ICLR

When Scaling Meets LLM Finetuning: The Effect of Data, Model and Finetuning Methods

Biao Zhang [1], Zhongtao Liu [2], Colin Cherry [2], Orhan Firat [1]

Presenter: Ye YUAN

Aug 19th, 2024

Google DeepMind [1]

Google Research [2]

Contents

- Author Introduction
- The reasons why I choose this paper
- Motivations and Preliminary
- Experiment Setup
- Experiments and Results
- Discussion and Conclusions
- QA

Authors Introduction

Biao Zhang

- Full-time Researcher at Google Deepmind since 2023. Received Ph.D. from University of Edinburgh, Master's and Bachelor's degrees from Xiamen University.

Zhongtao Liu

- Full-time Software Engineer at Google Research. Received Ph.D. from National University of Singapore, Master's from Georgia Tech, Bachelor's from Nanjing Science and Technology University.

Colin Cherry

- Research Scientist at Google Research, Montreal. Previously, Officer in Text Analytics at National Research Council Canada, and Researcher in the Natural Language Processing group at Microsoft Research. Received Ph.D. from University of Alberta.

Orhan Firat

- Principal Scientist, Director at Google DeepMind, NYC. Currently oversee the evaluation team of the Gemini Program; Work on Science of Scaling, Multitask Learning (and multilingual models), and Deep Learning along with Benchmarking.

The Reasons why I want to share this paper

- As a researcher from academia, we care more about large language models (LLMs) fine-tuning, rather than LLMs pre-training, due to limited computational resources.
- However, existing studies about scaling laws mainly focus on pre-training: ☹
  - "Scaling Laws for Neural Language Models" from OpenAI [1].
  - "Training Compute-Optimal Large Language Models" Chinchilla from Google [2].
  - …
- This study fills the gap of scaling laws for fine-tuning LLMs ☺.

[1] J. Kaplan et al., Scaling Laws for Neural Language Models. 2020. [Online]. Available: https://arxiv.org/abs/2001.08361
[2] J. Hoffmann et al., Training Compute-Optimal Large Language Models. 2022. [Online]. Available: https://arxiv.org/abs/2203.15556

Motivation

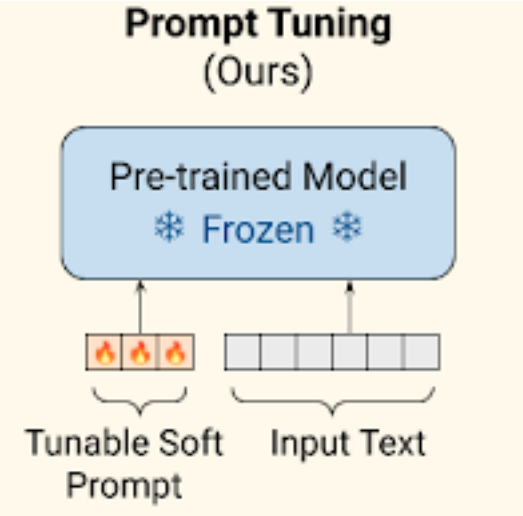
- Many potential factors affecting the performance of LLM fine-tuning:
  - Pre-training conditions: LLM model size; Pre-training data size.
  - Fine-tuning: Downstream tasks; Fine-tuning data size; Fine-tuning method.

- Benefits:
  - Improve fine-tuning performance.
  - Understand pre-training from the fine-tuning perspectives.

