MAGNET: Augmenting Generative Decoders with Representation Learning and Infilling Capabilities

Savya Khosla[1,2]* **Aditi Tiwari**[2] **Kushal Kafle**[1] **Simon Jenni**[1] **Handong Zhao**[1] **John Collomosse**[1] **Jing Shi**[1]

[1]Adobe Research [2]University of Illinois Urbana-Champaign

Abstract

While originally designed for unidirectional generative modeling, decoder-only large language models (LLMs) are increasingly being adapted for bidirectional modeling. However, unidirectional and bidirectional models are typically trained separately with distinct objectives (generation and representation learning). This separation overlooks the opportunity for developing a more versatile language model and for these objectives to complement each other. In this work, we propose MAGNET, a method for adapting decoder-only LLMs to generate robust representations and infill missing text spans. MAGNET employs three self-supervised training objectives and introduces an attention mechanism that combines bidirectional and causal attention, enabling unified training across all objectives. Our results demonstrate that LLMs adapted with MAGNET (1) surpass strong text encoders on token-level and sentence-level representation learning tasks, (2) generate contextually appropriate text infills by leveraging past and future contexts, (3) perform open-ended text generation without excessive repetition of words or phrases, and (4) preserve the knowledge and reasoning capability gained by the LLM during pretraining.

1 Introduction

Decoder-only LLMs have gained popularity in NLP due to their efficient training and scalability. However, their reliance on causal attention restricts their effectiveness in tasks that require understanding of bidirectional context. This limitation is particularly evident in (1) representation learning tasks such as sentiment analysis and named entity recognition, where understanding the full context of sentences or words is crucial, and (2) text infilling, where filling in missing spans must ensure coherence with the surrounding text.

Some recent efforts (BehnamGhader et al., 2024; Li and Li, 2023; Li et al., 2023; Duki'c and vSnajder, 2024; Du et al., 2021; Donahue et al., 2020) have sought to adapt decoder-only LLMs for representation learning and text infilling. However, as shown in Figure 1, methods that enhance LLMs for text infilling fail to make them effective text encoders, while methods focused on representation learning diminish their generative capabilities.

In this work, we introduce MAGNET (Modified Attention for Generation and Encoding of Text), a method for adapting decoder-only LLMs that have been trained for text generation into more versatile language models. Specifically, MAGNET enables an LLM to (1) generate robust sentence-level and token-level representations, (2) infill missing text spans while maintaining coherence with bidirectional context, (3) perform open-ended text generation without excessive repetition, and (4) preserve the knowledge acquired during pretraining. In essence, MAGNET equips an LLM with representation learning and infilling capabilities while preserving its generative strengths. To achieve this, we use three self-supervised training objectives: (1) a *masked modeling objective* to learn token-level representations, (2) a *contrastive objective* to learn sentence-level representations, and (3) a *missing-span generation objective* to infill text and retain generative capabilities. To facilitate simultaneous training across all these objectives, we deploy a specially crafted attention mask that combines bidirectional and causal attention.

Without any model-specific design, we apply MAGNET to Llama-2-7B (Touvron et al., 2023). We demonstrate that the proposed method requires simple modification and fine-tuning of an off-the-shelf LLM to augment it with representation learning and infilling capabilities. Our results show that MAGNET-adapted Llama-2-7B outperforms other

*Work done during internship at Adobe Research. Correspondence to savyak2@illinois.edu.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 27328–27346

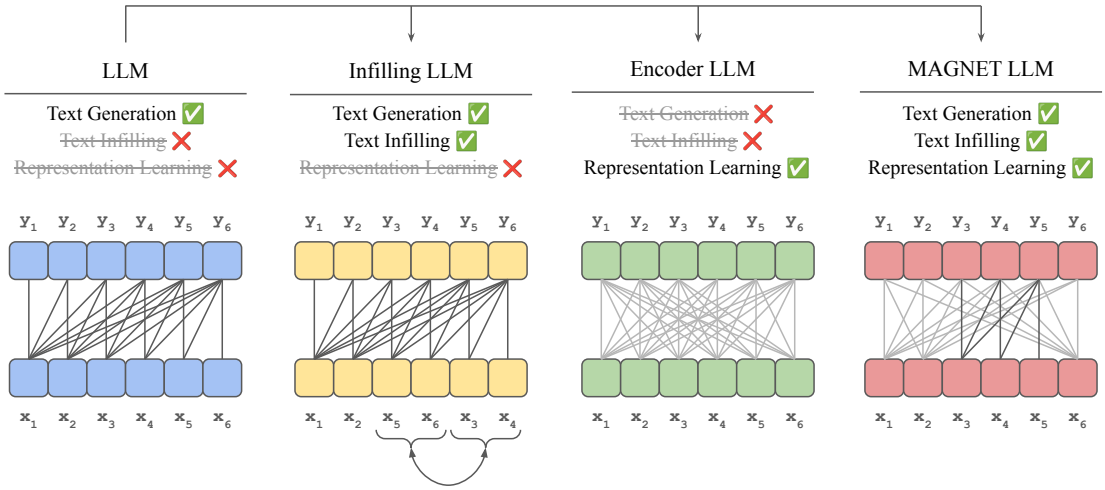


Figure 1: Traditionally, LLMs are trained for text generation using unidirectional attention between the input x and output y (depicted by black lines), whereas text encoders are trained for representation learning using bidirectional attention (depicted by gray lines). MAGNET adapts the attention mechanism of LLMs to combine both unidirectional and bidirectional attention, enhancing them with representation learning and infilling capabilities, while retaining their core generative functions.

methods that adapt the same model for token-level and sentence-level representation learning tasks[1]. We also show that MAGNET improves the infilling capability of the LLM by enabling it to consider the bidirectional context. Further, we analyze the repetition problem in text generated by models that are trained or fine-tuned to encode text and demonstrate that MAGNET-adapted models are significantly better at open-ended text generation than other text encoders. Lastly, we show that MAGNET preserves the knowledge and reasoning capabilities acquired by the LLM during pretraining.

2 Related Works

Representation Learning. Text representation learning focuses on understanding contextual relationships within sentences. Traditionally, encoder models dominated this field due to their bidirectional context modeling, using masked language modeling for token-level representations (Devlin et al., 2019; Liu et al., 2019; He et al., 2020; Clark et al., 2020; He et al., 2021) and special tokens with similarity-based optimization for sentence-level understanding (Gunel et al., 2020; Reimers and Gurevych, 2019; Wu et al., 2020; Carlsson et al., 2021; Gao et al., 2021; Wei et al., 2020). Recent work has explored adapting decoder-only LLMs for text encoding through various methods, including introducing special tokens to the model's vocabulary (Zhang et al., 2024), using last-token or mean-pooled representations (Neelakantan et al., 2022; Wang et al., 2023), or fine-tuning with masked modeling (BehnamGhader et al., 2024) or label supervision (Li et al., 2023; Duki'c and vSnajder, 2024). While some approaches modify the decoder's causal attention to be bidirectional (BehnamGhader et al., 2024; Muennighoff et al., 2024; Li and Li, 2023; Duki'c and vSnajder, 2024; Man et al., 2024), this often compromises the model's text generation abilities. In contrast, MAGNET employs a hybrid attention mechanism that combines causal and bidirectional attention, enabling both robust representation learning and preserved generation capabilities.

Text Infilling. Text infilling requires considering both left and right context when generating text in the middle of a sequence. Encoder-decoder models (Raffel et al., 2019; Lewis et al., 2019; Kalinsky et al., 2023) can handle this task by encoding available context and decoding infilled text. Other approaches have extended masked language modeling to perform span infilling (Joshi et al., 2019; Shen et al., 2023, 2020). Decoder-only models have also been adapted for infilling through various strategies: training models to directly fill marked blanks (Donahue et al., 2020; Du et al., 2021), rearranging training examples to align with infilling objectives (Bavarian et al., 2022; Yang et al., 2019; Aghajanyan et al., 2022; Fried et al., 2022), or using dual generation from both ends of a sentence

[1]It is to be noted that while these other methods adapt the model exclusively for representation learning, MAGNET incorporates additional objectives, making the LLM more versatile and showcasing the advantages of unified training.

until convergence (Nguyen et al., 2023; Serdyuk et al., 2017). However, while these approaches successfully enhance LLMs with infilling capabilities, none have attempted to simultaneously equip them with both infilling and representation learning abilities, as done by MAGNET.

