Rep2Text: Decoding Full Text from a Single LLM Token Representation

Anonymous ACL submission

Abstract

Large language models (LLMs) have achieved remarkable progress across diverse tasks, yet their internal mechanisms remain largely opaque. In this work, we address a fundamental question: to what extent can the original input text be recovered from a single last-token representation within an LLM? We propose Rep2Text, a novel framework for decoding full text from last-token representations. Rep2Text employs a trainable adapter that projects a target model's internal representations into the embedding space of a decoding language model, which then autoregressively reconstructs the input text. Experiments on various model combinations (Llama-3.1-8B, Gemma-7B, Mistral-7B-v0.1, Llama-3.2-3B) demonstrate that, on average, over half of the information in 16-token sequences can be recovered from this compressed representation while maintaining strong semantic integrity and coherence. Furthermore, our analysis reveals an information bottleneck effect: longer sequences exhibit decreased token-level recovery while preserving strong semantic integrity. Besides, our framework also demonstrates robust generalization to out-of-distribution medical data.

1 Introduction

Large language models (LLMs) have achieved significant progress on a wide array of tasks. Despite their impressive performance, these models are often regarded as "black boxes," limiting our understanding of their internal working mechanisms. Consequently, a growing body of research aims to decode the information encoded within LLMs. These approaches vary widely, from training linear probes (Zou et al., 2023; Gurnee and Tegmark, 2025) or sparse autoencoders (SAEs) (Shu et al., 2025) to interpret specific features, to mapping internal representations directly to the vocabulary space through methods like Logit Lens (nostalgebraist, 2020) and Tuned Lens (Belrose et al., 2023). In this work, we focus on a distinct but related challenge: *representation decoding*, which aims to decode the internal representations of language models back into the full, original text.

Existing work on decoding model activations into text can be grouped into two major categories. The first category studies representation inversion from a safety perspective. They focus on inverting input tokens using both sentence and token embeddings (Devlin et al., 2019; Li et al., 2023; Morris et al., 2023; Huang et al., 2024b; Dong et al., 2025). The second line of research studies representation inversion from an interpretability perspective. These work interpret activations using in-context information, such as template-based prompts or original input, through either patching (Chen et al., 2024; Ghandeharioun et al., 2024) or guided sampling (Luo et al., 2025; Huang et al., 2024a). Although existing research has achieved preliminary results in representation inversion, significant challenges remain. First, most existing work investigates how information can be recovered from a sequence of token representations. In contrast, our interest is in decoding full text from *the last-token representation* from LLMs. Second, while existing work can recover text, it often does not answer the question of what information is encoded within a single last-token representation.

To address these challenges, we attempt to answer the following research question: To what extent can we invert the information from the last-token representation of an input sequence? Our overall goal is to explore single-token, activation-based input inversion. This is particularly challenging because the last-token representation is optimized for next-token prediction and can be regarded as an information bottleneck. Through quantitatively comparing the inverted text with the original input text, we aim to provide an understanding of what knowledge is preserved and encoded in the last-token representation of LLMs.

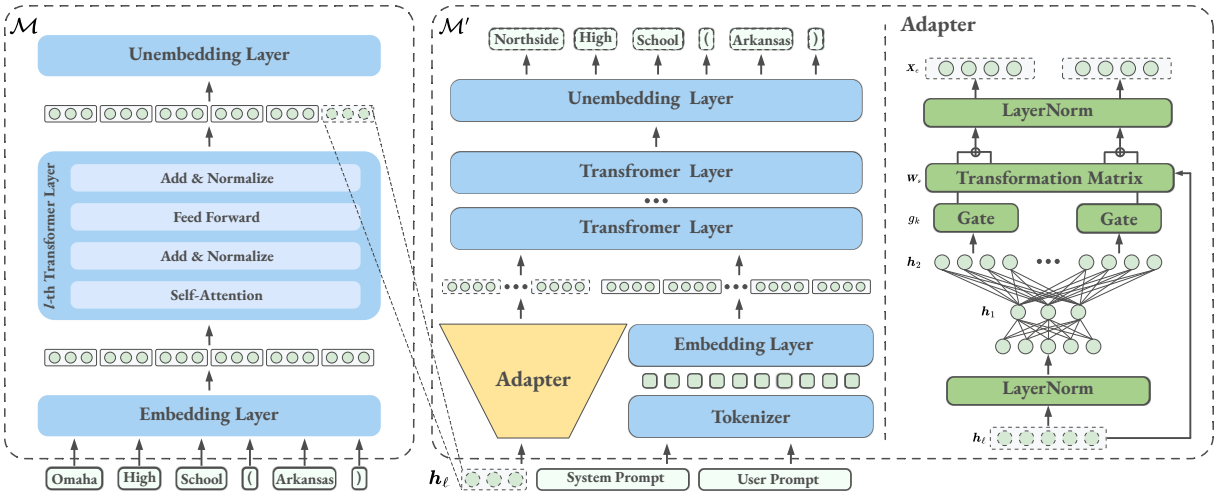


Figure 1: Overview of Rep2Text. The last-token representation obtained from the l-th layer of the target model $\mathcal{M}$ is projected into the embedding space of the decoding model $\mathcal{M}'$ via the adapter. The projected embeddings, together with those of the system and the user prompts, are then fed into the decoding model to reconstruct the corresponding text sequence.

Based on the above-mentioned motivation, in this work we propose **Rep2Text** (Representation to Text), a framework for decoding text from the last-token representations of LLMs as illustrated in Figure 1. Inspired by the design of large vision language models (LVLMs) such as LLaVA (Liu et al., 2023), Rep2Text trains a representation inverter which contains a decoding language model and an adapter. The adapter is used to map the input token representation from the target model into the token embedding space of the decoding language model, thereby aligning their latent spaces. These projected embeddings are subsequently passed into the decoding LLM, enabling it to interpret and generate text consistent with the original input sequence. Through comparing the inverted text against the original text, we quantify the information retained in the last-token representation.

Our experiments reveal that, remarkably, **over half of the original information in 16-token sequences can be recovered from a single last-token representation while maintaining strong semantic coherence**. This finding directly answers our central research question and demonstrates that, despite being optimized as an information bottleneck for next-token prediction, last-token representations retain a substantial amount of recoverable information about the input sequence. To validate the effectiveness of our approach, we combine established quantitative metrics with LLM-as-a-judge evaluations to measure information retention at token, structure, and semantic levels. Our results show that representations from different models exhibit varying recovery rates, revealing potential vulnerabilities in some models, while the recovery rate remains robust across decoding models of different sizes. Further analysis reveals that structural information is most prominent in early-to-middle layers, while semantic information becomes more pronounced in middle-to-late layers. We also find that recovery rate is strong for sequences shorter than 16 tokens but degrades for longer ones. Finally, evaluations on out-of-distribution clinical notes demonstrate promising generalizability, confirming that our method captures genuine representational properties rather than merely overfitting to in-distribution patterns.

2 Rep2Text Framework

In this section, we introduce the proposed *Rep2Text* framework (see Figure 1). Rep2Text employs a trainable adapter that bridges the target model's representation space to a decoding language model's embedding space, enabling us to systematically investigate what information is preserved in compressed last-token representations and how much of the original input can be recovered. The decoding LLM then autoregressively reconstructs the text from these projected embeddings.