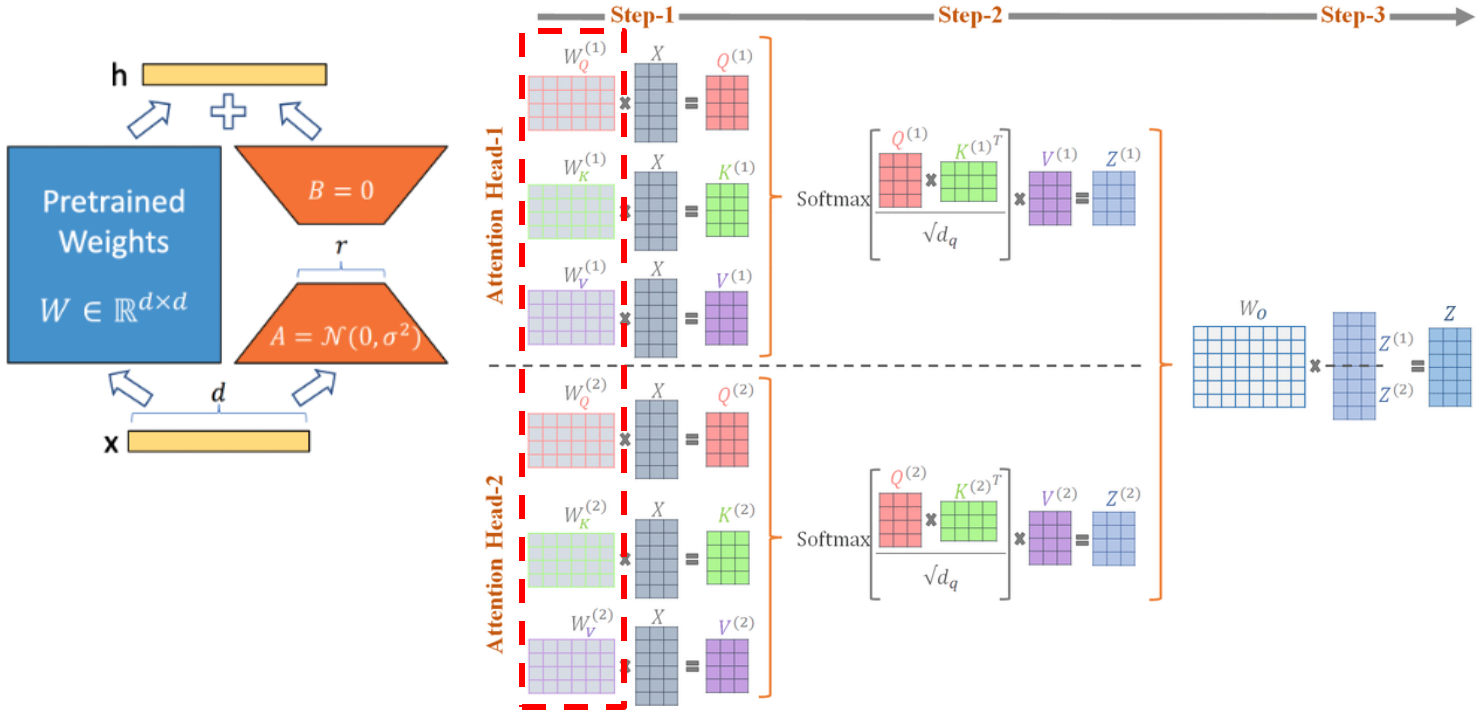
Preliminary (Fine-tuning Methods)

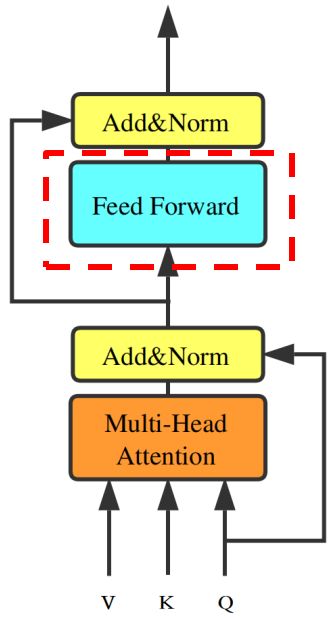
- Full-model tuning (FMT): updates all LLM parameters.
- Parameter-efficient tuning (PET): optimizes parts of (newly added) parameters.