Unifying Text Understanding and Generation. Prior works on unifying natural language understanding and generation within a single framework usually focus on proposing pretraining objectives and task formulations. These approaches typically extend traditional masked language modeling, with innovations like permutation-based objectives for bidirectional context modeling (Yang et al., 2019), autoregressive blank infilling (Du et al., 2021), multi-directional attention masks (Dong et al., 2019), and sequence-to-sequence pretraining (Song et al., 2019; Raffel et al., 2019). However, these approaches require pretraining new networks from scratch, despite decoder-only models demonstrating exceptional scalability and effectiveness. Instead of starting from scratch, we propose a parameter-efficient method that builds upon the rich representations already learned by existing large language models, transforming them into a unified framework for representation learning, text infilling, and text generation.

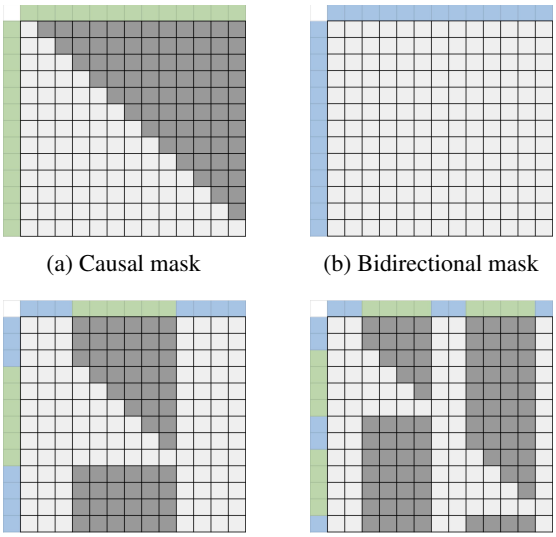
3 Method

Decoder-only models process input sequences through successive blocks of multi-head self-attention, feed-forward networks, and layer normalization. The self-attention mechanism converts the input $\mathbf{x} \in \mathbb{R}^{l \times d}$ into queries $\mathbf{Q}$, keys $\mathbf{K}$, and values $\mathbf{V}$ using linear projections, and computes attention using the formula:

$$\texttt{Attn}_i(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \texttt{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top + \mathbf{M}}{\sqrt{d_k}}\right)\mathbf{V}$$

where $\texttt{Attn}_i$ is the i^{th} head of the multi-head self-attention, d_k represents the dimensionality of the keys/queries, and $\mathbf{M}$ represents the causal mask. This causal mask $\mathbf{M}$ for an autoregressive LLM is a $l \times l$ strictly upper triangular matrix, as shown in Figure 2a, and it ensures that each token can only attend to itself and tokens that precede it.

MAGNET updates the causal attention mechanism of an LLM to incorporate elements of bidirectionality and thereafter fine-tunes the model using self-supervised objectives. We look at the modifications to the attention mechanism in Section 3.1 and the training objectives in Section 3.2.



(a) Causal mask (b) Bidirectional mask

(c) Mask with 1 span (d) Mask with 2 spans

Figure 2: Attention masks for different types of attention mechanisms. The rows of the matrices correspond to the query tokens and the columns correspond to the key tokens. Light gray cells indicate 0, dark gray cells represent $-\infty$, green marks span token positions, and blue marks context token positions. Each context token attends to every other context token, and each span token attends to all context tokens and the preceding span tokens in the same span.

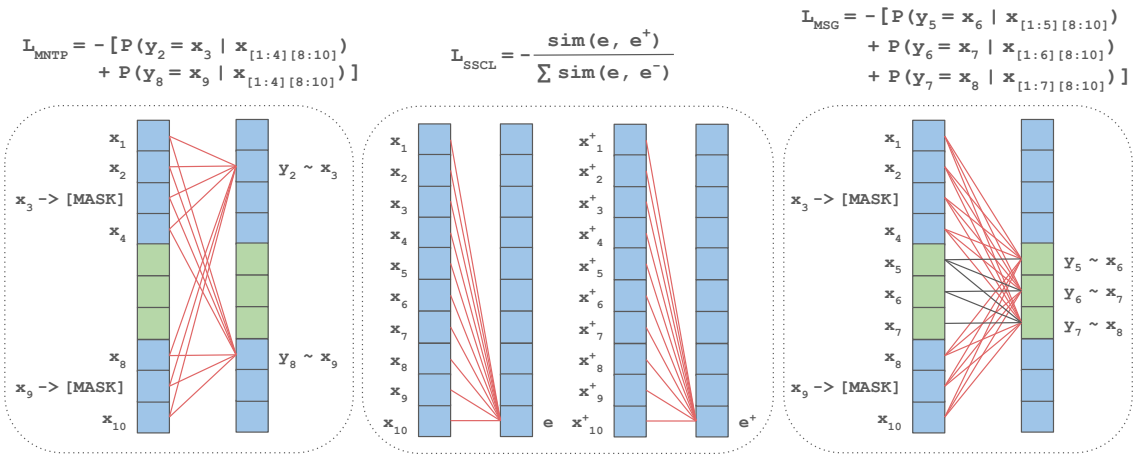
3.1 Modifying Attention

We categorize the input tokens as either *context tokens* or *span tokens* and use the attention mask shown in Figure 2.

Context tokens. Each context token (shown in blue in Figure 2) attends to all other context tokens in the sequence. The attention mask has 0s at output positions corresponding to context tokens, allowing each context token to access information at every other context token. This transformation shifts the original unidirectional LLM into a bidirectional model.

Span tokens. The span tokens (shown in green in Figure 2) are a contiguous span of input tokens that attend to all context tokens and have causal attention among themselves. By enabling span tokens to access surrounding context, we effectively convert the original LLM into an infilling language model. Additionally, the causal attention among span tokens preserves the LLM's generative capabilities, which could be compromised if bidirectionality is fully unlocked (see Section 4.4 for details).

During training, an input sequence includes one or more spans of span tokens surrounded by context tokens. During inference, the attention mechanism can operate in three modes: (1) fully



(a) Masked Next Token Prediction (b) Self-Supervised Contrastive Learning (c) Missing Span Generation

Figure 3: MAGNET training objectives include: (a) Masked next token prediction, which is applied on the output corresponding to the token preceding the masked context token. (b) Self-supervised contrastive learning, which is applied on the model's representation corresponding to the last token. (c) Missing span generation, which is applied on the output corresponding to the span tokens. In this illustration, the red lines denote bidirectional attention and the black lines denote causal attention. Further, for (a) and (c), the output token y_i is trained to predict the input token x_{i+1}, as denoted by "$y_i \sim x_{i+1}$"

causal/unidirectional for open-ended text generation tasks, (2) fully bidirectional representation learning tasks, or (3) a combination of causal and bidirectional for text infilling.

3.2 Training Objectives

MAGNET fine-tunes an off-the-shelf LLM using three self-supervised objectives. These objectives are illustrated in Figure 3 and discussed below.

3.2.1 Masked Next Token Prediction (MNTP)

MNTP enables the model to realize its newly enabled bidirectional attention capability. The task is defined as follows: Given an input sequence $\mathbf{x} = (x_1, x_2, ..., x_L)$, we select a fraction of the input tokens for masking and train the model to predict these masked tokens. In our setup, we find that selecting 20% of the input tokens for masking works well. Further, following (Devlin et al., 2019), we replace 80% of the selected tokens with a [MASK] token, 10% with a random token from the model's vocabulary, and leave the remaining 10% unchanged. Since LLMs are trained to predict the next token in a sequence, we use the token representations from position l to predict a masked token at position $l+1$ (as shown in Figure 3a). In Appendix D, we also explore the possibility of using the standard masked token prediction (MTP) objective, where the output at token l predicts the masked token at position l and find that MTP performs poorly for LLMs that are trained to predict autoregressively. MNTP is optimized using categorical cross-entropy loss:

$$\mathcal{L}_{\text{MNTP}} = \frac{-1}{NL} \sum_{n=1}^{N} \sum_{l=1}^{L} \sum_{v=1}^{V} \left(\mathbb{1}_{\text{mask}}(l+1) \cdot (y_{lv}^{(n)} \log(\hat{y}_{lv}^{(n)})) \right)$$

where N denotes batch size, L denotes sequence length, V denotes vocabulary size, $\mathbb{1}_{\text{mask}}(l+1)$ is 1 if position $l+1$ is masked and 0 otherwise, and y_{lv} and $\hat{y}_{lv}$ represent true and predicted probabilities for v^{th} token in vocabulary at position l in the sequence. Note that this task is conducted exclusively with the context tokens.

