Unveiling the Potential of BERT-family: A New Recipe for Building Scalable, General and Competitive Large Language Models

Yisheng Xiao[♡], **Juntao Li**[♡*], **Wenpeng Hu**[◇], **ZhunChen Luo**[◇], **Min Zhang**[♡],
[♡]Soochow University, SuZhou, China
[◇]Information Research Center of Military Science, PLA Academy of Military Science
ysxiaoo@stu.suda.edu.cn; ljt@suda.edu.cn
wenpeng.hu@pku.edu.cn; zhunchenluo@gmail.com; minzhang@suda.edu.cn

Abstract

BERT-family have been increasingly explored for adaptation to scenarios beyond language understanding tasks, with more recent efforts focused on enabling them to become good instruction followers. These explorations have endowed BERT-family with new roles and human expectations, showcasing their potential on par with current state-of-the-art (SOTA) large language models (LLMs). However, several certain shortcomings in previous BERT-family, such as the relatively sub-optimal training corpora, learning procedure, and model architecture, all impede the further advancement of these models for serving as general and competitive LLMs. Therefore, we aim to address these deficiencies in this paper. Our study not only introduces a more suitable pre-training task that helps BERT-family excel in wider applications to realize generality but also explores the integration of cutting-edge technologies into our model to further enhance their capabilities. Our final models, termed **Bi**directional **G**eneral **L**anguage **M**odels (**BiGLM**), exhibit performance levels comparable to current SOTA LLMs across a spectrum of tasks. Moreover, we conduct detailed analyses to study the effects of scaling and training corpora for BiGLM. To the best of our knowledge, our work represents the early attempt to offer a recipe for building novel types of scalable, general, and competitive LLMs that diverge from current autoregressive modeling methodology. Our codes and models are available on Github[1].

1 Introduction

Generative large language models (LLMs) have significantly influenced various aspects of society, reshaping how we access and interact with information and knowledge (Touvron et al., 2023a,b; Team et al., 2023; OpenAI, 2023). Among them, almost all the recent models adopt the decoder-only model architecture with the autoregressive (AR) modeling paradigm, with the representative being the GPT series models (Radford et al., 2019; Brown et al., 2020; OpenAI, 2023). While this recipe has demonstrated effectiveness in achieving scalability and generality in current LLMs (Tay et al., 2022; Biderman et al., 2023; Touvron et al., 2023a), it also exposes several challenges, such as the well-known teacher forcing problem (Zhang et al., 2019), generation hallucinations (Ji et al., 2023; Rawte et al., 2023; Zhang et al., 2023; Tonmoy et al., 2024), and reduced efficiency during inference (Xiao et al., 2022; Xia et al., 2024; Zhang et al., 2024a). These challenges serve as a catalyst for us to attempt to find, at least discuss the potential of alternative approaches for developing scalable, general, and competitive large language models.

Hence, we investigate the potential of BERT-family, which adopt the encoder-only model architecture with the masked language modeling (MLM) paradigm. Our explorations are driven by several key observations: (1) BERT-family have been one of the most widely used language models in previous years (Devlin et al., 2018; Liu et al., 2019), which contain variants boasting billions of model parameters (Conneau and Lample, 2019; Shoeybi et al., 2019), showcasing its scalability potential. (2) The bi-directional attention mechanism inherent in BERT-family, equips these models with a profound understanding of semantic information, earning them a reputation for excelling in various language understanding tasks. (3) With theoretically indicating that BERT-family can generate coherent textual content (Dong et al., 2019; Wang and Cho, 2019), researchers have leveraged these models in non-autoregressive generation tasks and yield positive feedback (Chan and Fan, 2019; Jiang et al., 2021; Su et al., 2021; Liang et al., 2023b,a). Recently, Xiao et al. further demonstrate that BERT-family can also become instruction followers with instruction tuning. These explorations all indicate

[*] Juntao Li is the Corresponding Author.

[1] github.com/LitterBrother-Xiao/BiGLM

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 29818–29833
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

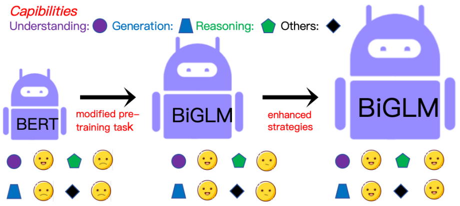


Figure 1: The presentation of the evolution of BiGLM.

the potential of generality for BERT-family.

Despite these positive attempts of BERT-family, we also notice the following shortcomings among them, such as the mismatching of pre-training paradigm for generation tasks and several sub-optimal designs of pre-train models including the model architecture, training procedure and data compositions compared to the up-to-date LLMs. Therefore, we aim to address these deficiencies and make the following contributions to build a novel type of scalable, general, and competitive LLMs:

- We introduce a feasible pre-training task to train new variants of BERT-family termed as **Bi**directional **G**eneral **L**anguage **M**odels (**BiGLM**), which provide a recipe for building LLMs beyond autoregressive modeling.

- We explore the potential of integrating the cutting-edge technologies whose effectiveness has been verified in current AR models into BiGLM to further enhance its capabilities.

- We evaluate BiGLM on a range of scenarios, including task-specific fine-tuning, zero-shot reasoning, and multitask learning. Results demonstrate that BiGLM can reach the performance levels that are on par with, and in some cases surpassing the previous SOTA models.

- We further conduct detailed analyses to study the effects of scaling and training corpora for our models, providing better understandings of BiGLM for current LLM community.

2 Bidirectional General Language Models

We draw lesson from the traditional masked language modeling (MLM) pre-training objective, which makes the model to learn to predict the specific masked tokens and has been widely used in BERT-like models (Devlin et al., 2018; Liu et al., 2019). Specifically, MLM first replaces partial tokens with the special masked token (e.g., [MASK]) in the training instance, and enables the model to predict the corresponding masked parts as follows:

$$\mathcal{L}_{\text{MLM}} = -\sum_{c_t \in C_{mask}} \log \mathcal{P}(c_t | C_{obs}; \theta), \quad (1)$$

where C_{mask} and C_{obs} denote the masked and unmasked parts in the training instance C, respectively. c_t denotes each masked token, and θ denotes the trainable parameters of the model. In conventional BERT-like models (Devlin et al., 2018), the masked tokens C_{mask} are typically randomly selected with a fixed small proportion (e.g., 0.15) of tokens within each training instance. While this pre-training task facilitates the learning of sentence-level representations, it falls short in capturing language generation capabilities compared to the traditional pre-trained AR language models trained with the widely-used causal language modeling objective (i.e., next token prediction task).

BiGLM aims to build a general pre-trained language model which simultaneously possesses the ability of language understanding and generation. Firstly, motivated by the previous works which adapt the traditional MLM to generation tasks (Ghazvininejad et al., 2019; Liang et al., 2023b,a; Xiao et al., 2024), we first decompose each training instance into two parts to simulate a scenario akin to conditional generation. Then, drawing inspiration from prior practice (Song et al., 2019; Li et al., 2022; Guo et al., 2020; Xiao et al., 2023), we further assign different masking strategies for these two parts to enable BiGLM learn the understanding and generation capabilities, respectively. Besides, we adopt specific attention masking mechanism to enhance the consistency between the training and inference process for BiGLM.

