Bidirectional GPT

Anonymous ACL submission

Abstract

Mainstream Large Language Models (LLMs) built on the GPT (Generative Pre-trained Transformer) architecture can only read and generate text in a left-to-right direction. This limitation prevents these models from comprehensively processing the training data as a whole and from directly deriving suitable prompts from given responses. Drawing inspiration from the global understanding capability of Bi-LSTM, we introduce Bi-GPT, an enhanced version of the standard GPT architecture that incorporates reverse generation capabilities. Instead of altering the underlying architecture or adding any extra parameters, Bi-GPT utilizes dual learning with both forward and backward data streams to enable bidirectional generation. To reduce the training cost, we design a two-stage pretraining strategy that can transform any existing LLM into the bidirectional version. We train Bi-GPT with different scales and conduct a comprehensive set of experiments, including conventional forward response generation, reverse instruction generation, and token classification tasks, to thoroughly validate its capabilities. The results show that the incorporation of bidirectional training data improves the forward generation capability (+8% on 5 datasets) and overall performance in token classification tasks. Furthermore, Bi-GPT effectively bridges the gap between responses and prompts, allowing for the exploration of potential prompt and meta-prompt generation from a single instance. In summary, Bi-GPT significantly expands the application scenarios and capabilities of GPT without adding any new parameters. [1]

1 Introduction

Large Language Models (LLMs), particularly those based on the GPT architecture, have demonstrated remarkable capabilities in text generation and comprehension tasks. These models operate in an autoregressive manner, generating text token by token from left to right. While this approach has proven effective for many tasks, it inherently limits the model's ability to process and understand text in a holistic manner. This unidirectional processing restricts the model's capacity to capture global context. Additionally, it hinders the model's ability to effectively utilize the training data. To address these limitations, several approaches have been proposed. For instance, researchers have explored techniques for prompt optimization (Wei et al., 2022; Kojima et al., 2022; Zhang et al., 2022), or techniques to improve the generation performance through self-consistency (Wang et al., 2022), self-reflection (Shinn et al., 2024; Madaan et al., 2024) and backward reasoning (Weng et al., 2022; Jiang et al., 2024). These methods aim to guide models through step-by-step reasoning or to iteratively refine their prompts and outputs. However, while these techniques improve performance on specific tasks, they do not fundamentally alter the unidirectional nature of GPT-based models.

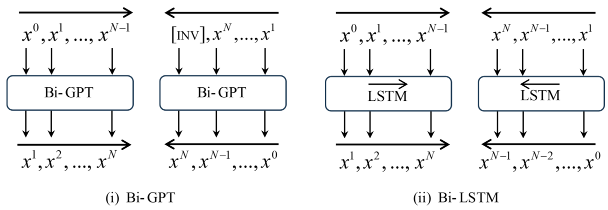

Figure 1: Comparison of Bi-GPT and Bi-LSTM. Bi-GPT adopts bidirectional data pretraining within a single model, whereas Bi-LSTM trains two separate models for forward and reverse instances.

Drawing inspiration from the bidirectional capabilities of models like Bi-LSTM (Schuster and Paliwal, 1997; Zhang et al., 2015), which process data in both forward and backward directions, we propose Bi-GPT, a GPT architecture incorporating reverse generation capabilities. As illustrated in 1, Bi-GPT enables bidirectional generation while

[1]The code, data, and pre-trained model will be publicly available in the final version.

maintaining the same architecture and parameter count, unlike Bi-LSTM, which necessitates distinct models for forward and backward inferences. This is accomplished through a novel dual-learning approach that leverages both forward and backward data streams. To reduce training costs, we propose a two-step pretraining strategy, validated on backbones of varying sizes, which enables the transformation of any existing unidirectional LLM into a bidirectional model. Through this combined framework of reverse pretraining and dual-task finetuning, we aim to endow LLMs with enhanced forward and backward generation capabilities, resulting in more robust and accurate performance across diverse tasks. We evaluate Bi-GPT through a comprehensive set of experiments, including conventional forward response generation, reverse instruction generation, and token classification tasks. Our results demonstrate that incorporating bidirectional training data significantly enhances the model's forward generation capability, with an average improvement of $+8\%$ across five datasets. Meanwhile, Bi-GPT bridges the gap between responses and prompts, enabling the exploration of prompt and meta-prompt generation from a single instance. Finally, Bi-GPT excels in token classification tasks, showcasing its ability to capture global contextual information. Our contributions can be summarized as follows:

- To the best of our knowledge, we are the first to propose a bidirectional GPT architecture that enables global memory and bidirectional generation capabilities without additional parameters.
- We propose a training strategy that transforms any existing forward-pretrained LLM into a bidirectional architecture at minimal cost.
- Bi-GPT largely enhances the forward generation capabilities, enables automatic prompt optimization through backward generation, and benefits token classification through providing global token representation.