Prompt Tuning (Soft Prompting) [1]:



Low Rank Adaptation (LoRA) [2]:





[1] B. Lester, R. Al-Rfou, and N. Constant, The Power of Scale for Parameter-Efficient Prompt Tuning. 2021. [Online]. Available: https://arxiv.org/abs/2104.08691
[2] E. J. Hu et al., LoRA: Low-Rank Adaptation of Large Language Models. 2021. [Online]. Available: https://arxiv.org/abs/2106.09685

Preliminary (Metrics)

- Perplexity [1]:
  - Perplexity is defined as the exponentiated average negative log-likelihood of a sequence. If we have a tokenized sequence $X=(x_0,x_1,\ldots,x_t)$, then the perplexity of X is :

$$\mathrm{PPL}(X) = \exp\left\{-\frac{1}{t}\sum_{i}^{t}\log p_\theta(x_i|x_{<i})\right\}$$

- BLEURT (used for general generation task) [2]:
  - BLEURT is an evaluation metric based on a calibrated BERT model. In short, it takes generated sentence and the ground-truth sentence, then outputs a similarity score.
- RougeL (used for machine translation task) [3]:
  - X is the ground-truth sentence, Y is the generated sentence.
  - m is the number of token in X, n is the number of token in Y.
  - LCS(·) is the number of common tokens but considering order.

$$R_{lcs} = \frac{LCS(X,Y)}{m}$$

$$p_{lcs} = \frac{LCS(X,Y)}{n}$$

$$F_{lcs} = \frac{(1+\beta^2)R_{lcs}P_{lcs}}{R_{lcs}+\beta^2 P_{lcs}}$$

[1] [Online]. Available: https://huggingface.co/docs/transformers/en/perplexity
[2] T. Sellam, D. Das, and A. P. Parikh, BLEURT: Learning Robust Metrics for Text Generation. 2020. [Online]. Available: https://arxiv.org/abs/2004.04696
[3] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," in Text Summarization Branches Out, Jul. 2004, pp. 74–81.

Experiments Setup

- **Downstream tasks**
  - machine translation and multilingual summarization.
  - WMT14 English-German (En-De) and WMT19 English-Chinese (En-Zn) for translation
  - A self-customized dataset MLSum for multilingual summarization, where each article is prepended a prompt indicating its language *"Summarize the following document in {language}".*
- **Pretrained LLMs**
  - Decoder-only Transformer with Multi-query attention
  - One set of En-De LLMs on 283B tokens, one set of En-Zn LLMs on 206B tokens.
  - Each set of models includes parameter sizes 1B, 2B, 4B, 8B, 16B

Experiments Setup

- **Fine-tuning Methods:**
  - Full-Model Tuning, Prompt Tuning, Low-Rank Adaptation
- This study explores **four factors** for the scaling (use fine-tuning data sizes as core):
  - LLM Model Sizes: Performance v.s. Model Sizes v.s. Fine-tuning Data sizes
  - Pre-training Data Sizes: Performance v.s. Pre-training Data sizes v.s. Fine-tuning Data sizes on 1B model
  - PET Parameter Sizes: Performance v.s. PET parameter Sizes v.s. Fine-tuning Data sizes on 1B model
- **Scaling law evaluation:**
  - To test the extrapolation ability of the fitted scaling, the authors use a held-out set to evaluate.

Details of Scaling Settings for Different Factors

LLM Model Sizes		1B, 2B, 4B, 8B, **16B**
Pretraining Data Sizes	En-De LLM	84B, 126B, 167B, **209B**, 283B
	En-Zh LLM	84B, 105B, 126B, 147B, **167B**, 206B
PET Parameter Sizes	Prompt Length	50, 100, 150, 200, 300, 400, **600**
	LoRA Rank	4, 8, 16, 32, 48, 64, **128**
Finetuning Data Sizes	Prompt & LoRA	8K, 10K, 20K, 30K, 40K, 50K, 60K, 70K, 80K, 90K, **100K**
	FMT– WMT En-De	100K, 500K, 1M, 1.5M, 2M, 2.5M, 3M, 3.5M, 4M, **4.5M**
	FMT– WMT En-Zh	1M, 2M, 3M, 4M, 5M, 10M, 15M, 20M, **25M**
	FMT– MLSum	100K, 200K, 300K, 400K, 500K, 600K, 700K, 800K, **900K**

*The data sizes are counted in tokens.

The bolded are the held-out data point, which are used for test the fitted scaling laws for their extrapolation ability.

How to Fit a Scaling Law?