2.1 Problem Statement

Given a layer-level representation from an LLM, our objective is to invert its ground-truth sequence as accurately as possible, thereby investigating the

extent to which the original input information is retained. Throughout this work, the terms *representation* and *activation* are used interchangeably to refer to the token-level hidden states extracted from different layers of decoder-only models.

Concretely, given a ground-truth sequence of n tokens $S = \langle s_1, \ldots, s_n \rangle$ and a *target model* $\mathcal{M}$ with L layers, we focus exclusively on the last-token representation. Let $\boldsymbol{h}^\ell$ denote the residual stream representation of the last token for input S at layer $\ell \in \{1, \ldots, L\}$ of model $\mathcal{M}$. We aim to decode $\boldsymbol{h}^\ell$ into an inverted text $\hat{S} = \langle \hat{s}_1, \ldots, \hat{s}_m \rangle$. Our goal is to quantify how much information is preserved within the bottleneck representation $\boldsymbol{h}^\ell$ by comparing the difference between the original input S and the reconstructed output $\hat{S}$.

2.2 Rep2Text Inverter Design

To invert the representation, we propose an *inverter* architecture inspired by the design of typical large vision-language models such as LLaVA. The inverter consists of two key components: (1) a trainable *adapter* that projects the target model's internal representation into the input token embedding space of the decoding language model, and (2) a *decoding language model* that generates the inverted text from these projected embeddings.

Specifically, we introduce a decoding model $\mathcal{M}'$ that can either be a copy of the target model $\mathcal{M}$ or a different LLM. To bridge the representation space of $\mathcal{M}$ and the embedding space of $\mathcal{M}'$, we train an adapter to project the token representation $\boldsymbol{h}^\ell \in \mathbb{R}^d$ from the target model $\mathcal{M}$ into the token embedding space of the decoding model $\mathcal{M}'$. The adapter is implemented as a two-layer MLP with gated skip connection with optional projection, defined as:

$$\begin{aligned} \boldsymbol{h}_1 &= \mathrm{GELU}(\boldsymbol{W}_1 \cdot \mathrm{LN}(\boldsymbol{h}_\ell) + \boldsymbol{b}_1), \\ \boldsymbol{h}_2 &= \boldsymbol{W}_2 \cdot \boldsymbol{h}_1 + \boldsymbol{b}_2, \\ \boldsymbol{X}_e &= \mathrm{LN}(\boldsymbol{W}_s \cdot \boldsymbol{h}_\ell + g_k \cdot \boldsymbol{h}_2), \end{aligned} \tag{1}$$

where $\mathrm{LN}(\cdot)$ and $\mathrm{GELU}(\cdot)$ represent the norm layer and activation function respectively. $\boldsymbol{W}_1 \in \mathbb{R}^{d \times d^{\mathrm{hid}}}$ and $\boldsymbol{W}_2 \in \mathbb{R}^{d^{\mathrm{hid}} \times k \cdot d'}$ refer to linear transformations in the first and second layers respectively, where d and d' represent the hidden dimensions of the target model and decoding model. Note that we set $d^{\mathrm{hid}} = f \cdot d$, where f is an expansion factor. $\boldsymbol{W}_s \in \mathbb{R}^{d \times d'}$ denotes the transformation matrix of the skip connection. When $d = d'$, $\boldsymbol{W}_s$ is an identity matrix enabling a true residual connection; when $d \neq d'$, $\boldsymbol{W}_s$ serves as a learned projection matrix to match dimensions. $\boldsymbol{h}_2 \in \mathbb{R}^{k \cdot d'}$ is reshaped into (k, d'), which can be regarded as k token embeddings. Each token embedding is constructed with a gated combination of the skip path and the MLP-transformed path to preserve the representation information as much as possible. The projected token embedding can be denoted as $\boldsymbol{X}_e = [\boldsymbol{x}_1; \cdots; \boldsymbol{x}_k]$, where the number k of projected tokens is a hyperparameter.

For each representation, the projected token embedding $\boldsymbol{X}_e$ is combined with system prompt embedding $\boldsymbol{X}_{\mathrm{sys}}$ and user prompt embedding $\boldsymbol{X}_u$. The combined sequence $[\boldsymbol{X}_e; \boldsymbol{X}_{\mathrm{sys}}; \boldsymbol{X}_u]$ is fed into the first layer of decoding model $\mathcal{M}'$ (after its embedding layer), bypassing the embedding layer. The decoding model then autoregressively generates the inverted text $\hat{S}$.

2.3 Rep2Text Inverter Training

For an target sequence of length L at step t, the inverter predicts its probability conditioned on text embedding and all previous predicted tokens. The joint probability of inverted sequence $\hat{S}$ is:

$$\begin{aligned} &p\left(\hat{S} \mid \boldsymbol{X}_e, \boldsymbol{X}_{\mathrm{sys}}, \boldsymbol{X}_u\right) \\ &\quad = \prod_{t=1}^{T} p_\theta\left(s_t \mid \boldsymbol{X}_e, \boldsymbol{X}_{\mathrm{sys}}, \boldsymbol{X}_u, \hat{S}_{<t}\right), \end{aligned} \tag{2}$$

where $\hat{S}_{<t}$ are the inverted tokens generated before step t. θ is the trainable parameters. In our paper, we consider two training schemes: (1) *adapter-only fine-tuning*, where only the adapter parameters are optimized; (2) *joint fine-tuning*, where the adapter is first fine-tuned independently and then the adapter is fully fine-tuned and the base model is updated via LoRA (Hu et al., 2022). Accordingly, θ refers to the trainable parameters under the chosen scheme.

During training, we employ teacher forcing to maximize the log-likelihood of ground-truth tokens. To stabilize training, we utilize label smoothing to soften the one-hot target distribution. The ground-truth token vocabulary distribution is denoted as

$$q_t(v_i) = (1 - \epsilon)\mathbf{1}\left[v_i = s_t\right] + \frac{\epsilon}{|V|}, \tag{3}$$

where $\mathbf{1}(\cdot)$ is an indictor function that equals 1 if the condition holds and 0 otherwise, and ϵ is the label smoothing factor, set to be 0.075. The training objective is the smoothed cross-entropy loss that defined as:

$$\mathcal{L}_t = -\sum_{i=1}^{|V|} q_t(v_i) \log p_\theta(v_i \mid \mathbf{X}_e, \mathbf{X}_{\text{sys}}, \mathbf{X}_u, S_{<t}) \,, \quad (4)$$
$$\mathcal{L}_{\text{LS}} = \frac{1}{T}\sum_{t=1}^{T} \mathcal{L}_t.$$

This training objective optimizes the adapter (and optionally the decoding model via LoRA) to minimize the prediction error across all token positions in the inverted sequence. A label smoothing term is incorporated to prevent overconfidence in token predictions, thereby improving generalization to unseen representations. Through this training process, the adapter learns to effectively map the compressed last-token representation from the target model's latent space into the decoding model's token embedding space, enabling the reconstruction of the original input sequence.

3 Experiments

In this section, we evaluate our framework on inverting ground-truth sequences under different combinations of target and decoding models (§3.2). To analyze information bottleneck within last-token representations, we investigate the inversion performance across varying lengths of ground-truth sequences (§3.3). We further invert representations from different layers of the target model to investigate how encoded information evolves throughout the network (§3.4). Finally, to demonstrate the effectiveness and generalizability of our approach, we conduct experiments on inverting representation on out-of-distribution datasets (§3.5).