2 Related Work

2.1 Architectures of LLMs

Early attempts at language modeling and sequence-to-sequence tasks were dominated by RNN-based architectures (Hochreiter, 1997; Zaremba, 2014; Chung et al., 2014). However, their sequential nature made it difficult to capture long-range dependencies and limited training efficiency. The transformer architecture (Vaswani, 2017) revolutionized this landscape by replacing recurrence with multi-head self-attention, allowing parallel computation over input tokens and more effective global context modeling. Building on this foundation, BERT (Devlin, 2018) leverages the Transformer's encoder with bidirectional attention, excelling in comprehension-oriented tasks such as classification and question answering. By contrast, GPT (Radford et al., 2019; Brown et al., 2020) adopts the Transformer's decoder with a unidirectional (left-to-right) attention mechanism, making it highly effective for generative tasks like text completion and dialogue.

2.2 Prompt Engineering

Prompt engineering (Liu et al., 2023a) aims to maximize the capabilities of LLMs by designing effective prompts, either as natural language instructions or learned vector representations. Approaches range from manual design to automated generation, including soft prompt learning (Lester et al., 2021), instruction tuning (Wei et al., 2021; Sanh et al., 2021), and retrieval-augmented generation (RAG)(Lewis et al., 2020). In-context learning and few-shot prompting refine inputs by incorporating retrieved demonstrations (Rubin et al., 2021; Su et al., 2022) or reasoning-based exemplars such as Chain-of-Thought (CoT) prompting (Wei et al., 2022). Zero-shot CoT removes the need for manual demonstrations by leveraging intrinsic model reasoning (Kojima et al., 2022), while automatic CoT generation (Auto-CoT) extends this by clustering diverse examples and generating reasoning chains (Zhang et al., 2022). More advanced methods include task decomposition (Zhou et al., 2022), self-consistency decoding (Wang et al., 2022; Li et al., 2022), and multi-agent collaboration for enhanced reasoning (Du et al., 2023; Liu et al., 2023b). Reinforcement learning refines prompts based on feedback (Shinn et al., 2024).

2.3 Dual Learning and Reverse Thinking

The core concept of dual learning is to leverage the primaldual structure inherent to a task, such as the bidirectional relationship in machine translation (Sennrich, 2015). This duality acts as a form of regularization during training, thereby enhancing performance across both tasks (Chen et al., 2024). Several works have been proposed to leverage backward reasoning to verify the chain-of-thought during the inference stage (Weng et al., 2022; Jiang et al., 2024). REVTHINK (Chen et al., 2024) incorporates backward question generation and backward reasoning as forms of regularization to improve reasoning capabilities and maintains the same test-time efficiency as zero-shot prompting.

3 Method

3.1 Conventional LLM

Large Language Models (LLMs) training generally consists of two stages: pretraining and fine-tuning. In the pretraining stage, LLMs are trained on large-scale textual data to learn general language representations. The training data typically consists of forward text sequences:

$$\overrightarrow{\mathrm{X}} = (x^0, x^1, \ldots, x^N). \tag{1}$$

where $\overrightarrow{\mathrm{X}}$ represents a sentence with standard left-to-right word order, and N is the total number of tokens. The model learns to predict the next token based on the preceding context:

$$p(x^0, x^1, \ldots, x^N) = \prod_{i=0}^{N} p(x^i | x^0, x^1, \ldots, x^{i-1}). \tag{2}$$

In the fine-tuning stage, standard LLMs are further trained on instruction-following data, typically structured as paired question-to-answer pairs $\overrightarrow{\mathrm{Q}} \rightarrow \overrightarrow{\mathrm{A}}$.

3.2 Bidirectional GPT

To enhance bidirectional learning, Bi-GPT introduces additional modifications in both pretraining and fine-tuning stages. During pretraining, Bi-GPT incorporates reverse text sequences:

$$\overleftarrow{\mathrm{X}} = ([\mathrm{INV}], x^N, x^{N-1}, \ldots, x^0). \tag{3}$$

where $[\mathrm{INV}]$ is a special token indicating reverse generation. The model is trained to predict tokens in both forward and reverse directions, with the conditional probability formulated as:

$$\begin{aligned} p(x^0, x^1, \ldots, x^N) = &\prod_{i=0}^{N} p(x^i | x^{i+1}, \ldots, x^N, [\mathrm{INV}]) \\ &\times \prod_{i=0}^{N} p(x^i | x^0, x^1, \ldots, x^{i-1}). \end{aligned} \tag{4}$$

Theoretically, this strategy supports full-parameter pretraining from scratch. However, such pretraining is computationally expensive and impractical due to the massive tokens required. To address this challenge, we propose a two-step pretraining strategy that first applies reverse pretraining A.2 and then combines both forward and reverse pretraining to convert a forward-trained LLM into a bidirectional one. During fine-tuning, Bi-GPT extends standard instruction-following data by incorporating dual learning. In addition to the standard question-to-answer sequences, Bi-GPT constructs reversed pairs by swapping and inverting both the question and the answer $\overleftarrow{\mathrm{A}} \rightarrow \overleftarrow{\mathrm{Q}}$. We prepend $[\mathrm{INV}]$ to the start and append $[\mathrm{PROMPT}]$ to the end of the reversed answer.

3.3 Pretraining Strategy

We compared three potential pretraining strategies as follows:

- Direct bidirectional pretraining: Start with bidirectional data using the forward pretrained model as initialization.
- Transition based on PPL intersection: First pretrain on reverse sequence data. When the perplexities (PPLs) of the forward and reverse sequences intersect, switch to bidirectional pretraining.
- Transition based on reverse PPL convergence: First pretrain on reverse sequence data until the reverse PPL converges. Then transition to bidirectional pretraining.