- Multiplicative Scaling Law:

$$\hat{\mathcal{L}}(X, D_f) = A * \frac{1}{X^{\alpha}} * \frac{1}{D_f^{\beta}} + E$$

- Additive Scaling Law [1]:

$$\hat{\mathcal{L}}(X, D_f) = \frac{A}{X^{\alpha}} + \frac{B}{D_f^{\beta}} + E$$

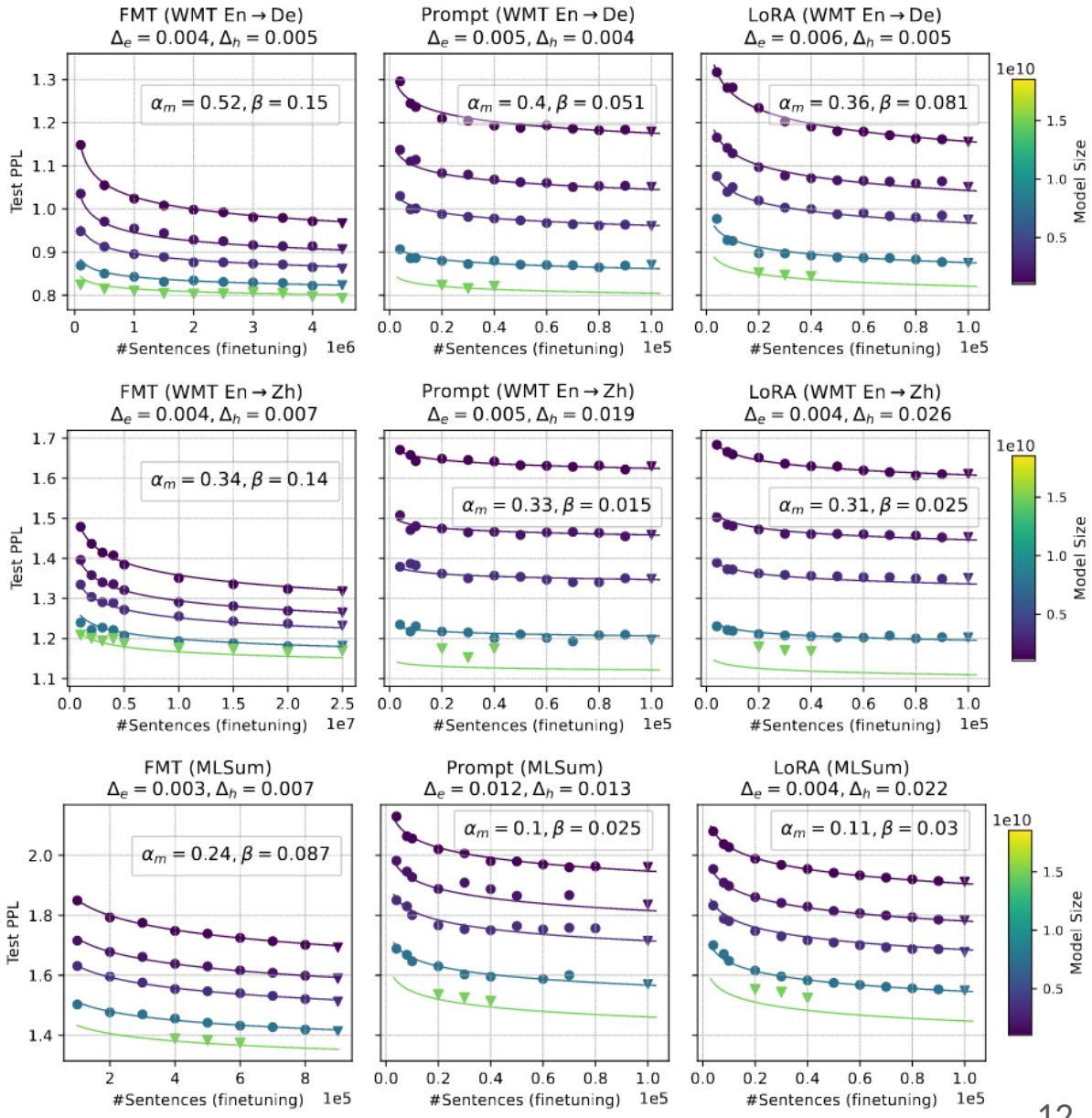
{A, E, α, β} are data-specific parameters to be fitted, D_f denotes finetuning data size, and X refer to each of the other scaling factors.