2.1 Pre-train Task

Specifically, as shown in Figure 2, given a specific training instance with the max context length L: $C = \{c_1, c_2, ..., c_{L-1}, c_L\}$, BiGLM decomposes C into a tuple (X, Y) based on a decomposition position $i, i \in (1, L)$, where $X = \{c_1, c_2, ...c_{i-1}, c_i\}$ denotes the prefix tokens, and $Y = \{c_{i+1}, c_{i+2}, ...c_{L-1}, c_L\}$ denotes the suffix tokens. This decomposition position controls the minimum length of the X and Y. In practice, we set a ratio $\alpha, \alpha \in (0, 0.5)$ in advance, and randomly sample the position i from $\alpha * L$ to $(1 - \alpha) * L$. Then, the prefix tokens are used to provide context information and help the model

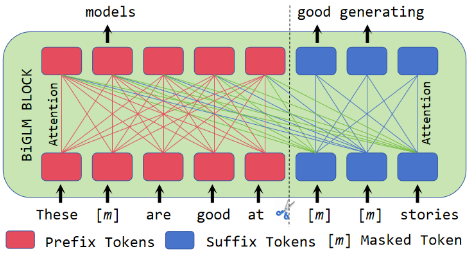


Figure 2: The pre-training task of BiGLM, where each specific training instance is decomposed into the prefix and suffix tokens. We assign random masking strategy with relatively small ratio for prefix tokens to learn understanding ability and uniform masking for suffix tokens to learn generation ability for BiGLM.

understand the whole sentence, we randomly sample a small ratio of mask tokens, which is similar to the traditional MLM in BERT, denoted as $(X_{mask}, X_{obs}) = \texttt{RANDOM\_MASK}(X, \beta_X)$, where X_{mask} and X_{obs} denote the masked and unmasked parts in X, β_X denotes the masking ratio. The suffix tokens tend to help the model learn the generation capability, we adopt uniform masking as mentioned in CMLM (Ghazvininejad et al., 2019), denoted as $(Y_{mask}, Y_{obs}) = \texttt{UNIFORM\_MASK}(Y, \beta_Y)$, where β_Y is sampled from a uniform distribution $U(0, 1)$. Then, BiGLM learns to predict the masked tokens based on different contexts. In practice, we adopt an adaptive masking function for the masking ratio β_X as mentioned in Xiao et al. (2023) to replace the fixed masking ratio in the traditional MLM for X, as $\beta_X = \lambda_1 - \lambda_2 * \beta_Y$, where λ_1 and λ_2 determines the masking ratio range of X. This operation can achieve more diverse masking conditions in X for BiGLM to learn and is based on the intuition that once more tokens in Y are masked, X should provide more context information (i.e., lower β_X). Moreover, we prevent the query of each token in X attending the tokens in Y in the attention module as shown in Figure 2 during training to keep consistent with the inference process since there is no target sequence in advance. Finally, the training loss of BiGLM can be computed as:

$$\begin{aligned}\mathcal{L}_{\text{BiGLM}} = &- \sum_{x_t \in X_{mask}} \log \mathcal{P}(x_t | X_{obs}; \theta) \\ &- \sum_{y_t \in Y_{mask}} \log \mathcal{P}(y_t | X_{obs}, Y_{obs}; \theta).\end{aligned} \tag{2}$$

2.2 Trails for BiGLM

In this section, we pre-train different model variants from scratch to conduct evaluation experiments for BiGLM[2]. Specifically, we first verify the necessity of two key components of our modified pre-training task, i.e., the decomposition of the training instance and the specific attention masking strategy. Then, we further conduct ablation studies to compare different methods to determine the decomposition points and various masking ratios for the prefix tokens in the training sequence.

Data and Architecture For the pre-training corpora, we adopt a deduplicated version of FineWeb-edu (Lozhkov et al., 2024) developed by SmolLM-Corpus (Ben Allal et al., 2024) which contains around 220B tokens, denoted as deduplicated FineWeb-edu. As for the model architecture, we follow the most practice in previous BERT-family to build an encoder-only language model with bi-directional attention mechanism, and further incorporate several modifications to align with current language models (Touvron et al., 2023a; Biderman et al., 2023): 1) We use Rotary Positional Embedding (RoPE) (Su et al., 2024) to replace the traditional absolute/relative position encoding to inject positional information. 2) We replace the traditional ReLU with swiglu (Shazeer, 2020) as our activation function 3) We adopt RMSNorm (Zhang and Sennrich, 2019) as our normalization method rather than the common layer normalization. We adopt a model version containing around 124M parameters whose num-layers/hidden-size/num-attn-heads are 12/768/12 to conduct experiments.

Training Details We pre-train all the model variants from scratch with a max length of 2048, batch size of 1024, and the training steps as 50k, i.e., totally with around 100B tokens. The learning rate is set as 3e-3 and decreases with `cosine` decay strategy. We utilize the Megatron-Deepspeed [3] library, and train all the models on 64 NVIDIA A100-PCIE-80GB GPU cards. As for the specific variants, we train the common BiGLM and then successively omit the attention masking strategy (i.e., w/o *attn*) and the decomposition process (i.e., w/o *both*) to obtain three variants. For the ablation studies, we compare the different decomposition ratios and different masking factors for X and Y. The details are presented in Appendix C.

Evaluation Details After the training process, we evaluate the models without fine-tuning on a

[2] In this paper, all the evaluation experiments are only conducted on English language data.

[3] https://github.com/microsoft/Megatron-DeepSpeed

Methods	ARC-E	ARC-C	PIQA	Sciq	Wino.	LogiQA	Race	SIQA	BoolQ	Hella.	Truth.	AVG.
RoBERTa-base	36.07	25.68	58.98	61.8	51.78	26.27	27.94	35.62	61.19	33.97	25.12	40.40
BiGLM	52.95	26.37	60.55	85.1	49.80	28.17	28.04	38.16	60.64	34.56	24.96	**44.48**
w/o *attn.*	51.09	23.89	59.90	83.8	52.56	29.03	27.37	38.08	61.53	32.80	25.95	44.18
w/o *both.*	41.58	22.69	56.58	76.6	49.96	27.80	28.80	38.11	60.40	31.23	24.84	41.69

Table 1: Results of various pre-training variants. **Wino.**, **Hella.**, and **Truth.** denote the WinoGrande, Hellaswag, and Truthfulqa datasets, **AVG.** denotes average result. *attn.* denotes the attention masking strategy.

range of widely-used zero-shot reasoning tasks, including ARC-easy, ARC-challenge (Clark et al., 2018), BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019), WinoGrande (Sakaguchi et al., 2021), LogiQA (Liu et al., 2020), Race (Lai et al., 2017), Sciq (Johannes Welbl, 2017), Hellaswag (Zellers et al., 2019), and Truthfulqa (Lin et al., 2021). We adopt Language Model Evaluation (Gao et al., 2021) framework to evaluate these datasets under a zero-shot setting (Biderman et al., 2023) and report normalized accuracy for PIQA, ARC-challenge, LogiQA, Hellaswag, and accuracy for other tasks (Biderman et al., 2023).