3.1 Experimental Setup

Datasets. Our adapters are trained on passages randomly truncated from Wikipedia articles contained in *The Pile* (Gao et al., 2020). Each truncated sequence consists of n tokens without overlap, where $n \in \{8, 16, 32, 64\}$ depending on the experimental configurations. Each training sample comprises a data pair consisting of the last-token representation of ground-truth sequence from a fixed layer of the target model and its corresponding ground-truth sequence itself. For adapter finetuning, we use a dataset containing 640K sequences. During the full fine-tuning stage, the training dataset incorporates an additional 960K sequences. For evaluation, we randomly sample 1000 sequences as the test set and evaluate the inverted outputs using a combination of quantitative metrics and LLM-as-a-judge assessments.

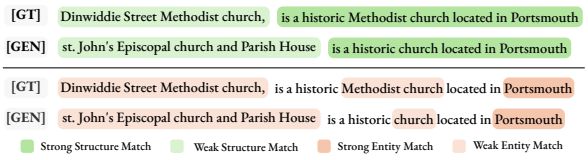


Figure 2: Examples of structure and entity similarity. Darker colors indicate higher similarity score.

Models. Our paper uses Llama-3.1-8B as both the target and decoding models throughout our experiments except in §3.2, where Gemma-7B (Google, 2024), Mistral-7B-v0.1 (AI, 2023), and Llama-3.2-3B (AI, 2025) are used as target models while keeping Llama-3.1-8B as the inverter, to study the feasibility and effectiveness of cross-model decoding. To examine the scaling behavior of representation inversion, we additionally evaluate Llama-3.2-3B (AI, 2025) as a smaller decoding model to illustrate if larger decoding models yield improved inversion performance.

Implementation Details. Most of our adapter-only fine-tuning experiments are conducted on sequences of length $n = 16$, which can be accommodated on 2 NVIDIA A100 GPUs and require approximately 7 hours to complete. The corresponding full fine-tuning experiments under the same setting take an additional 12 hours using the same hardware. During adapter-only fine-tuning, we use a learning rate of 1e-3. For full fine-tuning, we apply a learning rate of 5e-4 for the adapter and 2e-4 for the LoRA parameters. Both training schemes are trained for 3 epoches, as further training yields diminishing improvements. Additional implementation details are included in the Appendix C. *In all experiments except those in §D.3, we fine-tune only the adapters while keeping the decoding model frozen,* to demonstrate that the desired alignments between the hidden representation space and the embedding space are primarily achieved by adapter fine-tuning alone, rather than model overfitting.

Evaluation Measurements. The evaluation of inverted sequences considers three key aspects of fidelity to the ground truth: token-level accuracy, sentence structure and entities preservation, and semantic similarity.

- *Token-level Accuracy.* We adopt ROUGE scores to measure the token-level accuracy (Lin, 2004). Specifically, *ROUGE-1*, and *ROUGE-2* measure the recovery rate of indi-

Table 1: Performance Comparison of Inverting Representation from Different Models Using Llama-3.1-8B.

Target Model	Decoding Model	ROUGE-1 (0-1)↑	ROUGE-2 (0-1)↑	ROUGE-L (0-1)↑	BERTScore (0-1)↑	Structure (0-1)↑	Entity (0-1)↑	Topic (0-1)↑
Gemma-7B	Llama-3.1-8B	$0.49_{\pm 0.23}$	$0.27_{\pm 0.24}$	$0.47_{\pm 0.23}$	$0.74_{\pm 0.14}$	$0.66_{\pm 0.23}$	$0.60_{\pm 0.27}$	$0.78_{\pm 0.25}$
Mistral-7B-v0.1	Llama-3.1-8B	$0.52_{\pm 0.24}$	$0.35_{\pm 0.28}$	$0.51_{\pm 0.23}$	$0.81_{\pm 0.11}$	$0.70_{\pm 0.22}$	$0.76_{\pm 0.23}$	$0.91_{\pm 0.15}$
Llama-3.1-8B	Llama-3.1-8B	$0.48_{\pm 0.23}$	$0.28_{\pm 0.24}$	$0.47_{\pm 0.22}$	$0.78_{\pm 0.11}$	$0.66_{\pm 0.22}$	$0.74_{\pm 0.23}$	$0.91_{\pm 0.14}$
Llama-3.2-3B	Llama-3.1-8B	$0.46_{\pm 0.22}$	$0.24_{\pm 0.22}$	$0.44_{\pm 0.22}$	$0.77_{\pm 0.11}$	$0.64_{\pm 0.22}$	$0.71_{\pm 0.23}$	$0.88_{\pm 0.16}$
Mistral-7B-v0.1	Llama-3.2-3B	$0.52_{\pm 0.25}$	$0.33_{\pm 0.27}$	$0.51_{\pm 0.23}$	$0.80_{\pm 0.12}$	$0.69_{\pm 0.22}$	$0.74_{\pm 0.24}$	$0.91_{\pm 0.15}$
Llama-3.2-3B	Llama-3.2-3B	$0.46_{\pm 0.22}$	$0.26_{\pm 0.23}$	$0.45_{\pm 0.21}$	$0.76_{\pm 0.11}$	$0.64_{\pm 0.22}$	$0.69_{\pm 0.24}$	$0.88_{\pm 0.16}$

vidual tokens and 2-grams, respectively, while *ROUGE-L* captures the longest common subsequence between the ground-truth and the inverted sequence. Detailed definitions of these metrics are provided in the Appendix D.

- *Sentence Structure and Entity Preservation.* To evaluate the preservation of sentence structure and entities, we use GPT-4.1-mini to rate the degree of preservation ranging on a 0-5 scale (normalized to 0-1), yielding the *Structure Score* and *Entity Score*, respectively. The structure score assesses how well the grammatical structure and sentence skeleton are preserved in the inverted sequences as shown in Figure 2. While the entity score measures how accurately entity names and their associated attributes are inverted. Detailed rating criteria are provided in Appendix F.

- *Semantic Similarity.* We use BERTScore F1 and LLM-as-a-judge to collectively evaluate the semantic similarity between the ground-truth and inverted sequences (Zhang et al., 2020). BERTScore quantifies similarity in the embedding space, whereas the LLM-based evaluation measures topic relevance between ground-truth and inverted sequences. The scoring guidelines for LLM-as-a-judge evaluation are included in Appendix F.

3.2 Inversion Results Analysis

As defined in §2.2, the inverter's base model can be either a copy of the target model or a different model. To demonstrate the feasibility of both configurations, we specifically choose Gemma-7B, Mistral-7B-v0.1, Llama-3.2-3B, and Llama-3.1-8B as target models. For the decoding model, we use Llama-3.1-8B and Llama-3.2-3B to interpret representation of 16-token sequences from 10th layer of each target model. The results are summarized in Table 1. We derive the following observations.

First, using only the last-token representation, our models recover approximately half of the original tokens on average, as measured by ROUGE-1. For bi-grams, the average ROUGE-2 recover rate reaches about 24%. The average ROUGE-L score shows that the longest common subsequences typically retain at least half of the original content, aligning with ROUGE-1. Moreover, the structure score exceeds 64% on average, indicating that the syntactic structure of ground-truth sentences is largely preserved during inversion. In terms of entity recovery, the lower bound reaches 60%, and preliminary qualitative inspection further reveals that entities are often recovered in a semantically consistent manner, even when not exact lexical matches. Finally, both BERTScore and topic score demonstrate that the inverted sentences maintain strong semantic relevance to original text. Thus, the model recovers **over half of the original content while maintaining strong semantic coherence and integrity**.