Scaling Factor	Multiplicative				Additive			
	FMT	Prompt	LoRA	Avg	FMT	Prompt	LoRA	Avg
LLM Model Size	0.0052	0.0043	0.0047	**0.0048**	0.012	0.0076	0.0045	0.0079
Pretraining Data Size	0.0057	0.0061	0.0084	**0.0068**	0.0048	0.0075	0.0082	0.0069
PET parameter size	-	0.005	0.0031	**0.004**	-	0.0069	0.0032	0.005

[1] J. Hoffmann et al., Training Compute-Optimal Large Language Models. 2022. [Online]. Available: https://arxiv.org/abs/2203.15556

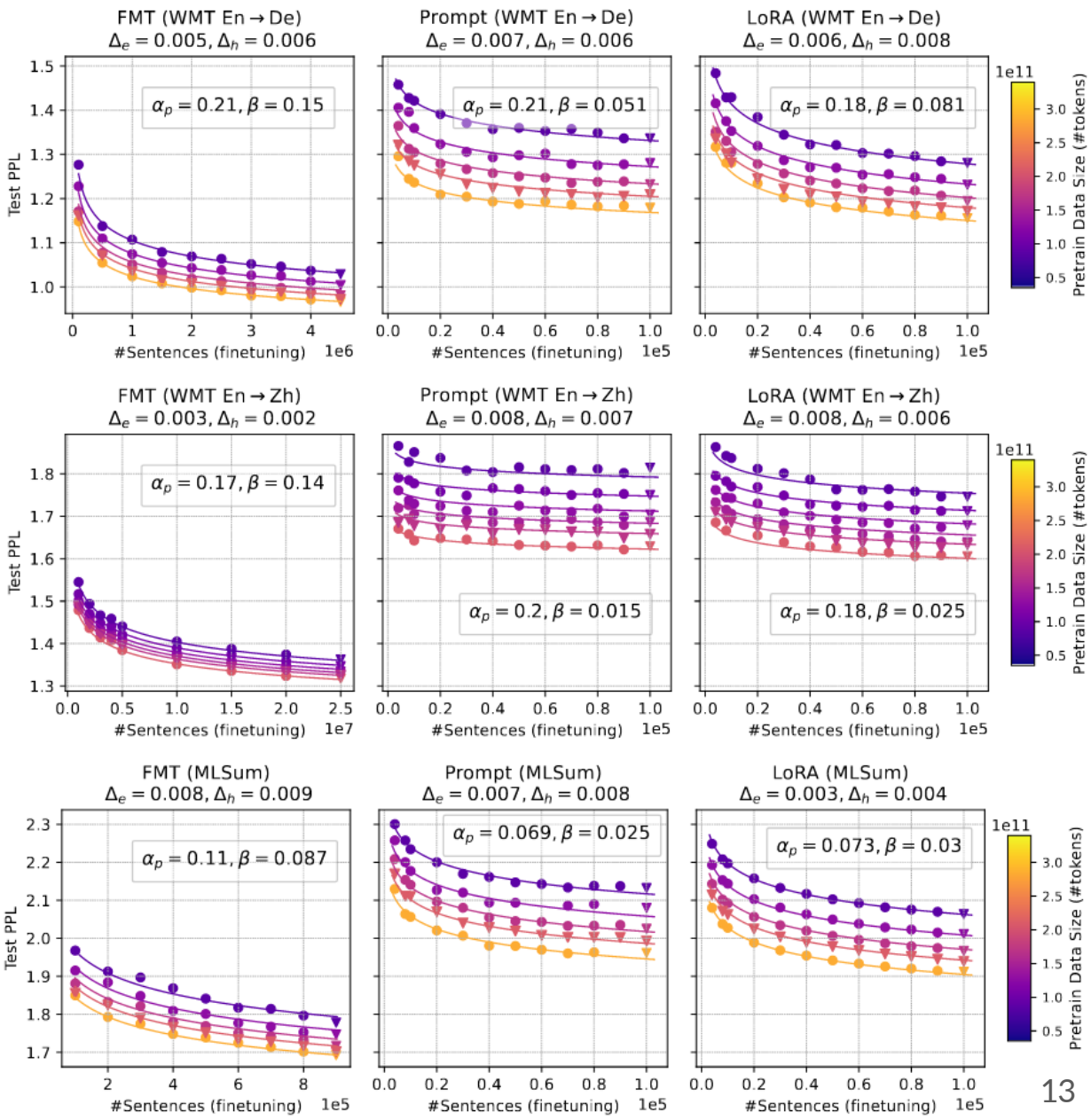
Performance v.s. Model Sizes v.s. Fine-tuning Data sizes