As illustrated in Fig. 2, it is evident that all three approaches can quickly restore the PPL for forward data to a lower level, after which the PPLs for both forward and reverse data steadily decrease and eventually converge. Based on a comparison of the final PPL values after convergence, the average PPL for the first scheme is 14.943, for the second scheme is 15.118, and for the third scheme is 14.073. The scheme with the lowest average PPL was selected as the optimal training strategy. This strategy involves first pre-training with reverse data until its PPL converges, followed by full-parameter pre-training with bidirectional data.

3.4 Application

Bi-GPT supports multiple applications, including forward response generation, prompt generation

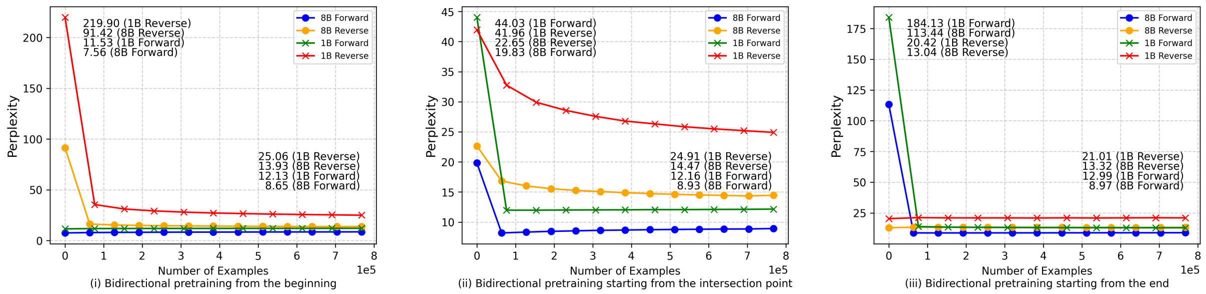

Figure 2: Perplexity of forward and reverse text during three pretraining strategies. (i)–(iii) present PPL during bidirectional pretraining from different starting points: (i) from the beginning, (ii) from the PPL intersection of forward and reverse text, and (iii) after reverse text PPL converges.

from a single instance, meta prompt generation, and token classification.

Forward Generation In the standard forward generation task, Bi-GPT generates responses given a normal forward question input, similar to conventional language models.

One-Instance Prompt Generation Additionally, Bi-GPT enables prompt generation from one instance by producing reverse prompts from reversed input text, as described in Eq. 5.

$$[\text{INV}], \overleftarrow{\text{A}}, [\text{PROMPT}] \xrightarrow{\text{Bi-GPT}} \overleftarrow{\text{Q}} \tag{5}$$

This capability allows for the automatic exploration of effective prompts based on a single instance.

Meta Prompt Generation Furthermore, Bi-GPT can generate meta-prompts by combining reverse answers and reverse questions as follows:

$$[\text{INV}], \overleftarrow{\text{A}}, [\text{PROMPT}], \overleftarrow{\text{Q}} \xrightarrow{\text{Bi-GPT}} \overleftarrow{\text{Meta}} \tag{6}$$

Token Classification For token classification tasks, such as Named Entity Recognition (NER), Bi-GPT leverages its global bidirectional representation to enhance prediction performance. Given a sequence of tokens of length N, predictions are made by combining logits from both forward and reverse directions as follows:

$$\begin{aligned} \text{Logits}_i = &\ \text{Bi-GPT}(x^0, \ldots, x^i) + \\ &\ \text{Bi-GPT}([\text{INV}], x^N, \ldots, x^i) \end{aligned} \tag{7}$$

where Bi-GPT incorporates a linear layer *score* to map the hidden size dimension to the number of categories in NER classification tasks. By integrating forward and reverse logits, Bi-GPT significantly improves token-level prediction accuracy.

4 Experiment

4.1 Experimental Setup

In this study, we utilized LLaMA-3 (Dubey et al., 2024) as the foundational model and explored LLMs with varying parameter scales, specifically from 1B to 8B parameters. During the pre-training phase, we employed 32 Huawei Ascend 910A 32GB NPUs for full-parameter training of the LLMs. In the fine-tuning stage, we utilized 4 Nvidia RTX 3090 24GB GPUs to perform LoRA fine-tuning on the pre-trained LLM.

4.2 Dataset

Our pretraining dataset consists of 9 million samples from Common Crawl, a publicly available web archive covering diverse sources like news, blogs, and academic pages. We evaluate the forward generation ability on five tasks, including the commonsense reasoning dataset ARC-challenge (Clark et al., 2018), the math reasoning dataset MATH (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), the tabular data reasoning dataset TabMWP (Lu et al., 2022), and the natural language inference dataset ANLI (Nie et al., 2019). To assess prompt optimization capability of Bi-GPT, we conducted experiments on the instruction-following dataset Alpaca-GPT4 (Peng et al., 2023). This dataset was chosen to evaluate the model's generation performance rather than accuracy-focused benchmarks mentioned above. For token classification tasks, we conducted experiments on two benchmark datasets: CoNLL2003 (Sang and De Meulder, 2003) and CoNLL++ (Wang et al., 2019). Additionally, samples for prompt generation tasks were randomly selected from the aforementioned datasets.