3.2.2 Self-Supervised Contrastive Learning (SSCL)

Since LLMs are not explicitly trained to capture the entire input context and generate sentence-level representations, we employ SSCL to transform them into text encoders. The task is defined as follows: Given an input sequence $\mathbf{x}$, we generate its augmented view $\mathbf{x}^+$ and align their encoded representations $\mathbf{e} = f(\mathbf{x})$ and $\mathbf{e}^+ = f(\mathbf{x}^+)$ in the embedding space, while distancing them from the encodings $\mathbf{e}^- = f(\mathbf{x}^-)$ of other input sequences $\mathbf{x}^-$ in a training batch. Specifically, we employ paraphrasing (Damodaran, 2021) to generate augmented views of an input, and add an instruction "*Given the sentence, find its representation:*" to

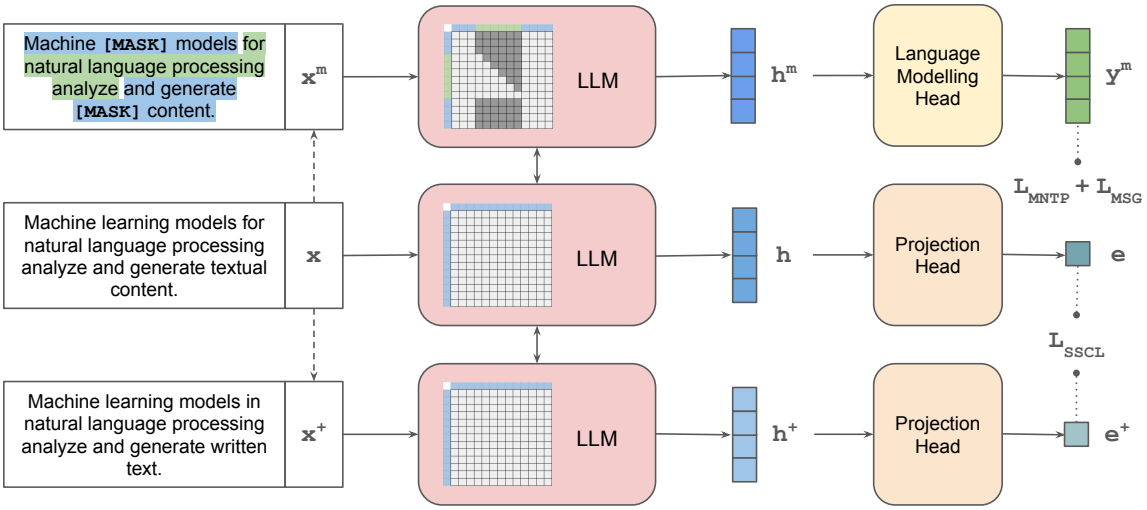


Figure 4: MAGNET processes three views of the input using different attention mechanisms within the same LLM. The model is trained (or fine-tuned) using three self-supervised learning objectives simultaneously to augment it with the ability to generate token-level and sentence-level representations and perform text infilling tasks, while maintaining its original left-to-right text generation capability.

the training examples (Jiang et al., 2023). Then, we use the output corresponding to the last token ([EOS]) of the final hidden states as the sentence encoding. Our choice of using the last token representation as the encoding is guided by the fact that MAGNET optimizes simultaneously for token-level and sentence-level representations. Since the last token's representation is not used for token-level optimization (because the representation of input token l is given by output token $l-1$), this choice enables us to disentangle the two representation learning tasks during joint training. We use InfoNCE (van den Oord et al., 2018) with in-batch negatives as the loss function:

$$\mathcal{L}_{\text{SSCL}} = \frac{-1}{N}\sum_{i=1}^{N}\log\frac{\exp(\mathbf{e}_i \cdot \mathbf{e}_i^{+}/\tau)}{\sum_{j=1}^{N}\exp(\mathbf{e}_i \cdot \mathbf{e}_j^{-}/\tau)}$$

where N represents the batch size and τ denotes the temperature for logit scaling.

3.2.3 Missing Span Generation (MSG)

MSG provides text infilling capabilities to the left-to-right autoregressive model. The task is defined as: Given a position p and an input sequence $\mathbf{x} = (x_1, ..., x_p, x_q, ..., x_L)$, generate a plausible sequence of m tokens $\mathbf{y} = (y_1, y_2, ..., y_m)$ that fits between x_p and x_q. More specifically, in our training setup, this task entails predicting a span token y_l conditioned on all context tokens in $\mathbf{x}$ and the preceding span tokens $y_{[1..l-1]}$. We train using categorical cross-entropy loss computed over the predicted span tokens:

$$\mathcal{L}_{\text{MSG}} = \frac{-1}{N}\sum_{n=1}^{N}\sum_{l=1}^{L}\sum_{v=1}^{V}\mathbb{1}_{\text{span}}(l)\cdot(y_{lv}^{(n)}\log(\hat{y}_{lv}^{(n)}))$$

where N denotes batch size, L denotes sequence length, V denotes vocabulary size, $\mathbb{1}_{\text{span}}(l)$ is 1 if the token at position l is a span token and 0 otherwise, and y_{lv} and $\hat{y}_{lv}$ are the true and predicted probabilities for token v in the vocabulary at position l in the sequence. The standard next token prediction task of LLMs can be considered as a special case of this objective, wherein all input tokens are span tokens (and the attention mechanism reduces to causal attention). Thus, a beneficial side effect of this task is that the model retains its text generation capability while learning bidirectional representations.

3.3 Approach Overview

Figure 4 provides an overview of MAGNET. Starting with a training example $\mathbf{x}$, the process unfolds in two parallel streams – (1) One or more contiguous spans of M tokens in $\mathbf{x}$ are marked as span tokens, while a fraction of the remaining tokens (context tokens) is masked to form $\mathbf{x}^{\text{m}}$. (2) $\mathbf{x}$ is augmented to get $\mathbf{x}^{+}$. The input sequences $\mathbf{x}$, $\mathbf{x}^{\text{m}}$ and $\mathbf{x}^{+}$ are processed by the base decoder model to produce hidden states $\mathbf{h}$, $\mathbf{h}^{\text{m}}$ and $\mathbf{h}^{+}$. From $\mathbf{h}^{\text{m}}$, a language modeling head generates $\mathbf{y}^{\text{m}}$, which is used to compute $\mathcal{L}_{\text{MNTP}}$ and $\mathcal{L}_{\text{MSG}}$. Parallelly, $\mathbf{h}$ and $\mathbf{h}^{+}$ are processed using a projection head to get $\mathbf{e}$ and $\mathbf{e}^{+}$, which are used to compute $\mathcal{L}_{\text{SSCL}}$.

The overall loss function is:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{MNTP}} + \lambda_2 \mathcal{L}_{\text{SSCL}} + \lambda_3 \mathcal{L}_{\text{MSG}}$$

For processing $\mathbf{x}$ and $\mathbf{x}^+$, the decoder uses a bidirectional attention mask (as shown in Figure 2b). For processing $\mathbf{x}^{\text{m}}$, the decoder employs an attention mask similar to those depicted in Figures 2c and 2d. In some cases, when all input tokens are marked as span tokens, the attention mask reduces to causal attention, as shown in Figure 2a.

4 Experiments

In this section, we demonstrate that MAGNET enhances a decoder-only LLM with representation learning and infilling capabilities while preserving its original generative abilities. Specifically, we show that LLMs adapted with MAGNET outperform the base model and other adaptation methods on representation learning tasks (Sections 4.1 and 4.2). We then highlight how MAGNET significantly improves the LLM's ability to infill missing spans (Section 4.3) and analyze the effect of MAGNET on the original generative and reasoning capabilities of the LLM (Section 4.5). Finally, we provide a brief analysis highlighting which layers retain similarity to the base model and which undergo the most modification.

