMB. 1/10 1/10	When I was at the movie store the other day, I passed up ... and Blonder is a blonde Blonder . 1/10 ..... D

Table 17: Qualitative reconstruction examples under aggregated gradients with batch size $B = 8$. Highlighted spans indicate incorrect tokens caused by cross-sample mixing.