Model Size	Method	ARC	MATH	GSM8K	TabMWP	ANLI	Avg.
1B	Llama3	34.35	17.22	22.97	75.72	48.25	33.77
	Llama3 w/ dual learning	36.37	17.24	23.43	80.97	48.67	41.34
	Bi-GPT w/o dual learning	34.73	17.22	23.05	74.66	48.75	39.68
	Bi-GPT	**36.52**	**17.35**	**24.41**	**83.58**	**48.92**	**42.16**
8B	Llama3	74.91	21.75	54.43	93.62	58.83	60.71
	Llama3 w/ dual learning	75.01	22.31	61.03	94.21	61.25	62.76
	Bi-GPT w/o dual learning	75.09	26.09	58.91	93.47	60.58	62.83
	Bi-GPT	**75.09**	**26.25**	**61.41**	**95.68**	**62.00**	**64.09**

Table 1: Forward reasoning performance on five held-in datasets. Llama3 served as the baseline model, trained solely on $\overrightarrow{\mathrm{Q}} \rightarrow \overrightarrow{\mathrm{A}}$ datasets, Llama3 w/ dual learning was augmented with $\overrightarrow{\mathrm{A}} \rightarrow \overrightarrow{\mathrm{Q}}$ training pairs. Bi-GPT w/o dual learning refers to the Bi-GPT model that excluded $\overleftarrow{\mathrm{A}} \rightarrow \overleftarrow{\mathrm{Q}}$ training data. Llama3 employs single-direction fine-tuning, while Bi-GPT can benefit from additional dual learning.

Inference Mode	Model	ROUGE-1	ROUGE-2	ROUGE-1	BLEU Score
$\overrightarrow{\mathrm{Q}} \rightarrow \overrightarrow{\mathrm{A}}'$	Llama3 w/ dual learning	0.403	0.208	0.317	0.148
	Bi-GPT	0.401	0.211	0.314	0.152
$\overrightarrow{\mathrm{A}} \rightarrow \overrightarrow{\mathrm{Q}}' \rightarrow \overrightarrow{\mathrm{A}}'$	Llama3 w/ dual learning	0.392	0.202	0.306	0.144
$\overleftarrow{\mathrm{A}} \rightarrow \overleftarrow{\mathrm{Q}}' \rightarrow \overrightarrow{\mathrm{A}}'$	Bi-GPT	**0.413**	**0.227**	**0.328**	**0.168**

Table 2: Evaluation metrics between the generated answer A' and the ground truth answer A. Superscript $\leftarrow$ indicates reversed text, while $\rightarrow$ indicates forward text. The first two rows present results tested on the original question Q from the test set, while the last two rows was tested on the model-generated question Q'.

4.3 Affections on Forward Generation

We trained Bi-GPT on datasets from five different domains and conducted forward generation experiments, where the questions in the original dataset were used as prompts. From the results presented in Table 1, it is evident that the incorporation of bidirectional training improves forward generation performance across most of the datasets for both 1B and 8B model scales. Specifically, models trained with dual learning achieve higher average scores than the standard models. The improvement is more pronounced in complex tasks such as GSM8K and TabMWP, suggesting that dual learning is particularly beneficial for complex tasks.

4.4 One-Instance Prompt Optimization

To evaluate Bi-GPT's capability in reverse generation, we conduct experiments assessing its ability to generate questions from given answers and reconstruct the original responses. This validates the model's reversibility and highlights its effectiveness in single-instance prompt generation. Additionally, we explore its potential to generate higher-level instructions from a single instance. The following subsections detail our evaluation of reversibility and meta-prompt generation.

4.4.1 Validation of Reversibility

In order to validate the reversibility of Bi-GPT, we conducted experiments to evaluate its performance in reverse inference and subsequent forward generation. This capability is crucial for demonstrating the model's bidirectional understanding and its potential for prompt optimization in scenarios where the original input may be suboptimal or unavailable. As shown in Table 2, Llama3 w/ dual learning is an extension of the baseline Llama3 model, incorporating a dual learning mechanism that utilizes both answer-question pairs $\overrightarrow{\mathrm{Q}} \rightarrow \overrightarrow{\mathrm{A}}$ and question-answer pairs $\overrightarrow{\mathrm{A}} \rightarrow \overrightarrow{\mathrm{Q}}$ for training. During the testing phase, we first allowed Bi-GPT to infer the reverse question $\overleftarrow{\mathrm{Q}}'$ from the reverse answer as described in Eq. 5, then generate the forward answer based on the flipped forward question and compute the correlation metrics with the ground truth answer. While the Llama3 with dual learning model directly infers the forward question $\overrightarrow{\mathrm{Q}}'$ from the forward ground truth answer and further generates the forward answer based on it, then calculates the correlation metrics as aforementioned.