We use Llama-3.1-8B as the decoding model to invert representations from Gemma-7B, Mistral-7B-v0.1, and Llama-3.2-3B, respectively. As shown in Table 1, the inversion performance is highest for Mistral-7B-v0.1, whose scores consistently surpass those of Gemma-7B and Llama-3.2-3B across all metrics. This indicates that **representations from different models exhibit varying levels of invertibility**. In particular, Mistral-7B-v0.1 encodes information that is more recoverable at the token, structural and topical levels, indicating stronger representational richness but also greater susceptibility to information leakage and privacy risks. To further substantiate this finding, we employ Llama-3.1-3B as the inverter to decode representations from Mistral-7B-v0.1 and observe

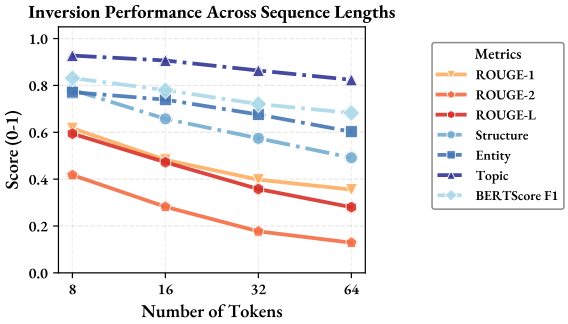


Figure 3: Performance comparison of inverting varying length of sequences.

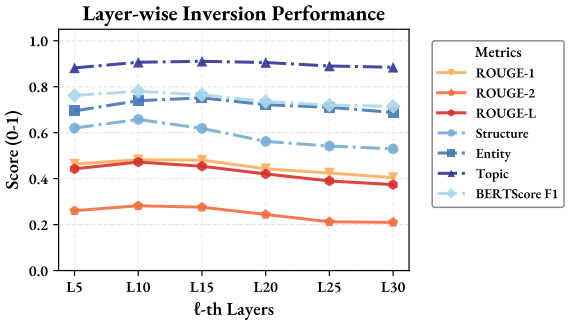


Figure 4: Performance comparison with layerwise representation inversions.

comparable performance across most evaluation dimensions. This confirms that representations from Mistral-7B-v0.1 consistently yield higher recoverability regardless of which decoding model is used.

To investigate the discrepancy between self-inversion and cross-model inversion, we further decode representations from Llama-3.2-3B using both itself and Llama-3.1-8B. The results show that **inversion performance remains robust across different decoding models**. A similar trend is observed when inverting representations from Mistral-7B-v0.1 with Llama-3.1-8B and Llama-3.2-3B, where all metrics remain nearly identical. Overall, using the same model as both the target model and decoding model does not necessarily guarantee superior inversion performance, and larger decoding models provides only marginal improvement in entity recovery. Moreover, the effectiveness of cross-model inversion lends support to the platonic representation hypothesis (Huh et al., 2024), which posits that as models scale, their internal representations converge toward a shared, aligned structure.

3.3 Inverting Sequence of Varying Length

Understanding how much information can be effectively inverted from the last-token representation is crucial for analyzing the information bottleneck in LLMs. Intuitively, we investigate the performance on inverting sequences with different number of tokens. Here, we train separate adapters to invert last-token representations of sequences with 8, 16, 32, 64 tokens at the 10th layer respectively. As demonstrated in §3.2, the inversion performance remains robust across different decoding models. Therefore, our subsequent experiments adopt the configuration where the same model, i.e. Llama-3.1-8B, is used as both target and decoding model.

We report the inversion results across varying sequence lengths in Figure 3. **A clear degradation in inversion performance is observed as the sequence length increases.** For example, the ROUGE-1 score drops from approximately 0.6 for 8-token sequence to around 0.3 for 64-token sequences, indicating that the inverter struggles to precisely reconstruct longer sequences. Similarly, the structure score decreases aggressively with the inverted sequence length, suggesting that recovering the grammatical skeleton of the original sentence becomes increasingly difficult when relying solely on the last-token representation. When it comes to the topic-level evaluations, including BERTScore, entity recovery, and topic score, the decline in performance is much less pronounced. The difference between the topic-level scores for 64-token and 8-token sequences remains within 20%. Therefore, as the inverted length increases, the inverter continues to capture the overall topic consistently, though it fails to recover much of the detailed content exactly. Detailed inverted sequences are presented in Appendix A.

3.4 Decoding Representation across Layers

Prior work show that middle-to-deep layers capture high-level semantics more effectively (Jin et al., 2025; Campbell et al., 2023). To further investigate how decoded information evolves across layers and to identify the optimal layer for inversion, we train adapters on representations extracted from different depths of Llama-3.1-8B using 16-token input sequences. Specifically, we select the 5th, 10th, 15th, 20th, 25th, and 30th layers for comparison.

The results in Figure 4 show that token-level metrics, structure score and BERTScore reach their highest values at the 10th layer. Since ROUGE-L captures the longest common subsequence and the structure score reflects grammatical integrity, both

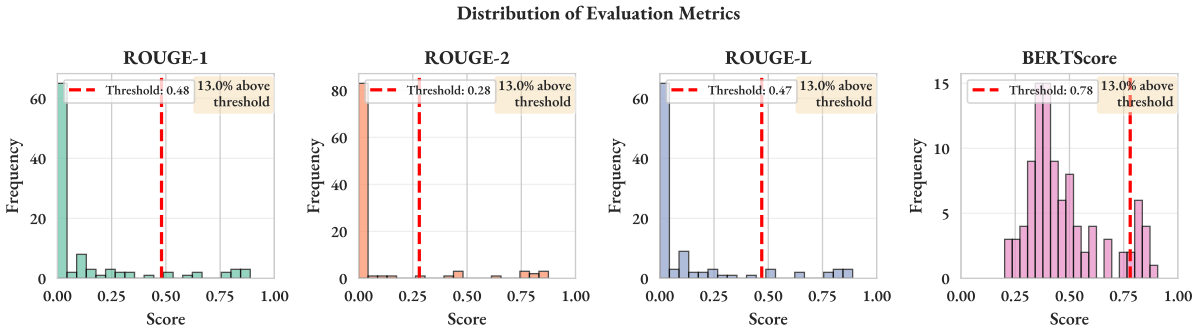


Figure 5: The score distribution on OOD clinical notes. The mean score obtained by Llama-3.1-8B wen used as both the target and decoding model (Table 1), serves as the threshold for assessing the inverter's capability to recover OOD data.