- Mean absolute derivation on the empirical fitting (∆e) and held-out (∆h).
- In general, multiplicative scaling law captures the scaling trend of model sizes under finetuning data scaling with small fitting and extrapolation errors.
- High mismatch when extrapolating to 16B, particularly for LoRA and Prompt on WMT19 En-Zh:
  - the insufficiency of empirical data
  - pretraining instability



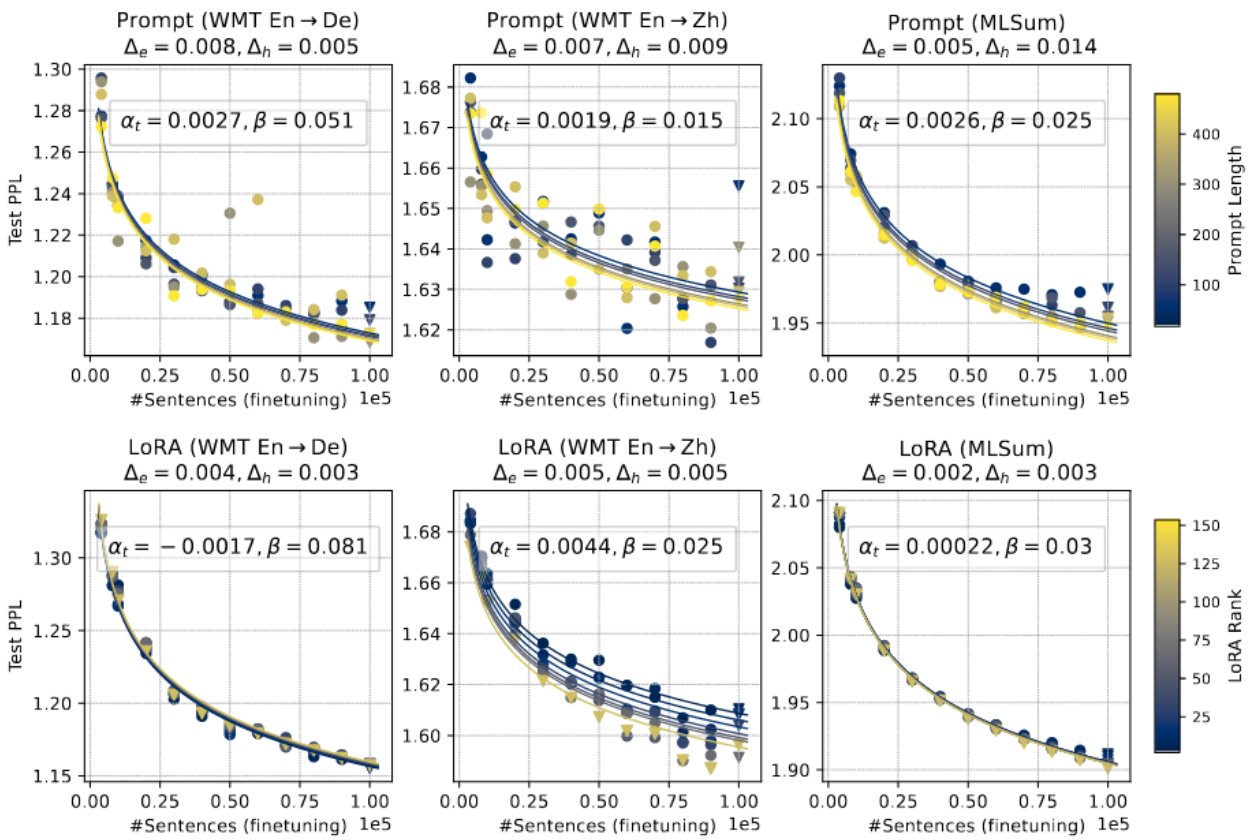
Performance v.s. Pre-training Data sizes v.s. Fine-tuning Data sizes