Compared to Llama3 with dual learning, the questions generated by Bi-GPT are even more likely to yield answers closer to the ground truth

Benchmark	Model Size	Method	Accuracy	Precision	Recall	F1 Score
CoNLL2003	1B	Llama3	0.951	0.659	0.711	0.684
		Bi-GPT	**0.973**	**0.823**	**0.855**	**0.839**
	8B	Llama3	0.958	0.701	0.755	0.727
		Bi-GPT	**0.976**	**0.848**	**0.882**	**0.864**
CoNLL++	1B	Llama3	0.957	0.695	0.742	0.718
		Bi-GPT	**0.974**	**0.837**	**0.860**	**0.849**
	8B	Llama3	0.958	0.701	0.750	0.725
		Bi-GPT	**0.979**	**0.865**	**0.890**	**0.878**

Table 3: Token classification results of unidirectional and bidirectional LLMs on CoNLL2003 and CoNLL++ benchmarks.

answers than the original test set questions according to Table 2. This indicates that Bi-GPT can be utilized to generate prompts that are more likely to elicit specific content for unknown inputs. The comparative results highlight the strong reversibility property of Bi-GPT. Specifically, Bi-GPT demonstrates the ability to effectively infer corresponding questions from given answers. We present some examples of prompts generated by Bi-GPT and the corresponding responses produced using these prompts, alongside the responses generated using the original test set prompts, as shown in Table 4.

4.4.2 Meta Prompt Generation

Bi-GPT is capable of generating meta prompt from a single instance, leveraging its bidirectional capability to establish a stronger connection between responses and prompts. As formulated in Eq. 6, the model takes a reversed answer and its corresponding reversed question as input, producing a meta prompt that encapsulates key contextual information. This capability enables Bi-GPT to dynamically construct meta prompts that refine or guide subsequent text generation, making it particularly useful for exploring prompt generation with one single instance. Meta-prompt generation examples are presented in Table 5. The left column categorizes the meta prompts generated by Bi-GPT, while the right column provides corresponding meta-question-answer examples. To clearly illustrate the components, different sections are highlighted using three distinct colors: the cyan section represents the original prompt, the blue section denotes the original answer, and the red section highlights the meta-prompt generated by Bi-GPT based on the answer-prompt pair.

4.5 Bi-GPT for Token Classification

To evaluate Bi-GPT's ability to capture contextual information, we conduct token classification experiments on the CoNLL2003 and CoNLL++ benchmarks. Compared to the unidirectional Llama3, Bi-GPT achieves significant improvements across all metrics. For instance, on CoNLL2003, Bi-GPT (1B) attains an F1 score of 0.839, surpassing Llama3's 0.684. Similar gains are observed for the 8B model, with Bi-GPT achieving an F1 score of 0.864 compared to Llama3's 0.727. These results demonstrate that Bi-GPT's bidirectional training enhances its ability to understand token-level context, making it more effective for tasks requiring fine-grained semantic understanding.

5 Conclusion

In this work, we propose Bi-GPT, an innovative bidirectional enhancement of the conventional GPT architecture that does not require additional parameters. By leveraging dual learning with both forward and backward data streams, Bi-GPT achieves bidirectional generation capabilities, providing a significant improvement in the processing of training data as a whole. Our approach not only enhances the model's forward generation capabilities, with an $+8\%$ improvement across five datasets, but also enables the exploration of prompt generation from a single instance. Furthermore, Bi-GPT demonstrates significant benefits for token classification tasks by providing a richer global token representation. Overall, our results showcase that Bi-GPT expands the range of possible applications for GPT-based models while maintaining a low computational cost, making it an attractive option for future developments in the field of large language models.