Table 2: Inverted examples with OOD clinical notes

Ground-truth Sequence	Inverted Sequence	R1	R2	RL	BS	SS	ES	TS
A. Rogers, 71, admitted for pneumonia, stable vitals.	://\n A. Rogers, 71, admitted for pneumonia, stable vitals. ://\nA. Rogers,	0.89	0.88	0.89	0.85	2	5	5
S. Evans, 59, COPD flare, on 2L oxygen	, 2/3/2018\n S. Evans, 59, COPD flare, on 2L	0.78	0.75	0.78	0.8	2	2	5
Diabetic foot ulcer dressed daily.	2. Diabetic foot ulcer dresseddaily . 3. Diabetic foot ulcer dressed daily. 4. Di	0.53	0.47	0.53	0.8	2	5	5
Karen J., 41, migraine recurrent, prescribed sumatriptan.	://www.medscape.com/viewarticle/509975\n Kaplan, M., 41, migraine recurrent	0.35	0.27	0.35	0.55	1	2	3
80yo M, dementia, agitated overnight.	://www.medscape.com/viewarticle/812201\n A 70-year-old man with a history of dementia	0.1	0.1	0.1	0.48	2	4	5
Pt denies pain, tolerating diet.	of 2.5 kg. 2.5 kg. 2.5 kg. 2.5 kg	0	0	0	0.35	0	0	0

metrics consistently indicate that syntactic information is best encoded around the 10th layer. In contrast, ROUGE-1 and entity recovery peak at the 15th layer, suggesting that lexical and entity-level details are better represented around the middle layer. The topic score hypes from the 10th layer to the 20th layer with only minor fluctuations, implying that high-level semantic information is progressively consolidated in later layers. In summary, the **early-to-middle layers (around L10-L15) contain the most concrete and literally recoverable representations** of the input's structure and lexical content. As information propagates to later layers, these specific details become increasingly abstracted and compressed, making them harder to invert, while the core, high-level topic is preserved and consolidated.

3.5 OOD Samples Inversion Analysis

To further evaluate the inversion performance of our approach and its applicability in interpreting representations, we apply the trained adapters on inverted out-of-distribution (OOD) data and assess their performance across all evaluation dimensions. In addition, we engage graduate students to manually rate the generated sentences according to the same instructions provided to GPT-4.1-mini, which allows for a direct comparison between human judgments and LLM-as-a-judge evaluations.

As shown in previous experiments, inversion performance declines as the sequence length increases. The results indicate that 16-token sequences yield promising inversion quality. Therefore, we select 16-token sequences as our test data. Since our adapters are trained on data from Wikipedia, which covers diverse factual knowledge including persons, events, and locations. We further evaluate the generalization of the inverter on unseen data from the medical domain. We prompt GPT-4.1-mini to generate 100 sequences resembling clinical notes following the instructions in Appendix E. Each sequence contains up to 16 tokens and includes

elements like patient names, admission dates or birth dates, and reported symptoms or treatments. We then apply the adapters trained on 16-token sequences using Llama-3.1-8B as the inverter.

The evaluation results for token-level recovery and BERTScore are shown in Figure 5. We use the mean score of each metric on the Wikipedia test set in Table 1 as a reference threshold. Notably, approximately 13% of the inverted clinical notes exceed the in-distribution average performance, suggesting partial generalization of the learned inversion capability to out-of-distribution data. Although the clinical data are unseen during training, some examples are still inverted with striking fidelity. Representative examples of inverted outputs at different performance levels are presented in Table 2. To provide a comprehensive view of inversion quality, we include examples inverted with varying ROUGE-1 scores. Here, R1, R2, RL, BS, SS, ES, and TS denote ROUGE-1, ROUGE-2, ROUGE-L, BERTScore, Structure Score, Entity Score, and Topic Score, respectively. The human sanity check results are presented in Appendix B.

Due to the lack of explicit alignment with medical text during training, the inverter occasionally generates introductory medical website content before the actual note. To accommodate this, we set the number of generated tokens to 24. As shown in Table 2, when ROUGE-1 score exceeds 0.5, most examples recover the main context accurately, otherwise core entities such as year, disease name can often still be recovered. However, some samples fail to recover critical information because of domain misalignment. **Overall, the inversion results on OOD clinical notes demonstrate the generalizability of our trained adapters in recovering meaningful information from unseen data.**

4 Related Work

Embedding Inversion. The inversion is typically formulated as an optimization problem in which the attack model attempts to generate hypotheses that produce embeddings as close as possible to the target embeddings. A few work attempt to recover ground-truth sequences using sentence embeddings from BERT models. Song and Raghunathan (2020) inverse the sentence embeddings into bag of words. Some work attempt to train an attacker model to decode the ground-truth sequence utilizing sentence embeddings and text embeddings (Li et al., 2023; Morris et al., 2023; Huang et al., 2024b). Further, Dong et al. (2025) extends embedding inversion to LLM's internal states at a certain layer, by learning token embeddings that can produce similar internal states. However, to fully recover the input text, these papers either rely heavily on iterative optimization or incorporating all sentence embeddings and token embeddings.

Activation Decoding. Some work seek to decode activations into natural language. Recent work such as SelfIE (Chen et al., 2024), and Patchscopes (Ghandeharioun et al., 2024) interpret representations through patching them into the forward pass of LLMs to decode natural language explanations. Besides, LIT (Pan et al., 2024) finetunes target model to answer questions related to given activations patched within. PCD (Huang et al., 2025) maps activations into concept vectors then train a model to answer questions about activations. Some work explains activations by assuming that similar meanings produce similar activations. They invert an activation to find inputs that would recreate it, and use those inputs as the explanation. InverseView (Huang et al., 2024a) trains a decoder to sample the input distribution for a given activation. InverseScope (Luo et al., 2025) explore task-specific features encoded in the input distribution. In contrast, we rely solely on the last-token representations, without any in-context templates or original inputs as auxiliary hints, to recover the information encoded within these representations.

5 Conclusions

In this work, we explore the research question that to what extent can the original input text be recovered from a single last-token representation of LLMs? To answer this question, we proposed Rep2Text, a novel framework that employs a trainable adapter to project a target model's last-token representation into the embedding space of a decoding language model, which then autoregressively reconstructs the input text. Our comprehensive evaluations indicate that over half of the information in 16-token sequences can be recovered from the compressed last-token representation while maintaining strong semantic integrity. Besides, experimental results show longer sequences lead to decreased inversion performance, with reliable recovery achieved for sequences under 16 tokens. Additionally, Rep2Text demonstrated promising generalization, successfully recovering information from out-of-distribution clinical data.

Limitations

Our Rep2Text framework has several limitations. First, we evaluate LLMs with parameters at most of 8B. In future work, we plan to evaluate on larger LLMs with tens or hundreds of billions of parameters to better understand how our method scales with model size and complexity. Second, due to the limited training resources available, we only use a sampled dataset from The Pile to fine-tune the inverter. Third, when applied to out-of-distribution medical notes, the inverter can suffer from domain misalignment, occasionally generating irrelevant medical website introductory text or failing to recover critical information.

References

Meta AI. 2025. Llama-3.2-3b. https://huggingface.co/meta-llama/Llama-3.2-3B. [Accessed 29-10-2025].

Mistral AI. 2023. Mistral-7b-v0.1. https://huggingface.co/mistralai/Mistral-7B-v0.1. [Accessed 29-10-2025].

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*.

James Campbell, Richard Ren, and Phillip Guo. 2023. Localizing lying in llama: Understanding instructed dishonesty on true-false questions through prompting, probing, and patching. *arXiv preprint arXiv:2311.15131*.

Haozhe Chen, Carl Vondrick, and Chengzhi Mao. 2024. Selfie: self-interpretation of large language model embeddings. In *Proceedings of the 41st International Conference on Machine Learning*, pages 7373–7388.

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.

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

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, and 1 others. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. 2024. Patchscopes: a unifying framework for inspecting hidden representations of language models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 15466–15490.

Google. 2024. Gemma-7b. https://huggingface.co/google/gemma-7b. [Accessed 29-10-2025].