Results The results on zero-shot reasoning tasks are presented in Table 1, we can find that (1) the corresponding two key components are necessary for our pre-training task. The decomposition of the training instance and assigning different masking are more critical for the success of BiGLM, while removing it leads to significant performance declines; (2) we also report the performance of previous competitive BERT-like model (i.e., RoBERTa-base (Liu et al., 2019)) with comparable parameters but trained with much more tokens (around 2T tokens), our model, even trained with only 100B tokens based on the same pre-training task (i.e., w/o *both*), can also achieve better performance, indicating the effectiveness of modifying the model architecture and pre-training data corpus. As for the ablation studies presented in Appendix C, we can find that all the variants perform comparably.

3 Enhanced Strategies for BiGLM

In this section, we explore the feasibility of integrating several effective cutting-edge technologies into BiGLM to further enhance the capabilities.

3.1 Model Architecture

Deeper Model Additional to the common modifications as mentioned in 2.2, recent work has proposed that while training a language model, going deeper is more crucial than going wider for performance improvement (Liu et al., 2024). In other words, after determining the total model parameters, we prefer adding the number of layers rather than wider the hidden-size. As a result, we follow the model designs in (Liu et al., 2024) to train a deeper BiGLM but with comparable parameter. Specifically, we set the num-layers/hidden-size/num-attn-heads as 30/576/9 to replace the original 12/768/12. Furthermore, we also adopt the method of grouped query attention (Chowdhery et al., 2023; Ainslie et al., 2023) which reduces the original parameters to allow more layers. The corresponding results are presented in Table 2, we can find the deeper model (BiGLM++) outperforms the original BiGLM by around 0.5 score on average. However, we need to recognize that the training time of the deeper model is around 2x than the common BiGLM in our experiments.

Dropout Module We evaluate the necessity of Dropout (Srivastava et al., 2014), which serves as a simple way to avoid over-fitting but been omitted in recent LLMs (Touvron et al., 2023a,b). We include this exploration based on that all previous BERT-family, even with billion parameters, still adopt the dropout module (Conneau and Lample, 2019; Shoeybi et al., 2019). The corresponding results are presented in Table 2, i.e., BiGLM v.s., BiGLM w/o *dropout*. We can find that omitting the dropout module leads to around 1 score improvement on average, indicating that the dropout module is also not necessary for BiGLM.

3.2 Training Procedure

Learning Rate Scheduler While researchers adopt Cosine Learning Rate Scheduler (Cosine LRS) to train most LLMs, Hu et al. have seek for better one, i.e., the Warmup-Stable-Decay Learning Rate Scheduler (WSD LRS), which divides the training process into three stages: 1) the warm-up stage as the same as previous practice, 2) the stable training stage with the learning rate unchanged, 3) the annealing stage with the learning rate decreas-

Methods	ARC-E	ARC-C	PIQA	Sciq	Wino.	LogiQA	Race	SIQA	BoolQ	Hella.	Truth.	AVG.
BiGLM	52.95	26.37	60.55	85.1	49.80	28.17	28.04	38.16	60.64	34.56	24.96	44.48
w/o *dropout*	53.76	26.40	60.78	87.2	51.73	28.33	30.12	38.49	62.03	35.01	25.17	45.37
BiGLM++	53.70	25.34	60.55	85.8	51.86	28.12	28.80	38.54	62.05	34.74	24.99	44.95
w/ *mixdata*	55.13	25.51	61.97	88.5	51.85	28.36	30.43	37.93	62.02	35.67	25.31	45.70
no annealing	51.34	24,74	60.56	84.4	54.06	27.54	30.05	37.32	61.99	34.30	25.21	44.68
rawdata annealing	52.95	26.28	61.91	86.3	52.09	28.38	30.33	38.98	61.74	34.93	24.73	45.33
rawdata annealing++	53.21	25.97	62.32	86.3	52.17	28.28	30.17	38.96	62.01	35.25	25.02	45.42
syndata annealing	50.04	26.02	62.68	79.4	51.70	28.02	29.47	37.78	61.53	35.13	25.46	44.29
mixdata annealing	52.44	25.68	61.37	85.4	50.51	28.13	29.19	38.15	61.74	35.31	24.75	44.79

Table 2: Results of adopting the enhanced strategies in BiGLM.

ing linearly. This scheduler provides a simpler way for continue training and has been adopted in recent competitive models, e.g., Llama 3.1 (Vavekanand and Sam, 2024) and Falcon-Mamba (Zuo et al., 2024). In Hu et al. (2024) where WSD LRS is first proposed, 10% of total training steps are adopted for annealing, i.e., final 5k steps for annealing for BiGLM since the total training steps is 50k. During the annealing stage, we adopt the same training data distribution (i.e., deduplicated FineWeb-edu) as that in the stable training stage. Besides, considering the relatively lower learning efficiency of BERT-family (Wettig et al., 2022), we trail for a longer annealing stage (i.e., *rawdata annealing++* in Table 2) with final 10k steps for annealing after 40k training steps. We present the corresponding results in Table 2, demonstrating that (1) we report a baseline that does not adopt the annealing stage (i.e., *no annealing*), i.e., a total of 50k steps for the warm-up and stable training stage, which results in a 0.27 score decline on average compared with the one trained with Cosine LRS (i.e., BiGLM++); (2) WSD LRS (i.e., *rawdata annealing*) outperforms Cosine LRS, and longer annealing stage leads to better performance, indicating that BiGLM needs more training steps during the annealing stage.

3.3 Data Composition

Pre-training Data Previous non-autoregressive works (Ghazvininejad et al., 2019; Kasai et al., 2020; Huang et al., 2022; Xiao et al., 2023) which also adopts the MLM objective for training have demonstrated that data distillation is quite important for competitive performance. They train their models with the data generated by the autoregressive models rather than the raw data, which can simplify the modalities in training data and reduce the modeling difficulties. This also alleviates the well-known multi-modality problem (Gu et al., 2018) which affects the performance of BERT-family for generation tasks (Liang et al., 2023a,b). However, the data composition is not well explored for previous BERT-family. Thus, we adopt the Cosmopedia v2 (Ben Allal et al., 2024), which is a collection of synthetic textbooks and stories generated by mistralai/Mixtral-8x7B-Instruct-v0.1[4] (Jiang et al., 2024) and contains around 39B tokens, to serve as the distillation data to verify the effectiveness of synthetic data. Specially, we compared the models trained on only the deduplicated FineWeb-edu and the mixture of deduplicated FineWeb-edu and Cosmopedia v2 for the same total tokens, as shown in Table 2, i.e., BiGLM++ v.s., w/ *mixdata*, training on the mixture data leads to significant performance improvements by 0.75 scores on average.

Annealing Data Previous works (Hu et al., 2024; Vavekanand and Sam, 2024) have pointed out the annealing stage always needs higher-quality training data, e.g., selective code and math data or exquisite synthetic data, to enable the better convergence of the model. Thus, we explore the effects on different data compositions during the annealing stage. Rather than adopting the same distribution as mentioned in Section 3.2, we include two variants which adopt only the synthetic data and the mixture data during the annealing stage, with results presented in Table 2 and termed as *syndata annealing* and *mixdata annealing*. Contrary to the common intuition, while including the higher-quality synthetic data, adopting *syndata annealing* and *mixdata annealing* both leads to performance declines, especially with *syndata annealing*, we attribute this to the mismatching of the data distribution between the stable training and the annealing stage.