Original Prompt	Corresponding Answer	Prompt By Bi-GPT	Corresponding Answer	Ground Truth
Provide two valid solutions to the given mathematical equation. 2x + 3 = 5\n	Two valid solutions to the given mathematical equation are x = 1 and x = -2.	Provide two valid solutions to the following equation: 2x + 3 = 5\n	Solution 1: \nSubtracting 3 from both sides of the equation, we get:\n2x + 3 - 3 = 5 - 3\n2x = 2\nx = 1\n\nSolution 2: \nDividing both sides of the equation by 2, we get:\n(2x + 3) / 2 = 5 / 2\n2x / 2 + 3 / 2 = 5 / 2\nx = 1	Sure. Here are two valid solutions to the given equation:\n\nSolution 1:\n2x + 3 = 5\nSubtract 3 from both sides of the equation: \n2x + 3 - 3 = 5 - 3\nSimplifying, we have:\n2x = 2\nDivide both sides by 2: \n(2x)/2 = 2/2\nSimplifying, we have:\nx = 1\nSo one solution to the equation is x = 1.\n\nSolution 2:\nAnother way to solve the equation is to move all terms to one side of the equation: \n2x + 3 - 5 = 0\nSimplifying, we have:\n2x - 2 = 0\nDividing both sides by 2: \n2x/2 - 2/2 = 0\nSimplifying, we have:\nx - 1 = 0\nAdding 1 to both sides:\nx - 1 + 1 = 0 + 1\nSimplifying, we have:\nx = 1\nSo the other solution to the equation is also x = 1.
Edit the content and make sure the reader understands what the writer is trying to explain. People who take part in sustainablity put effort into their lifestyle to make a positive contribution to society.\n	Individuals who participate in sustainability strive to incorporate eco-friendly practices into their daily routines, with the goal of making a positive impact on the environment and society as a whole.	Generate a sentence about individuals who engage in sustainable practices.	Individuals who engage in sustainable practices are making a positive impact on the environment and helping to create a better future for themselves and future generations.	Individuals who engage in sustainable practices make a conscious effort to adopt a lifestyle that positively contributes to society and helps preserve the environment for future generations.
Write a CSS code to apply a few style rules on a webpage. Heading font-size: 18px\nBackground Colour: #f7f :\n	Here is a CSS code that applies the style rules you mentioned:\n\n```css\nh1 \n font-size: 18px;\n\nbody \n background-color: #f7f;\n\n```	Write a CSS code to apply the specified style rules. We want the body of the page and all heading elements on the page to have a background color of #f7f7f7 and a font size of 18px.\n	Here is the CSS code to apply the specified style rules:\n\n```css\nbody, h1, h2, h3, h4, h5, h6 \n background-color: #f7f7f7;\n font-size: 18px;\n\n```	Here is a sample CSS code that you can use to apply the specified style rules on a webpage:\n\n```css\nbody \n background-color: #f7f7f7;\n\nh1, h2, h3, h4, h5, h6 \n font-size: 18px;\n\n```\n\nIn this code, the `body` selector targets the entire webpage, and the `background-color` property sets the background color of the page to the specified hexadecimal value `#f7f7f7`.\n\nThe `h1`, `h2`, `h3`, `h4`, `h5`, and `h6` selectors target all heading elements on the page, and the `font-size` property sets their font size to `18px`.
Identify the relationship between \"cat\" and \"yarn\". \n	\"Cat\" and \"yarn\" are two words that are not directly related to each other. \"Cat\" refers to a small, carnivorous mammal with soft fur and a short, tapered tail, while \"yarn\" refers to a long, thin strand of fibers twisted together, often used for knitting or weaving. The only connection between the two words is that they are both commonly used in the context of crafts and hobbies, where cats are often depicted as participants or yarn is used as a material.	Describe the relationship between \"cat\" and \"yarn\".\n	\"Cat\" and \"yarn\" are two words that are not directly related to each other. \"Cat\" is a noun that refers to a small, carnivorous mammal with soft fur and a short, tapered tail. \"Yarn\" is a noun that refers to a continuous strand of twisted fibers, often used for knitting or weaving. The only connection between the two words is that cats are known to play with yarn, using it as a toy or creating a ball of yarn.	The relationship between \"cat\" and \"yarn\" is often one of play. Many cats find yarn balls interesting and entertaining to play with, as they can chase, paw, and roll it around with their paws. This interaction can provide a form of physical exercise and mental stimulation for the cat. However, it should be noted that playing with yarn can be dangerous for cats, as there is a risk of ingesting the yarn, which can cause serious health issues.
Write a few sentences to describe a scene. Night, empty street.\n	The night is dark and quiet as the empty street stretches out before me. The only sound is the distant hum of traffic in the distance. The streetlights cast a warm glow on the deserted sidewalks, illuminating the occasional stray cat or discarded piece of trash. The air is cool and still, and the only movement is the gentle sway of the streetlights as they dance in the breeze.	Imagine you are writing a script for a movie. The scene focuses on an empty street. Write a description of the scene.\n	The camera pans down a long, empty street. The asphalt is smooth and black, with a faint sheen from the morning dew. The streetlights cast a warm, yellow glow, illuminating the scene. There is not a single car or person in sight. The only sound is the distant hum of traffic from the nearby highway. The camera slowly pans up, revealing a tall, abandoned building in the distance. The building is dilapidated, with peeling paint and broken windows. The camera then pans back down, revealing the entire street, which is completely empty. The scene is somber and eerie, with a sense of loneliness and abandonment.	It is night, and the street is eerily empty. The streetlights cast a soft, yellow glow on the pavement, but there are no cars or people in sight. The silence is broken only by the distant hum of the city and the occasional rustling of leaves in the gentle breeze. It's as if the world is holding its breath, waiting for something to happen.

Table 4: Case study on responses to the original prompts and Bi-GPT-generated prompts. The red highlights represent the similar parts between the ground truth answer and the answer generated by Bi-GPT using the self-produced prompt.