Wes Gurnee and Max Tegmark. 2025. Language models represent space and time. In *The Twelfth International Conference on Learning Representations*.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Vincent Huang, Dami Choi, Daniel D Johnson, Sarah Schwettmann, and Jacob Steinhardt. 2025. Predictive concept decoders: Training scalable end-to-end interpretability assistants. *arXiv preprint arXiv:2512.15712*.

Xinting Huang, Madhur Panwar, Navin Goyal, and Michael Hahn. 2024a. Inversionview: a general-purpose method for reading information from neural activations. *Advances in Neural Information Processing Systems*, 37:137903–137964.

Yu-Hsiang Huang, Yuche Tsai, Hsiang Hsiao, Hong-Yi Lin, and Shou-De Lin. 2024b. Transferable embedding inversion attack: Uncovering privacy risks in text embeddings without model queries. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4193–4205.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. 2024. The platonic representation hypothesis. *The Forty-First International Conference on Machine Learning (ICML)*.

Mingyu Jin, Qinkai Yu, Jingyuan Huang, Qingcheng Zeng, Zhenting Wang, Wenyue Hua, Haiyan Zhao, Kai Mei, Yanda Meng, Kaize Ding, and 1 others. 2025. Exploring concept depth: How large language models acquire knowledge and concept at different layers? *The 31st International Conference on Computational Linguistics (COLING 2025)*.

Haoran Li, Mingshi Xu, and Yangqiu Song. 2023. Sentence embedding leaks more information than you expect: Generative embedding inversion attack to recover the whole sentence. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 14022–14040.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Yifan Luo, Zhennan Zhou, and Bin Dong. 2025. Inversescope: Scalable activation inversion for interpreting large language models. *arXiv preprint arXiv:2506.07406*.

John Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander M Rush. 2023. Text embeddings reveal (almost) as much as text. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12448–12460.

nostalgebraist. 2020. interpreting gpt: the logit lens. LessWrong.

Alexander Pan, Lijie Chen, and Jacob Steinhardt. 2024. Latentqa: Teaching llms to decode activations into natural language. *arXiv preprint arXiv:2412.08686*.

Dong Shu, Xuansheng Wu, Haiyan Zhao, Daking Rai, Ziyu Yao, Ninghao Liu, and Mengnan Du. 2025. A survey on sparse autoencoders: Interpreting the internal mechanisms of large language models. *EMNLP Findings*.

Congzheng Song and Ananth Raghunathan. 2020. Information leakage in embedding models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pages 377–390.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *The International Conference on Learning Representations (ICLR)*.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, and 1 others. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

A Inverted Examples with Varying Token Lengths and Recovery Rate

As shown in Table 4, for sequences with 8 tokens and 16 tokens, some inverted sequences fully recover the original text, while others fail to capture fine-grained details yet still preserve clear grammatical structures. For example, when inverting 16-token sequence, although the ROUGE-1 score is only 0.4, the original sentence "*Rob James may refer to:\n\nRob James (singer) (*" and the inverted sequence "*Mark Jones\n\nMark Jones may refer to:\n\nMark Jones (singer) (*" share the same syntactic pattern, "[NAME]\n\n[NAME] (singer) (", and convey equivalent topic-level information. However, when the number of tokens exceed 16, the inverted sequences remain highly topic-relevant but tend to lose their global structural coherence, even when achieving a reasonable ROUGE-1 score.

B Human Sanity Check

To assess the reliability of LLM-as-a-judge evaluations, we conducted a human sanity check by asking a graduate student to rate the inverted sequences of out-of-distribution (OOD) clinical notes following the same instructions provided to GPT-4.1-mini. The results are presented in Table 3. For convenience, all scores were normalized from a 0-5 scale to a 0-1 range. As shown in the table, the LLM tends to assign lower scores for sentence structure but higher ones for topic relevance, while the entity scores are closely aligned. Overall, the discrepancy between the LLM and human ratings remains within a reasonable range.

Table 3: Comparison of mean semantic scores between human and GPT-4.1-mini

	Structure	Entity	Topic
GPT-4.1-mini	0.12	0.28	0.57
Student	0.23	0.29	0.36

C More Implementation Details

Throughout our experiments, we use two NVIDIA A100 GPUs (80 GB each) for training, except for the experiment on inverting 64-token sequences, where four A100 GPUs are utilized. Most training runs are completed within seven hours. For the ablation studies on the number of projected token vectors, the number of projected tokens varies according to the experimental settings. In all other cases, the number of projected tokens is kept identical to the number of tokens being inverted, which has been shown to yield optimal performance (see Appendix D.3). In addition, the hidden expansion factor is fixed at 0.5 across all experiments, except in the ablation studies analyzing the relationship between inversion performance and hidden expansion factors.

During training, the batch size is fixed at 1028, and all experiments are conducted for three epochs. The warmup ratio for adapter-only fine-tuning is set to 0.15. We adopt a cosine learning rate scheduler. The dropout rate after the first layer of the adapter is fixed at 0.1, the weight decay is set to 0.01, and the label smoothing factor is fixed at 0.075.

D Definition of ROUGE Score

D.1 ROUGE-1 and ROUGE-2

For ROUGE-k, it computes the F-measure of k-grams extracted from a sequence. Suppose the set of k-grams is denoted as $G_k(S)$ and $G_k(\hat{k})$ for ground-truth sequence and inverted sequence respectively. ROUGE-k is computed as follows:

$$\begin{gathered}\text{Overlap}_k = \sum_{g \in G_k(S) \cap G_k(\hat{S})} \min\left(\text{cnt}_S(g), \text{cnt}_{\hat{S}}(g)\right) \\ R_k = \frac{\text{Overlap}_k}{|G_k(S)|}, \quad P_k = \frac{\text{Overlap}_k}{|G_k(\hat{S})|} \\ \text{ROUGE}_k = \frac{(1+\beta^2) R_k P_k}{R_k + \beta^2 P_k}\end{gathered} \tag{5}$$

where $\text{cnt}(\cdot)$ denotes the count of the set.

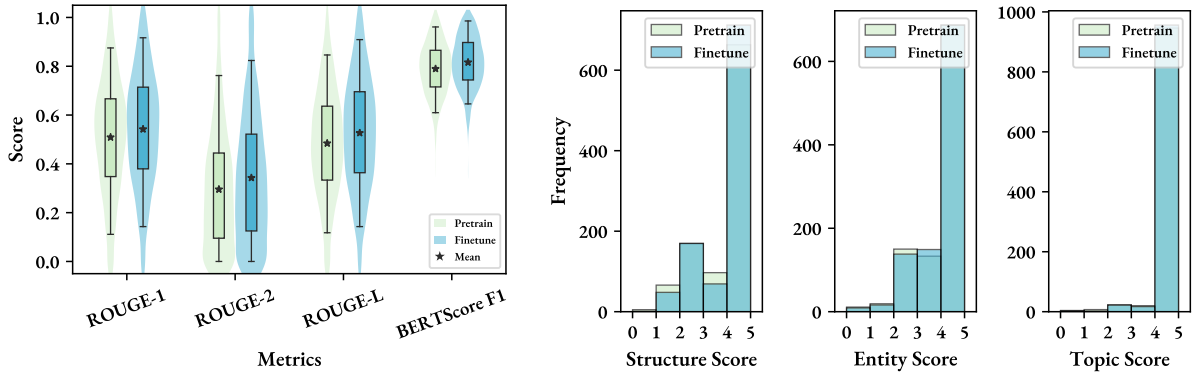
D.2 ROUGE-L

Give an ground-truth sequence $S = \langle s_1, \ldots, s_n \rangle$ and an inverted sequence $\hat{S} = \langle \hat{s}_1, \ldots, \hat{s}_m \rangle$, the length of their longest common subsequence is $LCS(S, \hat{S})$. The ROUGE-L score F_{LCS} is defined as follows:

$$\begin{gathered}R_{LCS} = \frac{LCS(S,\hat{S})}{m}, \quad P_{LCS} = \frac{LCS(S,\hat{S})}{n} \\ F_{LCS} = \frac{(1+\beta^2) R_{LCS} P_{LCS}}{R_{LCS} + \beta^2 P_{LCS}}\end{gathered} \tag{6}$$