- Fix model size on 1B model
- Intuitively, finetuning relies on the knowledge encoded in the LLM, where model size and pretraining data size both matter.
- However, when fix β (the scaling exponent for fine-tuning data size), the scaling exponent for model size α_m often outnumbers that for pretraining data size α_p across finetuning methods and tasks, i.e. $\alpha_m > \alpha_p$.



Performance v.s. PET parameter Sizes v.s. Fine-tuning Data sizes

- Increasing PET parameter sizes affects finetuning performance marginally, and even results in inverse scaling in some settings.
- Scaling Prompt length suffers from training instability, which has also been seen in previous studies [1, 2]. In this respect, LoRA is more stable and reliable.



[1] E. J. Hu et al., LoRA: Low-Rank Adaptation of Large Language Models. 2021. [Online]. Available: https://arxiv.org/abs/2106.09685
[2] B. Lester, R. Al-Rfou, and N. Constant, The Power of Scale for Parameter-Efficient Prompt Tuning. 2021. [Online]. Available: https://arxiv.org/abs/2104.08691

Influence of Fine-tuning Data Size on Different Methods

- β for FMT is often significantly higher than that for PET across settings, indicating that FMT is more data-hungry and also benefits more from increasing finetuning data.
- LoRA often slightly surpasses that for Prompt, achieving better finetuning performance with more finetuning data than Prompt.

Table 4: Fitted scaling parameters for different settings.

Params	WMT14 En-De			WMT19 En-Zh			MLSum		
	FMT	Prompt	LoRA	FMT	Prompt	LoRA	FMT	Prompt	LoRA
Scaling for LLM model size and finetuning data size									
A_m	1.2×10^5	3.9×10^3	2.1×10^3	3.3×10^3	8.5×10^2	6.6×10^2	3.3×10^2	23	26
α_m	0.52	0.4	0.36	0.34	0.33	0.31	0.24	0.1	0.11
Scaling for pretraining data size and finetuning data size									
A_p	6.3×10^2	2.7×10^2	1.4×10^2	2.4×10^2	2×10^2	1.3×10^2	42	16	17
α_p	0.21	0.21	0.18	0.17	0.2	0.18	0.11	0.069	0.073
Scaling for PET parameter size and finetuning data size									
A_t	-	1	1.4	-	1	1.2	-	2.6	2.4
α_t	-	0.0027	−0.0017	-	0.0019	0.0044	-	0.0026	0.000 22
E	0.75	0.62	0.62	1	0.77	0.73	0.98	0.000 51	0.2
β	0.15	0.051	0.081	0.14	0.015	0.025	0.087	0.025	0.03

Influence of Model Size on Different Methods

- Since the majority of LLM parameters is frozen during finetuning, PET relies heavily on the encoded knowledge in pretrained LLMs when adapting them to downstream tasks.