Meta Category	Examples
Background supplementary	Background: The frequency of sound waves is measured in hertz (Hz), or the number of waves that pass a fixed point in a second. Human beings can normally hear sounds with a frequency between about 20 Hz and 20,000 Hz. Sounds with frequencies below 20 hertz are called infrasound Sounds. with frequencies above 20,000 hertz are called ultrasound Some. other animals can hear sounds in the ultrasound range. For example, dogs can hear sounds with frequencies as high as 50,000 Hz. You may have seen special whistles that dogs but not people can hear. The whistles produce a sound with a frequency too high for the human ear to detect. Other animals can hear even higher-frequency sounds. Bats, for example, can hear sounds with frequencies higher than 100,000 Hz. Paragraph: David wants to categorize animals according to the sound waves they can hear. Eventually, he was able to categorize them in three different groups, group A, group B, and group C. He placed the animals that can only hear ifrasound in group A. Then he placed the animals that are only able to hear the same frequency of sound waves as humans do in group B. The last group of animals were able to hear only ultrasound. He placed them in group C. Given the paragraph above, please answer correctly the following question: Which group would be able to hear the highest frequency, group A, group B, or group C? The paragraph describes three groups of animals group A, B, and C. Group A cannot hear infrasound; meanwhile the other two groups can detect this frequency range. However, only group C can detect ultrasound because humans are unable to do so. Therefore, we conclude that among all three groups of animals mentioned in the paragraph above, it is group C who can perceive sound waves with higher frequencies than the others (ultrasound). In conclusion: Group C is able to hear sounds that have a high frequency compared to those heard by Groups A and B.
Methodological guidance	Calculate the number of thorns in a bush of roses by multiplying the number of roses by the number of thorns per rose. Dan plants 3 rose bushes. Each rose bush has 25 roses. Each rose has 8 thorns. How many thorns are there total? First find the total number of roses: 3 bushes * 25 roses/bush = «3*25=75»75 roses Then multiply the number of roses by the number of thorns per rose: 75 roses * 8 thorns/rose = «75*8=600»600 thorns #### 600
Story tell	Given a story, solve a math problem. The story is about a young man named Josh who has a dream of becoming rich. He has invested a lot of money in stocks, but Josh wants to do something other than investing in the stock market. Josh decides to try flipping a house. He buys a house for $80,000 and then puts in $50,000 in repairs. This increased the value of the house by 150%. How much profit did he make? The cost of the house and repairs came out to 80,000+50,000=$«80000+50000=130000»130,000 He increased the value of the house by 80,000*1.5=«80000*1.5=120000»120,000 So the new value of the house is 120,000+80,000=$«120000+80000=200000»200,000 So he made a profit of 200,000-130,000=$«200000-130000=70000»70,000 #### 70000
Zero-shot CoT	Provide a step-by-step solution to the following question. Shannon loves her homemade madeleine cookies. Her recipe makes 12 cookies. Shannon makes her own madeleine cookies and eats 2 a night as a treat. She wants to make enough cookies to last her for 30 days by storing them in the freezer. Her recipe makes 1 dozen madeleine cookies. How many dozens of cookies will she need to make so she has enough for 30 days? She eats 2 cookies a night so for 30 nights she needs 2*30 = «2*30=60»60 cookies Her recipe makes 12 cookies and she needs to make 60 cookies so that's 60/12 = «60/12=5»5 dozen #### 5
Roly play	Imagine you are Sherlock Holmes, and I've just arrived at the house of a woman named Lori, who is trying to figure out the number of eggs she needs to prepare for her party. Lori gave me the key to her kitchen and a carton of 2 dozen eggs. I stood next to her while she was preparing for the party and discovered that Lori needed 1 whole egg to make 2 deviled egg halves. She anticipates that each of her guests will eat 3 deviled egg halves. If she is inviting 16 guests to her party, how many dozens of eggs will she need? She is inviting 16 guests that will eat 3 deviled egg halves each so she needs 16*3 =«16*3=48»48 halves 1 whole egg is needed to make 2 halves so 48 halves is 48/2 = «48/2=24»24 whole eggs 1 dozen is equal to 12 and she needs 24 eggs so she needs 24/12 = «24/12=2»2 dozen eggs #### 2
Task description	Make a deduction from the information in the scenario, given the following information about the situation. Can you find out how much was withheld from her wage? Sally has realized she did not receive a full wage this week. Her bank account, which held $200 at the start of the week, now holds $420. She has received no other money into her bank account this week. If her weekly wage should be $300, how many dollars were withheld from Sally's wage? The wage she received was $420 - $200 = $«420-200=220»220 This means her wage was $300 - $220 = $«300-220=80»80 short #### 80

Table 5: Meta prompt generation from one single instance case study. The cyan section represents the original prompt, the blue section denotes the original answer, and the red section shows the meta-prompt generated by Bi-GPT based on the original prompt and answer.

6 Limitations

While the proposed strategy allows for full-parameter pretraining from scratch, it is computationally expensive and impractical due to the vast number of tokens required. Additionally, our experiments focused on large language models ranging from 1B to 8B parameters, limiting the ability to evaluate scaling performance for larger models. Further research is needed to explore the effectiveness and feasibility of scaling this approach to models with more parameters, which may present additional challenges in terms of computational resources and training time.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Justin Chih-Yao Chen, Zifeng Wang, Hamid Palangi, Rujun Han, Sayna Ebrahimi, Long Le, Vincent Perot, Swaroop Mishra, Mohit Bansal, Chen-Yu Lee, et al. 2024. Reverse thinking makes llms stronger reasoners. *arXiv preprint arXiv:2411.19865*.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

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 preprint arXiv:1803.05457*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems, 2021. *URL https://arxiv.org/abs/2110.14168*.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, and et al. Kadian, Abhishek. 2024. The llama 3 herd of models.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

S Hochreiter. 1997. Long short-term memory. *Neural Computation MIT-Press*.