D.3 Ablation Study

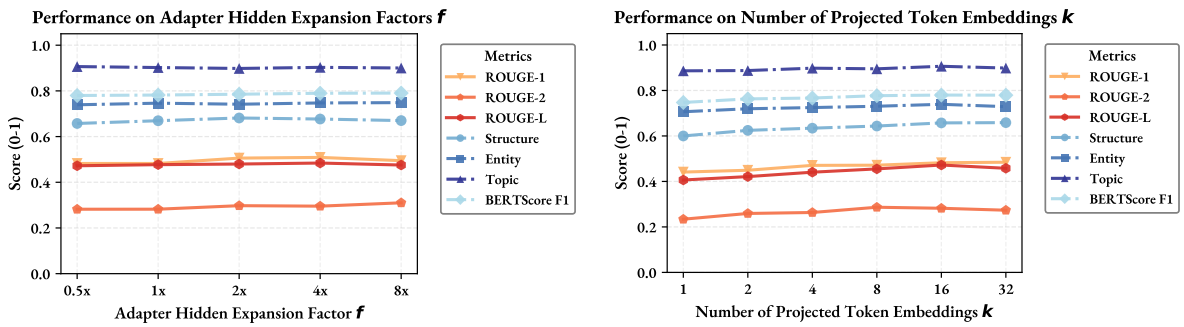
In this subsection, we study training strategies to improve the inverter. First, we compare two training schemes: pretraining and finetuning, and observe a small gain from finetuning. We then analyze how adapter design affects performance by varying the hidden and output dimensions of its two-layer MLP.



(a) Score distribution

(b) Score Frequency

Figure 6: Pretrain vs Finetune performance comparison



(a)

(b)

Figure 7: Inversion Performance on varying expansion factors

Two Training Schemes. We first study the performance of inverters under two training schemes to gain knowledge on when representation space and token embedding space are aligned. One training scheme only finetunes the adapters, the other continually finetunes the adapter and decoding language model together after finetuning the adapter, where the adapter is finetuned fully and decoding language model is finetuned using LoRA.

We train an adapter to invert 16-tokens input sequences first for pretraining scheme. Based on the trained adapter, we further finetune both the adapter and inverter's base model. In Figure 6a, whiskers represent 5% to 95% respectively. Among token-level scores, finetune is better than pretrain only. For semantic-level scores, finetune can slightly. improve the structure recovery. But for entity and topic, BERTScore, entity, and topic score almost make even. Consequently, finetune can slightly help improve alignment but most alignments are reached during pretraining stage. In our following experiments, we focus on pretraining scheme only to examine inverter's performance.

Hidden Expansion Factor. Ablation experiments on output dimensions of adapters have been done separately with both layers of adapters. We train adapters on inverting 16-token sequences at 10th layer of Llama-3.1-8B. As defined in §2.2, for the 1st layer of the adapter, the output dimension $d^{hid} = f \cdot d$ is scaled by f according to input dimension. To examine how the variation of f influence the inversion performance, we sample f to be 0.5, 1, 2, 4, 8 and fix the output dimension of the 2nd layer. As show in Figure 7a, the inversion performance is robust to variations of hidden dimensions between 1st and 2nd layer. Semantic-level recovery is barely impacted by variations, while token-level accuracy (ROUGE-1) improves slightly as hidden dimensions scale up. This could attribute to better memory ability as the adapter scales up.

For the second layer of the adapter, we set the output dimensions to be $k \cdot d'$ where d' is the hidden dimensions of the inverter model. We hope the projected token embeddings capture as much

information as possible from the last-token representation. Under 16-token sequence inversion task, we attempt to set k to be 1, 2, 4, 8, 16, 32 to study whether more projected token can help extract more information from the token representation. As illustrated in Figure 7b, the inversion performance on both token-level and semantic-level improves with the scaling up of k. However, when $k = 32$ exceeds the number of inverted tokens, the inversion performance degrades compared to $k = 16$. This could come from repetitive information introduced by additional token embeddings.

Overall, the first-layer expansion factor has little effect on inversion performance. Performance is best when the number of projected token embeddings matches the ground-truth token length. Based on this, we set the expansion factor to 0.5 and use the ground-truth token count for the projected embeddings in the next section.

E Instructions for Clinical Notes

Generate 100 short sentences about fictional patients, mimicking the style of the MIMIC-III clinical dataset. Each sentence should describe a patient and mention at least one of the following:

- name (any first name or initial)
- disease or diagnosis
- symptom(s)
- age, date of birth, or admission date (optional)

Requirements:

- Each sentence $\leq$ 12 words
- Diverse grammatical forms: clinical notes, summaries, shorthand/abbreviations, fragments, clauses, snippets
- Realistic medical language, no real patient data
- Randomized structure, tone, and detail level

Example styles:

- "John D., 56, admitted 04-12-2009 for chest pain."
- "Pt c/o dizziness and nausea; hx of diabetes."
- "Admitted 02/17/10 — fever, hypotension, suspected sepsis."
- "Mary T, DOB 1982-11-09. COPD exacerbation. Stable overnight."
- "Elderly male, HTN, SOB on exertion."

Output Exactly 100 unique sentences.

F Scoring Prompts

We include prompts for GPT-4.1-mini as below.

Structure Score

You are given two sentences:

[GT]: `self.gt_sen`
[GEN]: `self.gen_sen`

Evaluate **Structural Frame Similarity**. Focus on whether [GEN] preserves the core grammatical structure and sentence skeleton of [GT]: same basic clause structure, same number/type of major phrases, similar syntactic relationships, consistent verb tense/aspect.

Scoring:

5: Identical structure — same words in same order
4: Nearly identical — same pattern with entity substitutions or minor reordering
3: Moderately similar — core structure maintained but notable changes (e.g., active to passive)
2: Somewhat similar — recognizable elements but significant differences
1: Minimally similar — only basic sentence type matches
0: Completely different structures

Answer: `[ANS] structure: [score]/5`

Entity Score

You are given two sentences:

[GT]: `self.gt_sen`
[GEN]: `self.gen_sen`

Evaluate **Entity Preservation**. Focus on whether [GEN] preserves key entities (people, places, organizations) from [GT]: same named entities, same key objects/concepts, equivalent entities in corresponding roles, preservation of entity relationships.