All training details are mentioned in Appendix A. Additionally, we present ablation experiments demonstrating the benefits of training a bidirectional language model with a causal objective in Appendix C. Note that our goal is not to achieve state-of-the-art results on a specific benchmark. Instead, we aim to enhance a pretrained LLM with additional capabilities while preserving its original performance. Therefore, our main baselines are the base LLM and other methods that augment the same LLM with specific capabilities.

4.1 Word-Level Tasks

We evaluate the token-level representations on three tasks – (1) chunking, (2) named entity recognition, and (3) part-of-speech tagging – using the CoNLL-2003 dataset (Sang and Meulder, 2003). After applying the training objectives proposed in Section 3.2, we train a linear classifier on top of the frozen representations obtained from the last hidden state of the model. The word-level embeddings are obtained by averaging the representations of the tokens that make up that word. Further, the representation of the token at position i is given by the embedding at position $i-1$.

Model	Chunking	NER	POS-Tags
Encoder models			
BERT-Large	71.77	90.09	75.12
XLNet-Large	79.70	93.67	83.02
DeBERTa-Large	85.74	94.97	86.49
StructBERT-Large	89.99	97.31	90.86
Llama 2 models			
Llama-2-7B	88.23	96.59	91.53
LLM2Vec	89.66	96.05	90.53
LLM2Vec$^{[\text{MNTP}]}$	91.61	97.16	92.61
MAGNET	**92.64**	**98.31**	**93.34**

Table 1: Results on word-level tasks. LLM2Vec (BehnamGhader et al., 2024) adapts the model using MNTP and SimCSE. LLM2Vec$^{[\text{MNTP}]}$ is an intermediate state of LLM2Vec that is trained only on MNTP. All numbers except those for MAGNET are taken from (BehnamGhader et al., 2024).

Table 1 compares MAGNET with powerful encoder models and LLM2Vec (BehnamGhader et al., 2024), a recent method for adapting decoder-only LLMs for representation learning. The second-best approach, LLM2Vec$^{[\text{MNTP}]}$, relies solely on MNTP for model adaptation. In contrast, MAGNET integrates both representation learning objectives (MNTP and SSCL) and generative objectives (MSG). The superior performance of MAGNET over LLM2Vec$^{[\text{MNTP}]}$, despite using the same training data, model, and parameters, highlights the synergistic advantages of a unified training strategy for word-level representation learning.

4.2 Sentence-Level Tasks

We evaluate sentence-level representations on multiple semantic similarity and clustering benchmarks (Muennighoff et al., 2022). We perform these tasks using the representation corresponding to the last token (`[EOS]`), without performing any task-specific training. Further, task-specific instructions (Table 10) are used for extracting relevant representations (Su et al., 2022; Wang et al., 2023).

We compare the text encoding capabilities of MAGNET with other recently proposed methods for transforming decoder models into text encoders, viz. LLM2Vec (BehnamGhader et al., 2024) and Echo Embeddings (Springer et al., 2024). Table 2 shows the results on Semantic Textual Similarity (STS) task and Table 3 shows the results on clustering tasks. As can be seen, MAGNET outperforms other adaptation methods on STS and clustering tasks. As previously noted, the fact that MAGNET surpasses LLM2Vec suggests the potential benefit

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg
Encoder models (finetuned using SimCSE)								
BERT-Base	68.40	82.41	74.38	80.91	78.56	76.85	72.23	76.25
RoBERTa-Base	70.16	81.77	73.24	81.36	80.65	80.22	68.56	76.57
RoBERTa-Large	**72.86**	83.99	75.62	**84.77**	**81.80**	81.98	71.26	78.90
Llama 2 models								
Llama-2-7B	50.98	74.02	62.86	67.09	71.03	63.56	67.22	65.25
Echo Embeddings	52.40	72.40	61.24	72.67	73.51	65.73	64.39	66.05
LLM2Vec	65.39	79.26	72.98	82.72	81.02	78.32	71.77	75.92
MAGNET	67.98	**84.66**	**77.67**	84.17	79.44	**82.88**	**78.77**	**79.36**

Table 2: Results on STS tasks. The encoder models are trained using SimCSE and their results are taken from Gao et al. (2021). The results for Llama-2-7B are obtained using the last token embedding from the final hidden state as the sentence representation. The results for LLM2Vec and Echo Embeddings are taken from BehnamGhader et al. (2024) and Springer et al. (2024), respectively.

Dataset	BiorxivClustering	TwentyNewsgroups	MedrxivClustering
Echo Embeddings	25.92	23.42	24.30
LLM2Vec	34.69	30.76	29.49
MAGNET	**35.10**	**53.31**	**30.21**

Table 3: Results on clustering tasks. The results for LLM2Vec and Echo Embeddings are taken from (BehnamGhader et al., 2024) and (Springer et al., 2024), respectively.

Method	ROC Stories	Wikitext-103
Llama-2-7B	13.9347	22.0399
MAGNET	**9.5161**	**15.4573**

Table 4: Results on the infilling tasks. We measure the perplexity (PPL) for sentence infilling and block-of-text infilling on ROC-Stories and Wikitext-103, respectively.

Method	Score
Unidirectional Llama-2-7B	53.5
Zero-Shot Setup	5.5
Five-Shot Setup	54.5
MAGNET	**62.0**

Table 5: Human evaluation for the infilling tasks. The score denotes the percentage of infillings that were considered contextually appropriate by human evaluators.

of a unified training approach.

4.3 Infilling Task

To test infilling capabilities, we evaluate the perplexity (PPL) of Llama-2-7B and MAGNET-adapted Llama-2-7B on the test set of ROC Stories (Mostafazadeh et al., 2016) and Wikitext-103 (Merity et al., 2016). For ROC Stories, we randomly mask out a sentence from each 5-sentence story, while for Wikitext-103, we mask up to three spans with lengths ranging from 8 to 32 tokens. Following (Donahue et al., 2020), we compute PPL only for the tokens comprising the original masked out spans. The results are presented in Table 4, and they show that the base model (Llama-2-7B) exhibits significantly higher perplexity for the masked spans compared to MAGNET, demonstrating that MAGNET effectively augments the base model with text infilling capabilities.

We also conduct experiments using zero-shot and few-shot learning to enable Llama-2-7B to incorporate all the surrounding context when infilling a missing span. We explore various prompting strategies and found that while a zero-shot setup did not yield sensible infillings, a five-shot setup with descriptive prompts resulted in more context-aware infillings (refer to Appendix B for details). For a comprehensive analysis, we conducted a human evaluation to compare the quality of infillings generated by the base model, its zero-shot variant, its few-shot variant, and its MAGNET adaptation. In this evaluation, we randomly sampled 100 stories from the ROC Stories dataset, masked out one of their middle sentences, and tasked the models with infilling the missing sentence. Two human annotators on Amazon Mechanical Turk (with at least a high school diploma) then independently assessed whether each generated sentence was contextually appropriate and contributed to a coherent story. The results are presented in Table 5, showing that the infillings generated by MAGNET-adapted model are more coherent than those generated by the base model. We show some qualitative exam-

Method	Wikitext-103		ROC Stories	
	Rep-Sen	Rep-4	Rep-Sen	Rep-4
Llama-2-7B	0.0056	0.0601	0.0381	0.0163
LLM2Vec	0.2044	0.4747	0.2945	0.5243
MAGNET	0.0151	0.2047	0.0737	0.2573

Table 6: Analyzing the repetition problem. Both LLM2Vec and MAGNET are applied for 3400 iterations.

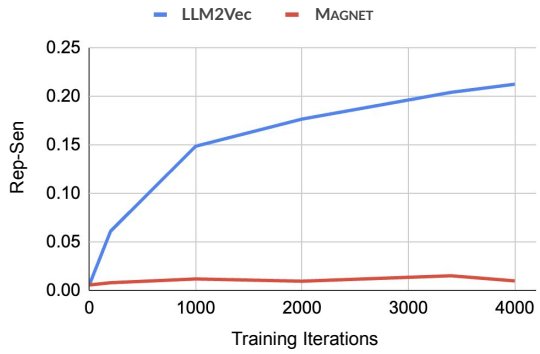


Figure 5: LLM2Vec increases text repetition with more training, while no such trend is observed for MAGNET.

ples of infilling in Table 12.