4 Experiments

Based on the above observations in Section 3.2, we adopt a mixture of deduplicated FineWeb-edu and Cosmopedia v2 to pre-train three versions of BiGLM with different parameters, termed as

[4] https://huggingface.co/mistralai

Methods	MNLI m/mm	SQuAD	XSUM	MSQG
	Accuracy	EM / F1	R-1 / R-2 / R-L	R-L / B-4 / MR
BERT-Base	84.3 / -	80.5 / 88.5	39.1 / 15.3 / 31.0	38.3 / 9.5 / 22.0
Roberta-Base	84.6 / -	83.0 / 90.4	41.5 / **17.5** / 33.5	38.5 / 10.5 / 22.7
BART-Base	84.1 / -	- / **90.8**	38.8 / 16.2 / 30.6	38.2 / 10.2 / 22.9
BiGLM-136M	**84.7** / -	**83.1** / 90.6	**41.6** / 17.3 / **33.6**	**39.2** / **10.6** / **23.6**
BERT-Large	86.7 / 85.9	88.0 / 93.7	39.8 / 15.8 / 31.9	38.9 / 10.2 / 22.9
Roberta-Large	90.2 / 90.2	**88.9** / **94.6**	44.5 / 20.4 / 36.3	40.1 / **11.2** / 23.6
DeBERTaV3-Large	**91.8** / **91.9**	- / -	- / - / -	- / - / -
BART-Large	89.9 / 90.1	88.8 / 94.6	**45.1** / **22.2** / **37.2**	38.8 / 9.2 / 24.3
BiGLM-360M	90.2 / 90.3	**88.9** / 94.5	44.6 / 21.0 / 36.4	**40.2** / **11.2** / **24.4**
DeBERTa-1.5B	**91.7** / **91.9**	- / -	- / - / -	- / - / -
Megatron-1.3B	90.9 / 91.4	89.1 / 94.9	- / - / -	- / - / -
Megatron-3.5B	91.4 / 91.4	90.0 / 95.5	- / - / -	- / - / -
BiGLM-1.3B	91.2 / 91.3	89.8 / 95.2	46.2 / 22.7 / 38.0	40.8 / 11.5 / 24.9
BiGLM-3.5B	**91.7** / **91.9**	**90.1** / **95.5**	**47.1** / **23.3** / **38.7**	**41.3** / **11.8** / **25.3**

Table 3: Result of task-specific scenarios. The evaluation metrics are simplified: EM / F1 : exact match score / F1 score, R-1 / R-2 / R-L : ROUGE-1 / ROUGE-2 / ROUGE-L, R-L / B-4 / MR : ROUGE-L / BLEU-4 / METEOR.

BiGLM-136M, BiGLM-360M, and BiGLM-1.3B. Besides, we include the above-mentioned common model modifications and omit the dropout module. Then, we we follow the deeper model architectures in (Liu et al., 2024) to train BiGLM-136M and BiGLM-360M, and follow the common design to train BiGLM-1.5B and BiGLM-3.5B considering the training efficiency. We set the max length as 2048 and batch size as 1024, then train BiGLM for a total of 300k steps (around 600B tokens). Additionally, we adopt the WSD LRS to train all the models with 20% time for the annealing stage. More details are presented in Appendix A.

4.1 Task-specific Fine-tuning

Datasets and Models We evaluate BiGLM for task-specific fine-tuning scenarios with the following dataset, i.e., MNLI (Williams et al., 2017) and SQuAD (Rajpurkar et al., 2016) for understanding tasks, XSUM (Narayan et al., 2018) and MSQG (MicroSoft Question Generation) for generation tasks. The details of these datasets are presented in Appendix B. For the baseline models, we adopt two representative BERT-family models (BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019) and DeBERTaV3 (He et al., 2020)). Besides, we also include BART (Lewis et al., 2019), which can perform well in both language understanding and generation tasks. We further adopt the model versions of BERT containing 1.3B and 3.5B parameters which are provided in Shoeybi et al. (2019) to compare with BiGLM-1.3B and BiGLM-3.5B.

Settings We fine-tune BiGLM on XSUM and MSQG following the previous practice (Liang et al., 2023a; Xiao et al., 2024), which utilizes the Mask-Predict decoding algorithm (Ghazvininejad et al., 2019) to adapt the BERT-family to language generation scenarios. Besides, we follow the practice in the traditional BERT-family for SQuAD, but for MNLI, rather than adopting the representation of the [cls] token to predict the label class in the traditional BERT-family, we enable BiGLM to predict the real label with a specific prompt (Bach et al., 2022). During fine-tuning, we train for a total of 5 epochs for MNLi and SQuAD, and 50 epochs for XSUM and MSQG. We validate BiGLM after each epoch and select the best one as our final model. As for the evaluation metrics, we follow Liu et al. (2021) to adopt ROUGE-1/2/L (Lin and Hovy, 2002) for XSUM, BLEU-4 (Papineni et al., 2002), Rouge-L and METEOR (Lavie and Agarwal, 2007) for MSQG. Besides, we report accuracy for MNLI, exact match, and F1 score for SQuAD following previous BERT-family (Liu et al., 2019).

Results The corresponding results are presented in Table 3, we can find that for these models under 1B parameters: (1) BiGLM can outperform the most baselines on MNLI and MSQG. (2) BiGLM achieve comparable and in some cases superior performance on SQuAD and XSUM. (3) BiGLM-360M achieves relatively inferior performance on XSUM compared to BART-Large. We attribute this to the non-autoregressive generation paradigm which falls short in generating longer targets. Be-