Weisen Jiang, Han Shi, Longhui Yu, Zhengying Liu, Yu Zhang, Zhenguo Li, and James Kwok. 2024. Forward-backward reasoning in large language models for mathematical verification. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 6647–6661.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2022. On the advance of making language models better reasoners. *arXiv preprint arXiv:2206.02336*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Yang Liu, Yuanshun Yao, Jean-Francois Ton, Xiaoying Zhang, Ruocheng Guo Hao Cheng, Yegor Klochkov, Muhammad Faaiz Taufiq, and Hang Li. 2023b. Trustworthy llms: A survey and guideline for evaluating large language models' alignment. *arXiv preprint arXiv:2308.05374*.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. 2022. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. *arXiv preprint arXiv:2209.14610*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2024. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2019. Adversarial nli: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2021. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633*.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681.

Rico Sennrich. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2024. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36.

Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, et al. 2022. Selective annotation makes language models better few-shot learners. *arXiv preprint arXiv:2209.01975*.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. 2019. Crossweigh: Training named entity tagger from imperfect annotations. *arXiv preprint arXiv:1909.01441*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. 2022. Large language models are better reasoners with self-verification. *arXiv preprint arXiv:2212.09561*.

Wojciech Zaremba. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

Shu Zhang, Dequan Zheng, Xinchen Hu, and Ming Yang. 2015. Bidirectional long short-term memory networks for relation classification. In *Proceedings of the 29th Pacific Asia conference on language, information and computation*, pages 73–78.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

A Implementary Details

A.1 Dataset Statistics

Dataset	Domain	License	Train	Validation	Train†	Validation†	Test
ARC(Challenge)(Clark et al., 2018)	Commonsense	CC BY-SA 4.0	1119	299	-	-	1172
MATH(Hendrycks et al., 2021)	Math	MIT	7500	-	7300	200	5000
GSM8K(Cobbe et al., 2021)	Math	MIT	7473	-	7273	200	1339
TabMWP(Lu et al., 2022)	Tabular	CC BY-SA 4.0	23,059	7686	-	-	7686
ANLI(r3)(Nie et al., 2019)	NLI	CC BY-NC 4.0	100,459	1200	-	-	1200
Alpaca-GPT4(Peng et al., 2023)	QA	CC BY-NC 4.0	52,002	-	50,002	1000	1000
CoNLL2003(Sang and De Meulder, 2003)	NER	-	14987	3466	-	-	3684
CoNLL++(Wang et al., 2019)	NER	Apache 2.0	14987	3466	-	-	3684

Table 6: The finetuning datasets used in this work, including the forward generation task, the prompt optimization task and the token classification task. For datasets without a validation set, we manually split the training set to form the column Train† and column Validation†.

A.2 Pretraining Details

As illustrated in Fig. 3, the upper panel presents the perplexity (PPL) curves for forward and reverse text during full-parameter pre-training using only reverse data. It is observed that as training progresses, the PPL for forward text increases, while the PPL for reverse text decreases rapidly. After the two curves intersect, they gradually tend to converge.

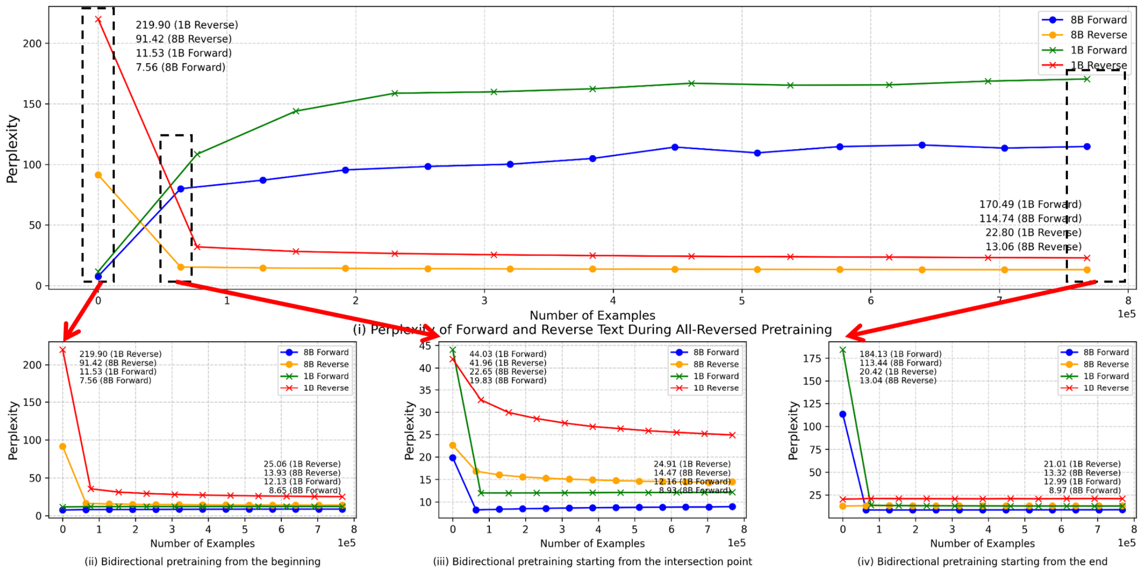

Figure 3: Perplexity of forward and reverse text during two-step pretraining. (i) shows the PPL on forward and reverse text for the Llama3 model (1B and 8B) during reverse-only pretraining. (ii)–(iv) present PPL during bidirectional pretraining from different starting points.