4.4 Repetition Problem

The repetition problem in text generation refers to the issue when generative models repeatedly produce the same phrases or sentences. Prior studies have identified that this issue often results from biases in the training data, limitations in the model's design, or standard likelihood training and inference (Holtzman et al., 2019; Welleck et al., 2019; Fu et al., 2020; Xu et al., 2022). In our study, we find that when generative decoder models are adapted into text encoders by enabling bidirectional attention (BehnamGhader et al., 2024; Li and Li, 2023; Li et al., 2023), the issue of repetition is significantly worsened. For example, Table 13 shows the texts generated (using greedy decoding) by the original Llama-2-7B and its LLM2Vec adaptation (BehnamGhader et al., 2024). We observe noticeable repetitions in the text generated by LLM2Vec-adapted-LLaMA, although the fine-tuning data (Wikitext-103) had almost no sentence-level repetitions (0.02%).

For quantitative detection of text repetitions, we compute *Rep-Sen* $= 1.0 - \frac{|\text{unique sentences}|}{|\text{sentences}|}$ and *Rep-n* $= 1.0 - \frac{|\text{unique n-grams}|}{|\text{n-grams}|}$, as done by prior works analyzing the repetition problem (Holtzman et al., 2019; Welleck et al., 2019; Xu et al., 2022). Specifically, we create a *prefix-dataset* from the test sets of Wikitext-103 and ROC Stories, consisting of 5-word and single-sentence prefixes, respectively. The model is then tasked with autoregressively generating text based on these prefixes. Table 6 shows the repetition metrics for Llama-2-7B and its adaptations using LLM2Vec and MAGNET. As can be seen, in comparison to LLM2Vec, MAGNET makes the base model significantly less prone to repeating sentences. For instance, for Wikitext-103, LLM2Vec makes Llama-2-7B 36.5 times more likely to repeat sentences, while MAGNET only makes it 2.7 times more likely. Further, as shown in Figure 5, the repetition problem exacerbates with additional iterations of LLM2Vec training, whereas no similar trend is observed with MAGNET.

We conjecture that LLM2Vec is significantly more prone to generating repetitive text because it exclusively focuses on learning representations with bidirectional attention. This training approach perhaps makes the decoder model somewhat similar to bidirectional LMs like BERT, which are known to repeat words when used for text generation (Table 13). MAGNET solves this issue by having autoregressive generation as an objective.

4.5 Knowledge and Reasoning Tasks

We assess the effect of MAGNET on the knowledge and reasoning capabilities acquired by the LLM during pretraining. Specifically, we evaluate its performance on HellaSwag (0-shot) (Zellers et al., 2019), BBH (3-shot) (Suzgun et al., 2022), ARC (0-shot) (Clark et al., 2018), MMLU (5-shot) (Hendrycks et al., 2021), and NaturalQuestions (5-shot) (Kwiatkowski et al., 2019), covering commonsense reasoning and world knowledge.

For this evaluation, we fine-tune the model on the SlimPajama dataset (Soboleva et al., 2023), which includes diverse text sources such as CommonCrawl, C4, GitHub, Books, ArXiv, Wikipedia, and StackExchange. This choice ensures that the fine-tuning data resembles Llama-2-7B's original pretraining distribution (despite its exact composition being unknown). By doing so, we mitigate potential biases introduced by highly structured datasets like Wikitext, which could favor Wikipedia-derived tasks such as NaturalQuestions while disadvantaging commonsense reasoning benchmarks like HellaSwag.

As shown in Table 7, MAGNET has minimal impact on the model's knowledge and reasoning capabilities. The minor variations observed can be attributed to differences in dataset composition during fine-tuning. Furthermore, to ensure a com-

Model	HellaSwag	BBH	ARC		NQ	MMLU			
			Easy	Challenge		Humanities	STEM	Social Science	Other
Llama-2-7B	75.51	33.57	73.95	44.28	24.02	43.27	36.09	53.04	54.84
MAGNET	75.08	32.22	74.33	44.52	24.22	42.25	36.63	52.64	52.40

Table 7: Evaluating the impact of MAGNET on Llama-2-7B's performance across benchmarks. The metrics are computed using the LM Evaluation Harness (Gao et al., 2024). Due to the undisclosed evaluation prompts for Llama, reproducing the exact baseline results is difficult. We adopt the same setup for both Llama-2-7B and MAGNET.

Projection Type	Average Norm
query	1.2386 ± 0.4749
key	1.2659 ± 0.4244
value	0.3013 ± 0.0610
output	0.2716 ± 0.0358

Table 8: Average Frobenius norm of LoRA updates across layers in LLaMA-2-7B.

prehensive evaluation, we assess the performance of the model fine-tuned with SlimPajama on representation learning tasks and find the results consistent with the metrics reported in Tables 1 and 2 (where Wikitext-103 was used for fine-tuning). Specifically, when fine-tuned with SlimPajama, the model attains 92.00% on chunking, 98.30% on NER, 93.21% on POS-tagging, and an average of 79.33% on the STS tasks.

4.6 Parameter Adaptation Analysis

MAGNET fine-tunes the model using LoRA applied to the query, key, value, and output projections of the pretrained LLM. The adapted projection weights can be expressed as:

$$\mathbf{W}_{\text{adapted}} = \mathbf{W}_{\text{base}} + \mathbf{A}\mathbf{B}^{\top}$$

where $\mathbf{W}_{\text{base}}$ denotes the original weight matrix, and $\mathbf{A} \in \mathbb{R}^{d\times r}$, $\mathbf{B} \in \mathbb{R}^{d\times r}$ are the low-rank matrices learned during fine-tuning.

To quantify the extent of adaptation, we compute the Frobenius norm of the update matrix $\mathbf{A}\mathbf{B}^{\top}$. Table 8 reports the average $\|\mathbf{A}\mathbf{B}^{\top}\|_F$ across all layers for each projection type in LLaMA-2-7B, indicating that MAGNET induces larger updates to the query and key projection matrices compared to the value and output projections. This aligns with the method's focus on modifying the attention mechanism by altering the attention mask. Since attention scores are computed using the product $\mathbf{Q}\mathbf{K}^{\top}$, adjusting the query and key matrices is essential to support the updated attention maps. In contrast, value projections carry contextual information after attention is applied. As these components are already well learned during pretraining, they require minimal modification for effective adaptation. We observe no clear pattern in the magnitude of parameter changes across layers.

5 Conclusion

In this work, we presented MAGNET, a method to transform causal LLMs into text encoders and infilling language models with bidirectional context-capturing ability. Through extensive experiments, we show that MAGNET uniquely equips LLMs with abilities that are beyond the scope of traditional text encoders or decoders. Thus, MAGNET shows the potential to unify text generation and text encoding within a single framework.

Limitations

Given Llama-2-7B's undisclosed pre-training data composition, there is a potential risk of test set contamination. We mitigate the undue influence of data contamination by benchmarking against the base model and other Llama-2-7B adaptations (LLM2Vec and Echo Embeddings). Future work should establish benchmarks guaranteed to be excluded from LLM pre-training data.

While MAGNET preserves open-ended generation better than other bidirectional adaptation methods, it does impact the generation quality. For instance, fine-tuning Llama-2-7B with MAGNET increases Wikitext-103 test-set perplexity from 6.4 to 7.6, indicating slightly reduced next-token prediction confidence despite maintaining artifact-free generation.

For infilling tasks, we focus solely on enabling the base LLM to leverage surrounding context for coherent text completion. The MAGNET-adapted LLM shows reduced performance when infilling lengthy (more than 128 tokens) mid-text sequences. This limitation can be addressed by: chunking long infills into smaller segments, modifying MSG parameters, or task-specific fine-tuning for the infilling task.

References

Armen Aghajanyan, Po-Yao (Bernie) Huang, Candace Ross, Vladimir Karpukhin, Hu Xu, Naman Goyal, Dmytro Okhonko, Mandar Joshi, Gargi Ghosh, Mike Lewis, and Luke Zettlemoyer. 2022. Cm3: A causal masked multimodal model of the internet. *ArXiv*.