Models	LogiQA	Sciq	ARC-E	ARC-C	Wino.	BoolQ	PIQA	SIQA	Race	Hella.	Truth.	AVG.
RWKV-169M (300B)	24.73	75.2	47.52	23.46	50.67	62.17	64.04	37.00	26.89	32.25	22.25	42.41
SmolLM-135M (600B)	27.04	83.5	61.61	28.75	53.28	62.17	68.08	39.66	31.77	42.61	25.21	**47.61**
BiGLM-136M (600B)	28.88	86.2	59.43	27.65	53.32	62.23	65.83	38.18	32.93	40.51	25.67	**47.35**
RWKV-430M (300B)	24.42	79.0	52.23	25.17	52.80	62.05	68.44	38.84	28.71	40.78	22.28	44.98
Qwen2-500M (7T)	29.34	90.8	54.55	28.84	57.46	58.84	69.91	42.78	33.97	49.08	24.48	49.10
SmolLM-360M (600B)	28.57	90.7	70.20	36.18	56.99	61.25	71.04	41.15	34.74	53.51	24.60	**51.76**
BiGLM-360M (600B)	29.29	91.8	67.95	34.25	54.72	62.17	67.78	41.33	37.13	52.97	25.80	**51.38**
RWKV-1.5B (300B)	27.80	84.9	60.82	29.01	55.33	52.95	72.36	41.20	32.54	52.95	21.79	48.33
TinyLlama-1.1B (3T)	25.81	89.3	61.66	32.68	59.43	61.56	73.56	42.27	36.94	46.70	22.28	53.17
Qwen2-1.5B (7T)	31.03	94.5	66.37	36.95	65.82	68.93	75.08	45.91	36.36	65.34	30.35	**56.05**
Gemma-2B (3T)	30.26	94.3	74.41	41.55	65.35	65.35	78.29	48.06	36.08	71.43	22.15	**57.02**
SmolLM-1.7B (1T)	28.57	93.2	76.47	46.25	60.93	62.57	76.01	43.20	36.84	65.74	24.26	55.83
BiGLM-1.3B (600B)	30.67	94.7	74.12	42.12	58.27	63.25	74.02	43.65	38.86	63.25	25.83	55.43
RWKV-3B (300B)	28.11	86.0	64.81	33.28	59.98	62.08	74.32	41.15	33.78	59.97	19.83	51.21
Sheared-LLaMA-3B (2T)	28.26	91.1	67.30	33.58	65.04	60.76	76.93	42.07	38.09	68.99	23.99	54.19
Qwen2.5-3B (7T)	33.49	95.4	77.31	47.44	68.43	74.95	78.51	49.95	38.37	72.54	32.07	**60.77**
Open-LLaMA-3B (1T)	28.57	92.2	69.28	36.35	61.80	62.91	74.97	42.22	37.32	64.31	22.40	53.85
BiGLM-3.5B (600B)	32.02	96.1	78.78	47.21	64.97	66.12	77.12	45.87	40.09	72.08	26.17	58.79

Table 4: Results of zero-shot reasoning scenarios.

sides, for those over 1B parameters, BiGLM-1.3B outperforms Megatron-1.3B and there only exists a tiny gap compared to Megatron-3.5B. BiGLM-3.5B outperforms all the baseline models.

4.2 Zero-shot Reasoning

Datasets and Models We adopt a range of zero-shot common sense reasoning and reading comprehension tasks as mentioned in Section 2.2. For baseline models, we adopt the previous LLMs containing the comparable parameters with BiGLM, including RWKV (Peng et al., 2023), SmolLM (Allal et al., 2024), Gemma (Team et al., 2023), several Llama and Qwen variants (Zhang et al., 2024b; Xia et al., 2023; Geng and Liu, 2023; qwe, 2024).

Settings The evaluation for BiGLM is the same as mentioned in Section 2.2. For these baseline models, we also adopt Language Model Evaluation framework to re-run their public released models in Huggingface[5] to obtain their final performance.

Results The corresponding results are presented in Table 4, demonstrating that considering the average performance compared to baselines: (1) for these models with less than 1B parameters, BiGLM outperform most previous LLMs and achieve comparable performance with the current state-of-the-art lightweight SmolLM; (2) while BiGLM-1.3B is trained for 600B tokens, it only slightly underperforms SmolLM-1.7B and Qwen2-1.5B which are trained for 1T and 7T tokens, respectively. Besides, considering specific single dataset, BiGLM can perform best on several datasets, Sciq, BoolQ, and Truthfulqa for BiGLM-135M and BiGLM-360M, Sciq and BoolQ for BiGLM-360M. (3) The performance of BiGLM-3.5B is similr to BiGLM-1.3B, which only underperforms Qwen2.5-3B.

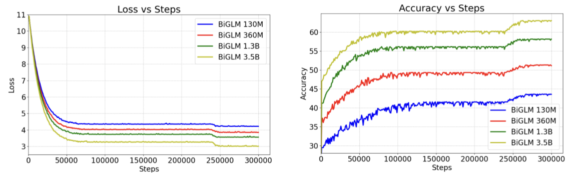
Methods	MMLU ZS / FS	SuperGLUE AVG ACC.	Genset B-2 / D-2
Flan-T5-Small (87M)	30.05 / 29.76	50.58	29.17 / 0.55
Flan-T5-Base (250M)	33.44 / 34.28	64.97	32.46 / 0.62
Flan-T5-Large (780M)	41.54 / 42.03	74.04	38.28 / 0.63
Flan-T5-XL (3B)	48.68 / 49.24	76.56	36.53 / 0.63
Instruct-XMLR (3B)	41.36 / 40.17	74.16	35.83 / 0.70
BiGLM-136M	31.14 / 32.98	53.21	31.23 / 0.72
BiGLM-360M	39.61 / 40.03	68.45	35.29 / 0.71
BiGLM-1.3B	46.17 / 46.59	75.53	40.18 / 0.71
BiGLM-3.5B	51.05 / 52.18	77.12	43.19 / 0.73

Table 5: Result of multitask learning scenarios. The metrics are simplified: ZS / FS: accuracy under zero-shot/few-shot settings, AVG ACC: average score on SuperGLUE, B-2 / D-2: BLEU-2 / Distinct-2.

4.3 Multitask Learning

Datasets and Models We evaluate BiGLM for multitask learning scenario after multitask instruction tuning (Chung et al., 2022; Taori et al., 2023), which ability of BERT-family has also been mentioned in (Xiao et al., 2024). We utilize FLAN dataset (Wei et al., 2021) which is composed of numerous tasks with the instruction format, to fine-tune BiGLM, then we adopt a held-in benchmark (SuperGLUE (Wang et al., 2022)), a held-out one (MMLU (Hendrycks et al., 2021)), and a subset containing several instances sampled from held-out generation tasks including WIKI-AUTO (Jiang et al., 2020) for text simplification, Quora Question Pairs (QQP) for paraphrase generation, and PersonaChat (Zhang et al., 2018) for dialogue generation. For baselines, we adopt Flan-T5 (Wei

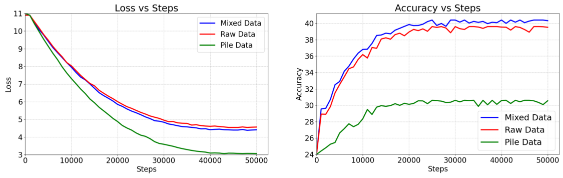
[5] https://huggingface.co



(a) Train Loss

(b) Accuracy of ARC

Figure 3: Results of scaling effects for BiGLM.



(a) Train Loss

(b) Accuracy of ARC

Figure 4: Results of models trained with different data.

et al., 2021) and instruct-XMLR (Xiao et al., 2024), whose details are presented in Appendix B.

Settings We fine-tune BiGLM on FLAN dataset for 5 epochs, and adopt a held-in validation set to evaluate the model after each epoch, then we select the best one as our final model. During training, we set the learning rate as 5e-5 and adopt the `linear` decay schedule. For MMLU, we report the corresponding zero-shot and few-shot results following previous practice, and for SuperGLUE, we report the average accuracy. Moreover, for other generation tasks, we randomly sample 100 instances from each dataset to compose a subset, denoted as Genset. We report BLEU (Papineni et al., 2002) and Distinct (Li et al., 2015) to measure the n-gram level precision and the diversity of generated texts.

Results Table 5 presents the corresponding results, we can find that (1) BiGLM-1.3B outperforms Instruct-XMLR in all scenarios, indicating the effectiveness of our various methods for training new BERT-family. (2) Compared with Flan-T5 models which trained with more tokens (1T) during pre-training stage, BiGLM can also reach the performance level with specific model parameters.