Scoring:

5: All entities preserved — all key entities from GT appear in GEN with same references
4: Nearly all preserved — minor omissions of non-critical entities or slight variations
3: Most preserved — majority of key entities maintained, some important ones missing/substituted
2: Some preserved — recognizable overlap but significant differences in key entities
1: Few preserved — minimal overlap, only generic categories match
0: No overlap — completely different entities

Answer: `[ANS] entity: [score]/5`

Topic Score

You are given two sentences:

[GT]: `self.gt_sen`
[GEN]: `self.gen_sen`

Evaluate **Topic Consistency**. Focus on whether [GEN] maintains the same main subject/topic as [GT]: same primary entity/concept, same domain/field, same general subject matter, maintains relevance to original topic.

Scoring:

5: Identical topic — exactly the same specific topic/entity with same focus
4: Highly similar — same main topic with slightly different aspects/perspectives
3: Related topic — closely related subjects within same domain/category
2: Loosely related — some connection but notably different topics/focuses
1: Minimally related — tangentially connected or only shares broad category
0: Unrelated — completely different subjects with no meaningful connection

Answer: `[ANS] topic: [score]/5`

Table 4: Inverted Examples with Varying token Lengths

Tokens (#)	Ground-truth Sequence	Inverted Sequence	R1	R2	RL	BS	SS	ES	TS
8	in the Centre-Val de Loire	in the Centre-Val de Loire	1	1	1	1	1	1	1
	is the second era of the Hade	is the third era of the Hade	0.86	0.67	0.86	0.99	1	0.8	1
	who enforce New Zealand's regulatory building control	the enforcement of New Zealand statutory building control	0.63	0.29	0.63	0.77	0.4	0.6	1
	Tyler, the Creator production discography\n\n	Metro Boomin production discography\n\n	0.44	0.29	0.44	0.77	0.8	1	1
	biologist \n Stanley Fields (actor) (	:\n\n John Allen (actor) (born	0.25	0	0.25	0.78	0.6	0.8	0.8
16	species within the genus Conus, these snails are predatory and venomous.	species within the genus Conus, these snails are predatory and venomous.	1	1	1	1	1	1	1
	List of shipwrecks in September 1842\n\nThe list of ship	List of shipwrecks in January 1840\n\nThe list of ship	0.8	0.67	0.8	0.99	0.8	0.8	1
	2017 NCAA Division I Softball Tournament\n\nThe 2017 NCAA Division I	2018 NCAA Division I Women's Soccer Tournament\n\nThe 2018 NCAA Division	0.61	0.38	0.61	0.88	0.8	0.6	0.8
	Rob James\n\nRob James may refer to:\n\nRob James (singer) (	Mark Jones\n\nMark Jones may refer to:\n\nMark Jones (singer) (	0.4	0.22	0.4	0.96	0.8	1	1
	Qi Yuwu, Julian Hee, Jeanette Aw, Felicia Chin,	, Pierre Png, Chen Hanwei, Felicia Chin, Fann Wong	0.25	0.14	0.25	0.82	0.8	1	1
32	1825 in Wales\n\nThis article is about the particular significance of the year 1825 to Wales and its people.\n\nIncumbents\nPrince of Wales \u2013	1840 in Wales\n\nThis article is about the particular significance of the year 1840 to Wales and its people.\n\nIncumbents\nPrince of Wales \u2013	0.91	0.86	0.91	0.99	0.8	0.8	1
	List of European Championships records in swimming\n\nThe European Championships records in swimming are the fastest times ever swum in European Swimming Championships' events.\n\nLong course (50	List of European records in swimming\n\nThe following are the current European records in swimming, as recognized by LEN.\n\nLong course (50 m pool)\n\nShort course (	0.6	0.39	0.53	0.75	0.6	0.8	1
	Institute of Higher National Defence Studies.\n\nThen, he went to the USA for a Master in International Business Management at the University of Florida. He began his career in	.\n\nHe then studied at the Institut d'\u00e9tudes politiques de Paris and at the \u00c9cole nationale d'administration (ENA). He started his career in	0.42	0.12	0.3	0.71	0.2	0.6	0.8
	William Robertson Nicoll\n\nSir William Robertson Nicoll CH LLD (10 October 1851 \u2013 4 May 1923) was a Scottish Free Church	John Duncan (minister)\n\nJohn Duncan (1 May 1845 \u2013 6 March 1914), also known as J. Duncan, was a Free	0.21	0.05	0.21	0.74	0.8	0.8	0.8
	Alessandro Guiccioli\n\nAlessandro Guiccioli (March 5, 1843 \u2013 October 3, 1922)	Gustave de Beaumont\n\nGustave de Beaumont (March 1, 1840 \u2013 March 2, 1921) was	0.09	0	0.09	0.83	0.6	0.4	0.8
64	Rush Hour 2\n\nRush Hour 2 is a 2001 American action comedy film directed by Brett Ratner and written by Jeff Nathanson, based on the characters created by Ross LaManna. A sequel to Rush Hour, it is the second installment in the Rush Hour series and stars Jackie Chan,	Rush Hour 2\n\nRush Hour 2 is a 2001 American buddy cop action comedy film directed by Brett Ratner and written by Jeff Nathanson. It is the sequel to the 1998 film Rush Hour and the second installment in the Rush Hour film series. The film stars Jackie Chan,	0.84	0.67	0.78	0.91	0.8	1	1

hull length of, a beam of, a height of, and a draught of. The submarine was powered by two Germaniawerft F46 four-stroke, six-cylinder supercharged diesel engines producing a total of for use while surfaced, two AEG GU 460/8\u201327 double-	. The U-boat had a displacement of when at the surface and while submerged. The U-boat had a total length of, a pressure hull length of, a beam of, a height of, and a draught of. The submarine was powered by two Germaniawerft F46 six-	0.61	0.48	0.49	0.8	0.4	1	1
the same rights of audience as members of the Bar of Northern Ireland.\n\nThe Advocate General was created as a separate office upon the devolution of policing and justice powers to the Northern Ireland Assembly on 12 April 2010.\n\nUnlike the Advocate General for Scotland, the position is not supported by a distinct government department.	the Scottish Parliament. The office was created in 1999, and is the equivalent of the Parliamentary Under-Secretary of State in the United Kingdom Government.\n\nThe office is not a ministerial post, and the holder is not a member of the Scottish Government. Responsibility for the office is held by the Scottish Secretary.	0.41	0.08	0.24	0.66	0.4	0.4	0.8
producer Thom Wilson and released in 1982 as catalog number VIRUS 10. Singer Jack Grisham credited himself as Jack Ladoga on the sleeve, following a tradition of using a different pseudonym on each release both to confuse audiences and to hide his true identity from the police. Drummer Todd Barnes credited himself	the band's first album, and the first to feature the band's new lineup. The band members used pseudonyms on the album, with the exception of guitarist and vocalist John \"Baz\" Bascaran, who used his real name because he was the only member of the band with a driver's license. Drum	0.21	0.04	0.15	0.59	0.8	0.6	0.8
Linux kernel.\n\nXC3018\nIt is a variant that only supports analog reception and DVB-T digital reception.\n\nTechnical specification\nOutline Dimensions: 7 x 7 x 0.85\u00a0mm\nSupply Voltage (DC): 1.8V, 3.3V\nSystem setting time:	2010.\n\nSpecifications\nFrequency: 2.4\u00a0GHz\nData rate: 1, 2, 5.5, 11\u00a0Mbps\nModulation: DSSS\nPower consumption: 0.1\u00a0W\nOperating temperature: 0\u00a0\u00b0C to 70\u00a0\u00b0C\n	0.09	0	0.09	0.62	0	0	0.8