Mohammad Bavarian, Heewoo Jun, Nikolas A. Tezak, John Schulman, Christine McLeavey, Jerry Tworek, and Mark Chen. 2022. Efficient training of language models to fill in the middle. *ArXiv*.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. *COLM*.

Fredrik Carlsson, Amaru Cuba Gyllensten, Evangelia Gogoulou, Erik Ylipää Hellqvist, and Magnus Sahlgren. 2021. Semantic re-tuning with contrastive tension. In *ICLR*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*.

Prithiviraj Damodaran. 2021. Parrot: Paraphrase generation for nlu.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Chris Donahue, Mina Lee, and Percy Liang. 2020. Enabling language models to fill in the blanks. *ArXiv*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, M. Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *NIPS*.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2021. Glm: General language model pretraining with autoregressive blank infilling. In *ACL*.

David Duki'c and Jan vSnajder. 2024. Looking right is sometimes right: Investigating the capabilities of decoder-only llms for sequence labeling. In *ACL*.

Daniel Fried, Armen Aghajanyan, Jessy Lin, Sida I. Wang, Eric Wallace, Freda Shi, Ruiqi Zhong, Wen tau Yih, Luke Zettlemoyer, and Mike Lewis. 2022. Incoder: A generative model for code infilling and synthesis. *ArXiv*.

Zihao Fu, Wai Lam, Anthony Man-Cho So, and Bei Shi. 2020. A theoretical analysis of the repetition problem in text generation. In *AAAI*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. A framework for few-shot language model evaluation.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *EMNLP*.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Ves Stoyanov. 2020. Supervised contrastive learning for pre-trained language model fine-tuning. *ICLR*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *ArXiv*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *ArXiv*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *ICLR*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *ArXiv*.

J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *ArXiv*.

Ting Jiang, Shaohan Huang, Zhongzhi Luan, Deqing Wang, and Fuzhen Zhuang. 2023. Scaling sentence embeddings with large language models. *ArXiv*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2019. Spanbert: Improving pre-training by representing and predicting spans. *TACL*.

Oren Kalinsky, Guy Kushilevitz, Alex Libov, and Yoav Goldberg. 2023. Simple and effective multi-token completion from masked language models. In *Findings*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc V. Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *TACL*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *ArXiv*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdel rahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*.

Xianming Li and Jing Li. 2023. Bellm: Backward dependency enhanced large language model for sentence embeddings. In *NAACL*.

Zongxi Li, Xianming Li, Yuzhang Liu, Haoran Xie, Jing Li, Fu Lee Wang, Qing Li, and Xiaoqin Zhong. 2023. Label supervised llama finetuning. *ArXiv*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*.

Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, and Thien Huu Nguyen. 2024. Ullme: A unified framework for large language model embeddings with generation-augmented learning. *ArXiv*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *ArXiv*.

N. Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James F. Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. *ArXiv*.

Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2024. Generative representational instruction tuning. *ArXiv*.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. In *EACL*.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas A. Tezak, Jong Wook Kim, Chris Hallacy, Johannes Heidecke, Pranav Shyam, Boris Power, Tyna Eloundou Nekoul, Girish Sastry, Gretchen Krueger, David P. Schnurr, Felipe Petroski Such, Kenny Sai-Kin Hsu, Madeleine Thompson, Tabarak Khan, Toki Sherbakov, Joanne Jang, Peter Welinder, and Lilian Weng. 2022. Text and code embeddings by contrastive pre-training. *ArXiv*.

A. Nguyen, Nikos Karampatziakis, and Weizhu Chen. 2023. Meet in the middle: A new pre-training paradigm. *ArXiv*.

Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP*.

E. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *CoNLL*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*.

Dmitriy Serdyuk, Nan Rosemary Ke, Alessandro Sordoni, Adam Trischler, Christopher Joseph Pal, and Yoshua Bengio. 2017. Twin networks: Matching the future for sequence generation. *ArXiv*.

Tianxiao Shen, Hao Peng, Ruoqi Shen, Yao Fu, Zaïd Harchaoui, and Yejin Choi. 2023. Film: Fill-in language models for any-order generation. *ArXiv*.

Tianxiao Shen, Victor Quach, Regina Barzilay, and T. Jaakkola. 2020. Blank language models. *ArXiv*.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. 2023. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. *ICLR*.

Jacob Mitchell Springer, Suhas Kotha, Daniel Fried, Graham Neubig, and Aditi Raghunathan. 2024. Repetition improves language model embeddings. *ArXiv*.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2022. One embedder, any task: Instruction-finetuned text embeddings. *ArXiv*.

Mirac Suzgun, Nathan Scales, Nathanael Scharli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. In *ACL*.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V.

Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *ArXiv*.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023. Improving text embeddings with large language models. *ArXiv*.

Xiangpeng Wei, Yue Hu, Rongxiang Weng, Luxi Xing, Heng Yu, and Weihua Luo. 2020. On learning universal representations across languages. *ArXiv*.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training. *ArXiv*.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. Clear: Contrastive learning for sentence representation. *ArXiv*.

Jin Xu, Xiaojiang Liu, Jianhao Yan, Deng Cai, Huayang Li, and Jian Li. 2022. Learning to break the loop: Analyzing and mitigating repetitions for neural text generation. *ArXiv*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NIPS*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *ACL*.

Jintian Zhang, Cheng Peng, Mengshu Sun, Xiang Chen, Lei Liang, Zhiqiang Zhang, Jun Zhou, Huajun Chen, and Ningyu Zhang. 2024. Onegen: Efficient one-pass unified generation and retrieval for llms. In *EMNLP*.

A Training Details

MAGNET fine-tunes Llama-2-7B using LoRA (Hu et al., 2021) with $r = 16$ and $\alpha = 32$. We use the AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1e-8$, apply bfloat16 quantization, and use scaled-dot-product attention (SDPA). All experiments are performed on a single NVIDIA A100 GPU, with the MAGNET adaptation of Llama-2-7B taking approximately 7 hours. We discuss the training dataset and hyperparameters for the different objectives/tasks below.

Datasets. For representation learning and infilling tasks, we train on Wikitext-103 (Merity et al., 2016) to ensure a fair comparison with our baselines in Sections 4.1 and 4.2. For knowledge and reasoning tasks (Section 4.5), we use SlimPajama (Soboleva et al., 2023) to mitigate biases from highly structured datasets like Wikitext, which could favor Wikipedia-derived tasks. The WikiText-103 dataset is released under the Creative Commons Attribution-ShareAlike license, and SlimPajama is available under the Apache 2.0 license, both permitting use in open-source research.

MNTP. We train for 4200 iterations using a batch size of 32, a learning rate of 3e-5, and a max sequence length of 512. We select 20% of the tokens for masking – 80% of the selected tokens are replaced with a `[MASK]` token, 10% tokens are replaced with a random token from the model's vocabulary, and 10% tokens are left unchanged. For Llama-2-7B, we use "_" as the mask token.

SSCL. We train for 800 iterations using a batch size of 64, a learning rate of 3e-5, and a max sequence length of 128. To extract representations we use the prompt "*Given the sentence, find its representation:*" and extract the representations corresponding to the last token. The training data is created by extracting lines longer than 20 words and paraphrasing them for the positive examples. We set $\tau = 0.1$ in equation 3.2.2.

MSG. Similar to MNTP, we train for 4200 iterations using a batch size of 32, a learning rate of 3e-5, and a max sequence length of 512. A training example can have up to 2 missing spans, with span length ranging from 4 to 128 tokens.

Overall Loss. For the first 3400 iterations, we optimize the loss (equation 3.3) with $\lambda_1 = 1$, $\lambda_2 = 0$, and $\lambda_3 = 1$, and for the next 800 iterations $\lambda_1 = 1$, $\lambda_2 = 9$, and $\lambda_3 = 1$. Initially, we train with only MNTP and MSG, as these objectives help the model learn to capture future context—a capability the base model lacks. However, this choice mainly contributes to faster training, as similar results are obtained when training with all objectives from the start.

Word-Level Tasks. Using the frozen representations from the last hidden layer of the base model, we train a linear classifier for the three word-level tasks (Chunking, NER, and POS-tagging). Specifically, we train on the CoNLL-2003 train set for 4000 steps using a batch size of 8, a learning rate of 5e-4, and a dropout rate of 0.1.