5 Analysis

5.1 Scaling Effects for BiGLM

In this section, we study the scaling effects for BiGLM which plays a vital role in the success of LLMs (Hoffmann et al., 2022; Touvron et al., 2023a). Specifically, we study the loss and performance changes across different model versions throughout the training process. For performance, we present the average accuracy score of ARC-easy and ARC-challenge. We present the corresponding curves in Figure 3, we can find that (1) increasing the model parameters can bring significant performance improvements and reduce the training loss. (2) We can also verify the effectiveness of WSD LRS as mentioned in Section 3.2 while witnessing an evident drop in training loss and improvement in performance after 240k training steps in the figure.

5.2 Effects on Pre-training Corpora

As the training corpora has shown great effects on final capabilities of LLMs, we conduct an analytic experiments of the loss and performance changes during training trained with data. Specifically, except adopting the mixed data and raw data as mentioned in Section 3.3, we also include the Pile (Gao et al., 2020; Biderman et al., 2022), which is a curated collection of English language datasets and has been widely used for training language models (Biderman et al., 2023; Peng et al., 2023). We train BiGLM-136M for 100B tokens and the corresponding results are shown in Figure 4, demonstrating that: (1) while lower loss can be achieved with the pile data, it does not lead to better performance, indicating that data distribution is highly related to the training loss. (2) Compared with raw data and mixed data, adopting the mixed data can achieve lower loss and better performance. Overall, we can only conduct consistent comparisons based on the training loss while there is no significant distribution differences between two corpora.

6 Conclusion

In this paper, we explore the potential of BERT-family for building scalable, general and competitive LLMs. By introducing a more feasible pre-training task and further integrate several cutting-edge technologies in BERT-family, our proposed model variants, which is trained from scratch with bidirectional attention mechanism and termed as Bidirectional General Language Models (BiGLM), can reach the performance levels that are on par with, and in some cases surpassing the current SOTA AR models with comparable parameters. Our works represent the early attempts for seeking novel types of LLMs, aiming to promote further development of the BERT family and further provide a new research direction for LLM community.

Limitations

Due to computational limitations, we only scaled BiGLM to 3.5B parameters, which is still considerably smaller than the current mainstream large language models with tens of billions of parameters, such as LLaMA-65B, Qwen-2-72B, and several GPT series models. Besides, the training data (600B) is also relatively not enough for BiGLM-1.5B and BiGLM-3.5B, leaving a problem that whether BiGLM can breaking through standard scaling laws. Additionally, previous works have pointed out that training language models with masked language modeling with bidirectional attention mechanism need more time to train the same tokens compared with current decoding-only LLMs with autoregressive modeling, which may lead to more computational costs.

7 Acknowledge

We want to thank all the anonymous reviewers for their valuable comments. This work was supported by the National Science Foundation of China (NSFC No. 62206194), the Natural Science Foundation of Jiangsu Province, China (Grant No. BK20220488), the Young Elite Scientists Sponsorship Program by CAST (2023QNRC001), and the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

2024. Qwen2 technical report.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. 2023. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *arXiv preprint arXiv:2305.13245*.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Leandro von Werra, and Thomas Wolf. 2024. Smollm - blazingly fast and remarkably powerful.

Stephen H Bach, Victor Sanh, Zheng-Xin Yong, Albert Webson, Colin Raffel, Nihal V Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, et al. 2022. Promptsource: An integrated development environment and repository for natural language prompts. *arXiv preprint arXiv:2202.01279*.

Loubna Ben Allal, Anton Lozhkov, Guilherme Penedo, Thomas Wolf, and Leandro von Werra. 2024. Smollm-corpus.

Stella Biderman, Kieran Bicheno, and Leo Gao. 2022. Datasheet for the pile. *arXiv preprint arXiv:2201.07311*.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

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.

Ying-Hong Chan and Yao-Chung Fan. 2019. A recurrent bert-based model for question generation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 154–162.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

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*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 7059–7069.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *Advances in Neural Information Processing Systems*, 32.

Dan Fu, Simran Arora, Jessica Grogan, Isys Johnson, Evan Sabri Eyuboglu, Armin Thomas, Benjamin Spector, Michael Poli, Atri Rudra, and Christopher Ré. 2024. Monarch mixer: A simple sub-quadratic gemm-based architecture. *Advances in Neural Information Processing Systems*, 36.

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

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, et al. 2021. A framework for few-shot language model evaluation. *Version v0. 0.1. Sept*, page 8.

Xinyang Geng and Hao Liu. 2023. Openllama: An open reproduction of llama.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6112–6121.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*.

Junliang Guo, Linli Xu, and Enhong Chen. 2020. Jointly masked sequence-to-sequence model for non-autoregressive neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 376–385.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. 2024. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*.

Xiao Shi Huang, Felipe Perez, and Maksims Volkovs. 2022. Improving non-autoregressive translation models without distillation. In *International Conference on Learning Representations*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. 2020. Neural crf model for sentence alignment in text simplification. *arXiv preprint arXiv:2005.02324*.

Ting Jiang, Shaohan Huang, Zihan Zhang, Deqing Wang, Fuzhen Zhuang, Furu Wei, Haizhen Huang, Liangjie Zhang, and Qi Zhang. 2021. Improving non-autoregressive generation with mixup training. *arXiv preprint arXiv:2110.11115*.

Matt Gardner Johannes Welbl, Nelson F. Liu. 2017. Crowdsourcing multiple choice science questions.

Jungo Kasai, James Cross, Marjan Ghazvininejad, and Jiatao Gu. 2020. Parallel machine translation with disentangled context transformer. *arXiv preprint arXiv:2001.05136*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794.

Alon Lavie and Abhaya Agarwal. 2007. Meteor: an automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, page 228–231, USA. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Pengfei Li, Liangyou Li, Meng Zhang, Minghao Wu, and Qun Liu. 2022. Universal conditional masked language pre-training for neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6379–6391.

Xiaobo Liang, Juntao Li, Lijun Wu, Ziqiang Cao, and Min Zhang. 2023a. Dynamic and efficient inference for text generation via bert family. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2883–2897.

Xiaobo Liang, Zecheng Tang, Juntao Li, and Min Zhang. 2023b. Open-ended long text generation via masked language modeling. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 223–241.

Chin-Yew Lin and Eduard Hovy. 2002. Manual and automatic evaluation of summaries. In *Proceedings of the ACL-02 Workshop on Automatic Summarization*, pages 45–51.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Dayiheng Liu, Yu Yan, Yeyun Gong, Weizhen Qi, Hang Zhang, Jian Jiao, Weizhu Chen, Jie Fu, Linjun Shou, Ming Gong, et al. 2021. Glge: A new general language generation evaluation benchmark. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 408–420.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. *arXiv preprint arXiv:2007.08124*.

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 preprint arXiv:1907.11692*.

Zechun Liu, Changsheng Zhao, Forrest Iandola, Chen Lai, Yuandong Tian, Igor Fedorov, Yunyang Xiong, Ernie Chang, Yangyang Shi, Raghuraman Krishnamoorthi, et al. 2024. Mobilellm: Optimizing subbillion parameter language models for on-device use cases. *arXiv preprint arXiv:2402.14905*.

Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. 2024. Fineweb-edu.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1797–1807.

OpenAI. 2023. Gpt-4 technical report.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, Kranthi Kiran GV, et al. 2023. Rwkv: Reinventing rnns for the transformer era. *arXiv preprint arXiv:2305.13048*.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training.

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.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2383–2392.

Vipula Rawte, Amit Sheth, and Amitava Das. 2023. A survey of hallucination in large foundation models. *arXiv preprint arXiv:2309.05922*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

David Samuel. 2024. Berts are generative in-context learners. *arXiv preprint arXiv:2406.04823*.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473.

Noam Shazeer. 2020. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, pages 5926–5936. PMLR.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Yixuan Su, Deng Cai, Yan Wang, David Vandyke, Simon Baker, Piji Li, and Nigel Collier. 2021. Non-autoregressive text generation with pre-trained language models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 234–243.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html.*

Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Steven Zheng, et al. 2022. Ul2: Unifying language learning paradigms. In *The Eleventh International Conference on Learning Representations*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

SM Tonmoy, SM Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. 2024. A comprehensive survey of hallucination mitigation techniques in large language models. *arXiv preprint arXiv:2401.01313*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Raja Vavekanand and Kira Sam. 2024. Llama 3.1: An in-depth analysis of the next-generation large language model.

Alex Wang and Kyunghyun Cho. 2019. Bert has a mouth, and it must speak: Bert as a markov random field language model. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 30–36.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. *arXiv preprint arXiv:2204.07705*.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, et al. 2024. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *arXiv preprint arXiv:2412.13663*.

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*.

Alexander Wettig, Tianyu Gao, Zexuan Zhong, and Danqi Chen. 2022. Should you mask 15% in masked language modeling? *arXiv preprint arXiv:2202.08005*.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Heming Xia, Zhe Yang, Qingxiu Dong, Peiyi Wang, Yongqi Li, Tao Ge, Tianyu Liu, Wenjie Li, and Zhifang Sui. 2024. Unlocking efficiency in large language model inference: A comprehensive survey of speculative decoding. *arXiv preprint arXiv:2401.07851*.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. 2023. Sheared llama: Accelerating language model pre-training via structured pruning. *arXiv preprint arXiv:2310.06694*.

Yisheng Xiao, Juntao Li, Zechen Sun, Zechang Li, Qingrong Xia, Xinyu Duan, Zhefeng Wang, and Min Zhang. 2024. Are bert family good instruction followers? a study on their potential and limitations. In *The Twelfth International Conference on Learning Representations*.

Yisheng Xiao, Lijun Wu, Junliang Guo, Juntao Li, Min Zhang, Tao Qin, and Tie-yan Liu. 2022. A survey on non-autoregressive generation for neural machine translation and beyond. *arXiv preprint arXiv:2204.09269*.

Yisheng Xiao, Ruiyang Xu, Lijun Wu, Juntao Li, Tao Qin, Tie-Yan Liu, and Min Zhang. 2023. Amom: adaptive masking over masking for conditional masked language model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13789–13797.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32.

Chen Zhang, Zhuorui Liu, and Dawei Song. 2024a. Beyond the speculative game: A survey of speculative execution in large language models. *arXiv preprint arXiv:2404.14897*.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024b. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistic*, pages 2204–2213.

Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. 2019. Bridging the gap between training and inference for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4334–4343.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023. Siren's song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Jingwei Zuo, Maksim Velikanov, Dhia Eddine Rhaiem, Ilyas Chahed, Younes Belkada, Guillaume Kunsch, and Hakim Hacid. 2024. Falcon mamba: The first competitive attention-free 7b language model.

A Details of Pre-training

We present the details of the pre-training models in Table 7.

B Details of Datasets and Models

We present the details for evaluation datasets here.

MNLI MNLI (Williams et al., 2017) consists of pairs of premise and hypothesis sentences, as well as labels indicating their relationship (i.e., entailment, neutral, and contradiction). It has two test sets, which comes from matching domains (MNLI-m) and mismatching domains (MNLI-mm) of the training set.

SQuAD SQuAD (Rajpurkar et al., 2016) is a reading comprehension dataset consisting of questions posed by crowdsourcing workers on a set of wikipedia articles, where the answer to each question is a paragraph of text from the corresponding article. Reseacher adopt this dataset to evaluate the extractive question answering for language models.

XSUM XSUM (Narayan et al., 2018) dataset contains 204,045/11,332/11,334 online articles and single sentence summary pairs from the British Broadcasting Corporation for training/validation/test.

MSQG MicroSoft Question Generation (MSQG) is a large-scale dataset for question generation tasks proposed in GLGE benchmark (Liu et al., 2021).

ARC AI2 Reasoning Challenge (ARC) () is datasets composed of genuine grade-school level, multiple-choice science questions. This is further divided into a Challenge Set and an Easy Set, where the former contains only questions answered incorrectly by both a retrieval-based algorithm and a word co-occurrence algorithm.

LogiQA LogiQA (Liu et al., 2020) is constructed from the logical comprehension problems from publically available questions of the National Civil Servants Examination of China, which are designed to test the civil servant candidates' critical thinking and problem solving.

Sciq Sciq (Johannes Welbl, 2017) contains 13,679 crowdsourced science exam questions about Physics, Chemistry and Biology. Among them, an additional paragraph with supporting evidence for the correct answer is provided.

WinoGrande WinoGrande (Sakaguchi et al., 2021) is formulated as a fill-in-a-blank task with binary options, aiming to enable the language model to choose the right option for a given sentence.

BoolQ BoolQ (Clark et al., 2019) a question answering dataset with labels as yes/no. Each example is a triplet of (question, passage, answer), with the title of the page as optional additional context.

PIQA PIQA (Bisk et al., 2020) composes of several natural language inference questions which evaluates the ability of physical commonsense reasoning for language models,

SIQA Social_IQa(SIQA) (Sap et al., 2019) is the first large scale benchmark for commonsense reasoning about social situations, which contains several multiple choice questions for probing emotional and social intelligence in a variety of everyday situations.

Race RACE (Lai et al., 2017) is a large-scale reading comprehension dataset collected from English examinations, which are designed for middle school and high school students.

Hellaswag Hellasawg (Zellers et al., 2019) is a dataset for commonsense natural language inference to evaluate the ability of language models to finish the specific sentence.

TruthfulQA TruthfulQA (Lin et al., 2021) aims to measure whether a language model is truthful in generating answers to questions. We transform this datasets into the multiple choice questions following previous practice.

MMLU MMLU (Hendrycks et al., 2021) is a massive multitask test consisting of multiple-choice questions from various branches of knowledge, including humanities, social sciences, hard sciences, and other areas that are important for some people to learn. It covers 57 tasks in total including elementary mathematics, US history, computer science, law, and more.

SuperGLUE SuperGLUE (Wang et al., 2022) is a enhanced version of GLUE containing more difficult language understanding tasks.

WIKI-AUTO WIKI-AUTO (Jiang et al., 2020) contains aligned sentences from English Wikipedia and Simple English Wikipedia, which evaluates the simplification abilities of the language models.