B Contextual Prompt Infilling

To thoroughly evaluate the infilling capability of the base model, we perform zero-shot and few-shot experiments where the model is shown both preceding and following context of a missing span of text.

B.1 Zero-Shot Evaluation

To this end, we experimented with four types of prompts to infill a missing line in five-line stories from the ROC Stories dataset. The four prompting strategies we used are:

Blank Infilling Prompt. In this setting, we add a blank token (_) at the infilling position and use the following prompt:

Generate the missing line represented by _ in the given text: <text>.
Generate a single sentence.
The missing line is:

Here, *<text>* represents the input text with "_" in place of a missing sentence.

Contextual Prompt. In this setting, we provide the past and future context of the missing line and use the following prompt:

Fill in the missing sentence between "<past-context>" and "<future-context>".
Generate only one sentence. The missing sentence is:

Prefix-Suffix Prompt. In this setting, we give the past context of a missing sentence as a prefix and the future context as a suffix and ask the model to generate the middle. Specifically, we use the following prompt:

Given the prefix and the suffix, generate the middle sentence.
Prefix: <past-context>.
Suffix: <future-context>.
Generate only one sentence.
Middle: .

Line-by-Line Prompt. In this setting, we make the prompt more descriptive by providing all the available context, specifying the line number for all the available lines, and asking for the missing line. For instance, if the task is to infill the second line of a five-sentence story, the prompt would be:

You have a five-sentence story with some missing text.
Here is the context for each line, with the missing line indicated:
Line 1: <line-1>
Line 2: [Missing Line]
Line 3: <line-3>
Line 4: <line-4>
Line 5: <line-5>
Please generate the missing line of the story.
Please generate only the missing line and nothing else.
The missing line is: Line 2:

For the abovementioned prompting strategies, we experimented with various prompt variations, including paraphrasing the instructions, using `"[MASK]"`, `"[blank]"` or `"_"` to denote the missing line, and addressing common avoidable errors using the instructions (for e.g., adding "*Generate only one line.*" to enforce single line infillings and avoid formatting issues). In general, we find that regardless of the prompting strategy used, Llama-2-7B repeats/paraphrases one of the provided lines or summarizes the context as the infilling. In some cases it even ends up generating totally random text (like code). This is perhaps because the model is not trained for the infilling task. Table 11 shows some qualitative examples of text infilling using the different prompting methods.

B.2 Few-Shot Evaluation

To improve infilling results from the base model, we employed few-shot learning techniques with various prompting styles – blank infilling, prefix-suffix, and line-by-line. Specifically, we provided five solved examples in the model's context using the chosen prompt format and asked the model to infill the missing line in the sixth example. We observed that more descriptive prompts and examples led to better output from the model, and the line-by-line prompting style seemed to be the most effective in enabling coherent infillings. We present qualitative examples of the infilling generated using this approach in Table 12.

C Training Objective Ablation Analysis

We perform ablation experiments to evaluate the effectiveness of our unified training with the three proposed objectives. Specifically, we compare the performance on representation learning tasks after adapting the LLM using different combinations of the objectives. The results are presented in Table 9. We find that while MNTP is the only objective that explicitly trains the model for better token-level representations, adding MSG marginally improves performance on word-level tasks. We conjecture that MSG, being closer to the original next-token prediction objective of the base LLM, acts as a regularizer and helps prevent extreme variations in the token representations produced by the model. For sentence-level tasks, which use the SSCL objective on the last token's representation, we observe no noticeable benefit or drawback from including MNTP and MSG. This shows that we can add token-level representation learning and infilling capabilities to the model without hampering performance on sentence-level tasks. We conjecture that the effects of unified training on sentence-level tasks are not evident from Table 9 due to the separation of sentence-level representation learning from token-level representation learning and generation, achieved by using only the last token's output as the sentence encoding.

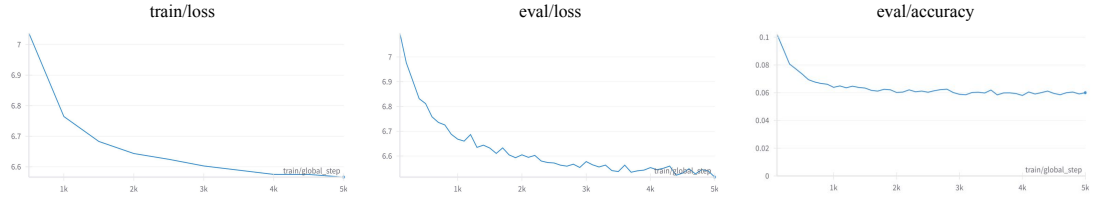
D Comparing MTP and MNTP Objectives

Traditionally, language models for representation learning are trained to predict a masked token at position l using the output at position l in the final hidden states (Devlin et al., 2019; Liu et al., 2019; Lan et al., 2019; Sanh et al., 2019). This approach is logical because the residual connections in the transformer block incorporate the l^{th} token's input representation into its output representation.

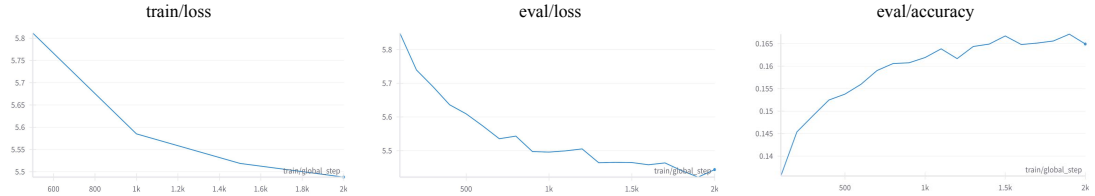
We conducted an experiment to test whether we can use LoRA to adapt the base LLM for l-to-l prediction (similar to BERT). The training curves for masked token prediction (MTP) and masked number token prediction (MNTP) are shown in Figure 6. As illustrated, with MTP, the loss converges, but the evaluation accuracy for masked token prediction decreases. This likely occurs because the base model is trained to predict the $(l+1)^{th}$ token at position l, and shifting to l-to-l prediction introduces a significant distributional shift that the model may struggle to accommodate swiftly. Thus, overall, we

Training Objectives	Chunking	NER	POS	STS12	STS13	STS14	STS15	STS16
MNTP	92.44	98.11	93.18	–	–	–	–	–
SSCL	–	–	–	**69.06**	84.53	**78.07**	84.09	78.52
MNTP + MSG	92.51	98.20	**93.38**	–	–	–	–	–
SSCL + MSG	–	–	–	68.46	84.52	77.33	**84.35**	79.17
MNTP + SSCL + MSG	**92.64**	**98.31**	93.34	67.98	**84.66**	77.67	84.17	**79.44**

Table 9: Ablation analysis of the proposed training objectives to assess the potential benefits of unified training.



(a) Masked Token Prediction



(b) Masked Next Token Prediction

Figure 6: Training curves for MTP and MNTP objectives. When using MTP, model's performance on masked token prediction (measured using eval set accuracy) deteriorates over training iterations.

find that MNTP is a more effective objective for converting a decoder-only LLM into a text encoder. Additionally, MNTP aligns well with the causal MSG objective and paves the way for a unified text generator and encoder.

Task	Instruction
STS*	Retrieve semantically similar text
BiorxivClusteringP2P	Identify the main category of Biorxiv papers based on the titles and abstracts
BiorxivClusteringS2S	Identify the main category of Biorxiv papers based on the titles
TwentyNewsgroupsClustering	Identify the topic or theme of the given news articles
MedrxivClusteringP2P	Identify the main category of Medrxiv papers based on the titles and abstracts
MedrxivClusteringS2S	Identify the main category of Medrxiv papers based on the titles

Table 10: Instructions used for getting sentence encoding for the different sentence-level tasks. "STS*" refers to all the STS tasks.

Method	Infilled Line
John was at the gym. _____ He couldn't get the barbell off his chest. He was too embarrassed to call for help. John suffocated as the weights crushed his chest.	
Blank Infilling	John was at the gym.
Contextual	John was at the gym, trying to lift a particularly heavy barbell, when he suddenly found himself unable to breathe and his face turned blue.
Prefix-Suffix	John suffocated as the weights crushed his chest.
Line-by-Line	John was at the gym, sweating profusely.
Toni took a trip to the local fair with her family. _____ The balloon was red and star shaped. Unfortunately it slipped out of her hands. She was sad to see it go.	
Blank Infilling	" She watched as it soared high up in the sky."tt = 0 while t < 10: t += 0.
Contextual	Toni took a trip to the local fair with her family and won a red and star-shaped balloon as a prize.
Prefix-Suffix	Toni took a trip to the local fair with her family and was excited to see a red star-shaped balloon.
Line-by-Line	Toni took a trip to the local fair with her family.
I was resting on my couch. _____ They raised their water guns. I immediately ran. I went outside and sprayed them with the hose when they followed.	
Blank Infilling	They raised their water guns at me.
Contextual	I heard the sound of water splashing and knew they were coming for me.
Prefix-Suffix	I immediately ran.
Line-by-Line	I was resting on my couch, bored out of my mind.
Kaylee owned a ferret who always ran free around the house. _____ She put flyers all over the neighborhood. There was a knock at the door three days later. Someone found her ferret.	
Blank Infilling	She was overjoyed to see her ferret wagging its tail when she opened the door.
Contextual	Kaylee put flyers all over the neighborhood.
Prefix-Suffix	Kaylee found her ferret three days after putting flyers all over the neighborhood.
Line-by-Line	She found the ferret hiding under the bed.
Cody really wanted a smart phone. _____ Cody protested that all of his friends had their own phones. His parents phoned some of the other parents. His parents learned that Cody had been exaggerating.	
Blank Infilling	Cody really wanted a smart phone.
Contextual	Cody's parents discovered that he had been lying about wanting a smartphone when they phoned some of the other parents and found out that none of their children had ever mentioned anything about Cody wanting a new phone.
Prefix-Suffix	Cody's parents phoned some of the other parents.
Line-by-Line	Cody really wanted a smart phone because he thought it would make him cooler.
Darla was afraid of spiders. _____ She screamed and jumped away. She yelled for her father to come in and help. Her father came and removed the spider.	
Blank Infilling	Darla was afraid of spiders.
Contextual	Darla was afraid of spiders because she had a bad experience with one when she was younger.
Prefix-Suffix	Darla was afraid of spiders and screamed and jumped away.
Line-by-Line	Darla was afraid of spiders.

Table 11: Zero-shot sentence infilling examples using Llama-2-7B with different prompting strategies.

Method	Infilled Line
John was at the gym. _____ He couldn't get the barbell off his chest. He was too embarrassed to call for help. John suffocated as the weights crushed his chest.	
Uni-Llama	He was on the treadmill, running at a steady pace.
FS-Llama	He was doing bench presses.
MAGNET	John was doing bench presses when he accidentally dropped the barbell on top.
Toni took a trip to the local fair with her family. _____ The balloon was red and star shaped. Unfortunately it slipped out of her hands. She was sad to see it go.	
Uni-Llama	She won a prize for her drawing of a cat.
FS-Llama	She won it in a game of darts.
MAGNET	She was so excited to see the balloon vendor and bought a red one.
I was resting on my couch. _____ They raised their water guns. I immediately ran. I went outside and sprayed them with the hose when they followed.	
Uni-Llama	I was feeling a bit under the weather, so I decided to take a break.
FS-Llama	My neighbor's kids were playing tag in the backyard.
MAGNET	I heard a noise and looked out the window and saw two boys with water guns.
Kaylee owned a ferret who always ran free around the house. _____ She put flyers all over the neighborhood. There was a knock at the door three days later. Someone found her ferret.	
Uni-Llama	She loved her ferret, but she knew that ferrets have a tendency to get into things they shouldn't.
FS-Llama	She put up a reward for the safe return of her ferret.
MAGNET	She was devastated when she lost him and searched everywhere for him.
Cody really wanted a smart phone. _____ Cody protested that all of his friends had their own phones. His parents phoned some of the other parents. His parents learned that Cody had been exaggerating.	
Uni-Llama	He had been saving up his allowance for months, but he just couldn't.
FS-Llama	Cody's friends told him they had been saving up for their phones.
MAGNET	His parents refused to buy him one because they thought it was too expensive and dangerous.
Darla was afraid of spiders. _____ She screamed and jumped away. She yelled for her father to come in and help. Her father came and removed the spider.	
Uni-Llama	She had always been afraid of them, even as a child.
FS-Llama	She saw a spider crawling on her bed.
MAGNET	She was in her room when she saw a spider on her bed.
A strange structure appeared on the outside of Connie's house one day. _____ She called an exterminator to come get them. The exterminator came and got rid of the wasps quickly. Connie was happy that the nest was gone.	
Uni-Llama	It was a large, glowing crystal that seemed to pulse with.
FS-Llama	She saw a wasp flying around her kitchen window.
MAGNET	Connie noticed that there were wasps in the house and outside.

Table 12: Sentence infilling examples using Llama-2-7B and its adaptation using MAGNET. Uni-Llama refers to the unidirectional model that only considers the left context, and FS-Llama is the few-shot variant that learns to use the full context to generate the infilling.

Model	Generated Text
	Initial phrase: *The film was well received*
Llama	*The film was well received* by critics and audiences alike, with many praising its unique blend of humor and heart. Here are some of the key reasons why critics and audiences loved "The Big Sick": 1. Unique blend of humor and heart: The film's ability to balance humor and heart was widely praised by critics and audiences. The film's lighthearted moments are often juxtaposed with more serious and emotional scenes, creating a film that is both funny and heartwarming. 2. Authentic representation: The film is ...
LLM2Vec	*The film was well received* by critics and audiences alike. The film was well received by critics and audiences alike. The film was well received by critics and audiences alike. The film was well received by critics and audiences alike. The film was well received by critics and audiences alike. The film was well received by critics and audiences alike. The film was well received by critics and audiences alike. The film was well received by critics and ...
MAGNET	*The film was well received* by critics and audiences alike and was nominated for several awards including the Academy Award for Best Visual Effects and the Saturn Award for Best Science Fiction Film. The film was also a commercial success and grossed over $100 million at the box office. = = Plot = = In the year 2018, a meteorite crashes into the Pacific Ocean, causing a massive tsunami that destroys most of the world's coastal cities. The survivors of the disaster band together ...
BERT	*The film was well received* and the " " " " " " " " " " " " " " " are " " are " are " are " are " are are " are are the are the the the the the the the the the the the the the so so and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and ...
	Initial phrase: *After 1805, the exploits of*
Llama	*After 1805, the exploits of* the British East India Company in India became increasingly controversial, and the company faced growing criticism from both the British public and Indian nationalists. Here are some of the key events that contributed to the decline of the British East India Company: 1. The Battle of Buxar (1757): The British East India Company defeated the French and their Indian allies in the Battle of Buxar, which gave them control over Bengal. This marked the beginning of the company's expansion in India. 2. The Battle of Plassey ...
LLM2Vec	*After 1805, the exploits of* the French army in Egypt continued to be successful. In 1806, the French army defeated the Ottoman army at the Battle of Aboukir. In 1807, the French army defeated the Ottoman army at the Battle of Austerlitz. In 1808, the French army defeated the Ottoman army at the Battle of Wadi Abu Jarai. In 1809, the French army defeated the Ottoman army at the Battle of Roli. In 1810, the French army defeated ...
MAGNET	*After 1805, the exploits of* the pirates of the Caribbean became less frequent and the British Navy began to take a more active role in suppressing piracy. The British Navy began to patrol the Caribbean and the Spanish Main, and the pirates were forced to flee to other areas. The pirates of the Caribbean were not completely eliminated, however, and piracy continued to be a problem in the Caribbean until the mid 19th century. The last pirate to be executed in the Caribbean was Captain Kidd, who was hanged ...
BERT	*After 1805, the exploits of* the the - the - way - to - the - other - to - the - other - to - the - other - to - the - other -. -. -. -. -. -. -. -. -. -. -. -. -. -. -. -. -. -. -. - " - " - " " - " " " " " " " " " " " " - " " - " " " " " " " " " " and the " - " " " - " " - " - " " _ " " _ " " " " " " " " "

Table 13: Text generated using greedy decoding with BERT-Base, Llama-2-7B, and its adaptations using LLM2Vec and MAGNET. To generate text from BERT, we recursively add a mask token at the end of the input sentence and use the model to predict the mask.