QQP Quora Question Pairs (QQP) consists of several pair of questions containing the same semantics, which can viewed as paraphrase pairs.

PersonaChat PersonaChat (Zhang et al., 2018) contains around 150k data triples formatted as (profile, conversation, response).

Flan-T5 Flan-T5 (Wei et al., 2021) is trained on FLAN instruction data based on the T5 pre-trained language models.

Instruct-XMLR Instruct-XMLR (Xiao et al., 2024) is instruction based fine-tuned based on XLM-R with an encoder-only model archetecture.

C Results of Ablation Study

As mentioned in Section 2.2, we compare different decomposition ratios (α in {0.1,0.2,0.3,0.4}), and different factors (λ_1, λ_2) in {(0.3, 0.2), (0.5,0.2), (0.5,0.4), (0.4,0.2)} to control the masking ratio range for X, i.e., $\beta_X \sim U(0.1, 0.3)$, $U(0.3, 0.5)$, $U(0.1, 0.5)$, $U(0.2, 0.4)$, respectively. Besides, while the masking ratio for Y is typically sampled from a uniform distribution $U(0, 1)$, we also compare different variants where $\beta_Y \sim U(0.1, 0.9)$, $U(0.2, 0.8)$, and $U(0.3, 0.7)$. As the result shown in Table 1 (BiGLM) is trained based on the setting that $\alpha = 0.2, \beta_X \sim U(0.1, 0.3), \beta_Y \sim U(0, 1)$, we present the other ones in Table 8, we can find that all the variants (i.e., different decomposition ratios, masking ratios for X and Y) achieve comparable performance compared to the first version of BiGLM which is trained with , except that adopting relative larger masking ratio for X ($\beta_X \sim U(0.3, 0.5)$), indicating that larger masking ratio for X which leads to fewer unmasked tokens (i.e., useful context information) may increase the learning difficulty and is not suitable for BiGLM.

D Training Cost Analysis

According to the training detailed as mentioned in Section 4, we present the training cost (i.e., the GPU hours of the training process) in Table 6.

Model	GPU Hours
BiGLM-136M	11392
BiGLM-360M	22528
BiGLM-1.3B	45568
BiGLM-3.5B	100250

Table 6: The training cost.

E Related Works

The traditional BERT families (Devlin et al., 2018; Liu et al., 2019; Clark et al., 2020; He et al., 2020; Conneau et al., 2020; Warner et al., 2024; Fu et al., 2024) have demonstrated excellent performance in the NLP community. Their bidirectional modeling characteristic enables them to learn the context representations well and facilitate the capture of comprehensive semantic information, leading to success in various language understanding tasks. However, their language generation abilities are relatively weak compared with autoregressive causal language models (Lewis et al., 2019; Song et al., 2019). Several previous works have introduced different methods to empower them with language generation abilities via non-autoregressive generation manner (Chan and Fan, 2019; Jiang et al., 2021; Su et al., 2021; Liang et al., 2023b,a; Xiao et al., 2024). However, the performance does not reach the level of strong AR models. Besides, they focus on simple generation tasks, and always rely on the fine-tuning process. As a result, the generation potential of the vanilla BERT-family without fine-tuning, is under-explored. Furthermore, more capabilities of BERT-family should be evaluated with the constantly updating requirements for language models. In this paper, we fill-in this blank and pre-train a new version of BERT-family, demonstrating their potential for building scalable, general, and competitive large language models. Among the previous works in BERT families, Samuel has pointed out that BERT families can be generative in-context learners and be adopted for solving reasoning task, their models generate the target tokens one-by-one in left-to-right order similar to AR models but exist relatively large performance gaps. Conversely, our proposed BiGLM generate the target tokens without ordering constraint and achieve comparable performance with current competitive AR models. Besides, more current work (Warner et al., 2024) also incorporates several enhanced training strategies which are also mentioned in Section 3 into the training process to enhance the capabilities of BERT family. However, they still focus on improving the performance in traditional NLU and text retrieval tasks which rely the understanding ability of BERT family. Comparatively, we conduct evaluation experiments in more range of testing scenarios such as text generation and common sense reasoning tasks to further broader the applications of BERT family.

Parameters	**BiGLM -136M**	**BiGLM -360M**	**BiGLM -1.3B**	**BiGLM -3.5B**
Num_layers	30	32	24	30
Hidden_size	576	960	2048	2560
Num_attn_heads	9	15	32	20
Num_key_value_heads	3	5	32	20
Init_std	0.02	0.02	0.013	0.013
Seq_length	2048	2048	2048	2048
Batch_size	1024	1024	1024	1024
Total_train_iters	300000	300000	300000	300000
Learning_rate	6e-4	6e-4	6e-4	6e-4
Annealing_iters	60000	60000	60000	60000
Annealing_min_lr	6e-5	6e-5	6e-5	6e-5
Clip_grad	1.0	1.0	1.0	1.0
Adam_beta	(0.9,0.95)	(0.9, 0.95)	(0.9, 0.95)	(0.9, 0.95)
Weight_decay	1e-2	1e-2	1e-2	1e-2

Table 7: Details of the pre-training models and setting.

Methods	**ARC-E**	**ARC-C**	**PIQA**	**Sciq**	**Wino.**	**LogiQA**	**Race**	**SIQA**	**BoolQ**	**Hella.**	**Truth.**	**AVG.**
BiGLM	52.95	26.37	60.55	85.1	49.80	28.17	28.04	38.16	60.64	34.56	24.96	**44.48**
$\alpha = 0.1$	52.23	25.72	60.26	84.5	51.60	28.45	27.53	37.49	60.51	34.23	25.02	44.32
$\alpha = 0.3$	52.14	25.46	60.41	85.6	50.43	27.19	27.46	38.37	61.13	34.02	24.68	44.29
$\alpha = 0.4$	51.60	25.09	62.02	86.0	51.22	26.73	30.14	37.05	59.17	33.14	24.85	44.27
$\beta_X \sim U(0.1, 0.5)$	50.63	23.72	60.06	83.8	52.40	28.73	28.61	38.39	60.61	33.89	24.96	44.16
$\beta_X \sim U(0.3, 0.5)$	51.05	24.06	59.85	83.6	52.17	26.27	27.75	36.75	59.14	32.80	24.31	43.43
$\beta_X \sim U(0.2, 0.4)$	51.22	23.63	60.12	83.9	52.33	27.19	28.52	37.95	60.74	33.51	24.97	44.01
$\beta_Y \sim U(0.1, 0.9)$	52.64	25.34	59.74	84.9	50.59	28.67	28.13	37.37	59.14	33.67	24.84	44.09
$\beta_Y \sim U(0.2, 0.8)$	51.84	25.26	59.09	85.3	52.17	28.31	28.71	37.01	61.26	32.57	24.24	44.16
$\beta_Y \sim U(0.3, 0.7)$	52.74	25.17	60.45	84.9	51.14	28.17	28.13	37.70	59.62	34.26	25.04	44.30

Table 8: Results of various pre-training variants. **Wino.**, **Hella.**, and **Truth.** denote the WinoGrande, Hellaswag, and Truthfulqa datasets, **AVG.** denotes average result. *attn.* denotes the attention masking strategy.