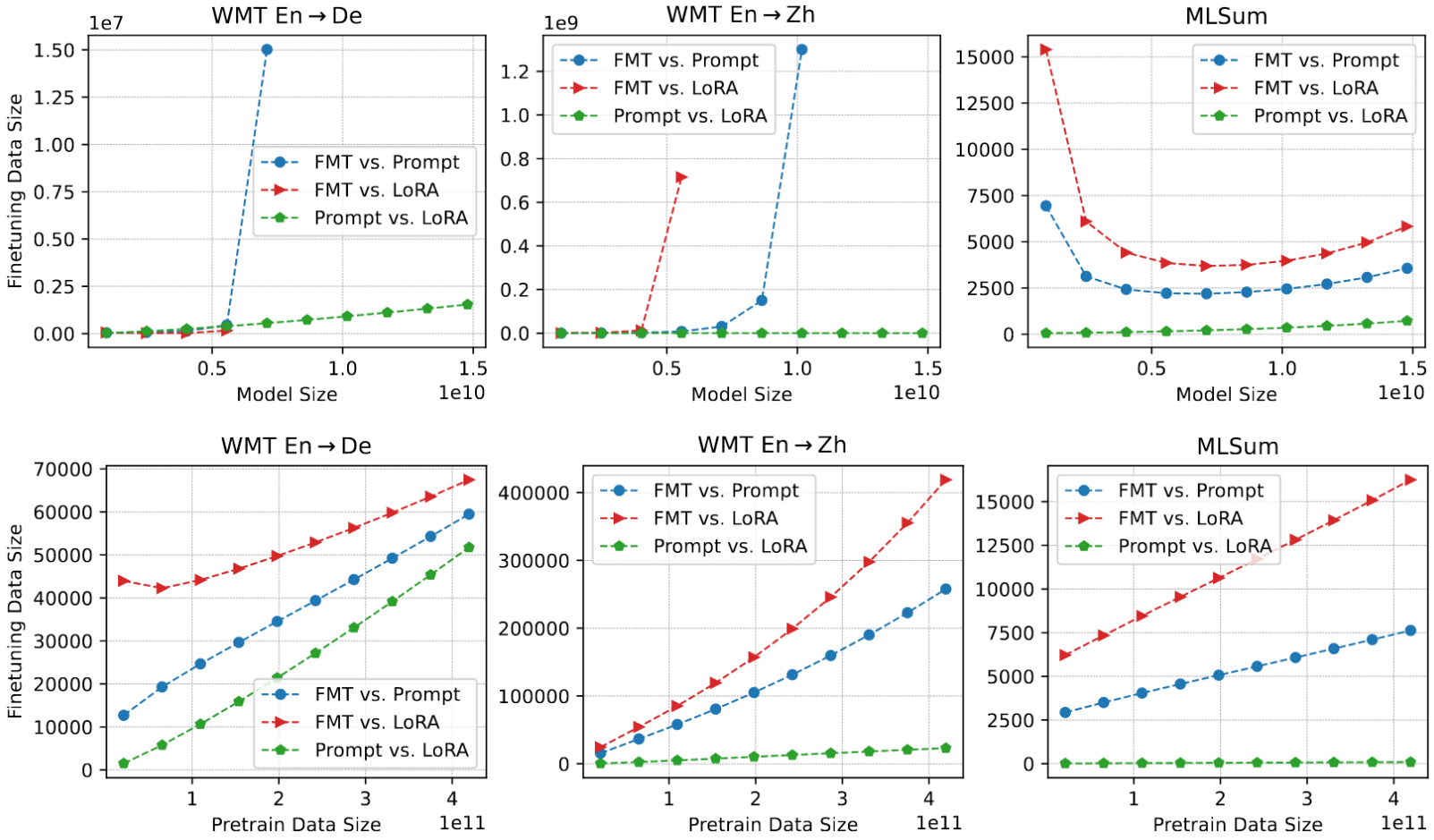
- α_m and α_p are clearly larger than β in PET.

Table 4: Fitted scaling parameters for different settings.

Params	WMT14 En-De			WMT19 En-Zh			MLSum		
	FMT	Prompt	LoRA	FMT	Prompt	LoRA	FMT	Prompt	LoRA
Scaling for LLM model size and finetuning data size									
A_m	1.2×10^5	3.9×10^3	2.1×10^3	3.3×10^3	8.5×10^2	6.6×10^2	3.3×10^2	23	26
α_m	0.52	0.4	0.36	0.34	0.33	0.31	0.24	0.1	0.11
Scaling for pretraining data size and finetuning data size									
A_p	6.3×10^2	2.7×10^2	1.4×10^2	2.4×10^2	2×10^2	1.3×10^2	42	16	17
α_p	0.21	0.21	0.18	0.17	0.2	0.18	0.11	0.069	0.073
Scaling for PET parameter size and finetuning data size									
A_t	-	1	1.4	-	1	1.2	-	2.6	2.4
α_t	-	0.0027	−0.0017	-	0.0019	0.0044	-	0.0026	0.000 22
E	0.75	0.62	0.62	1	0.77	0.73	0.98	0.000 51	0.2
β	0.15	0.051	0.081	0.14	0.015	0.025	0.087	0.025	0.03

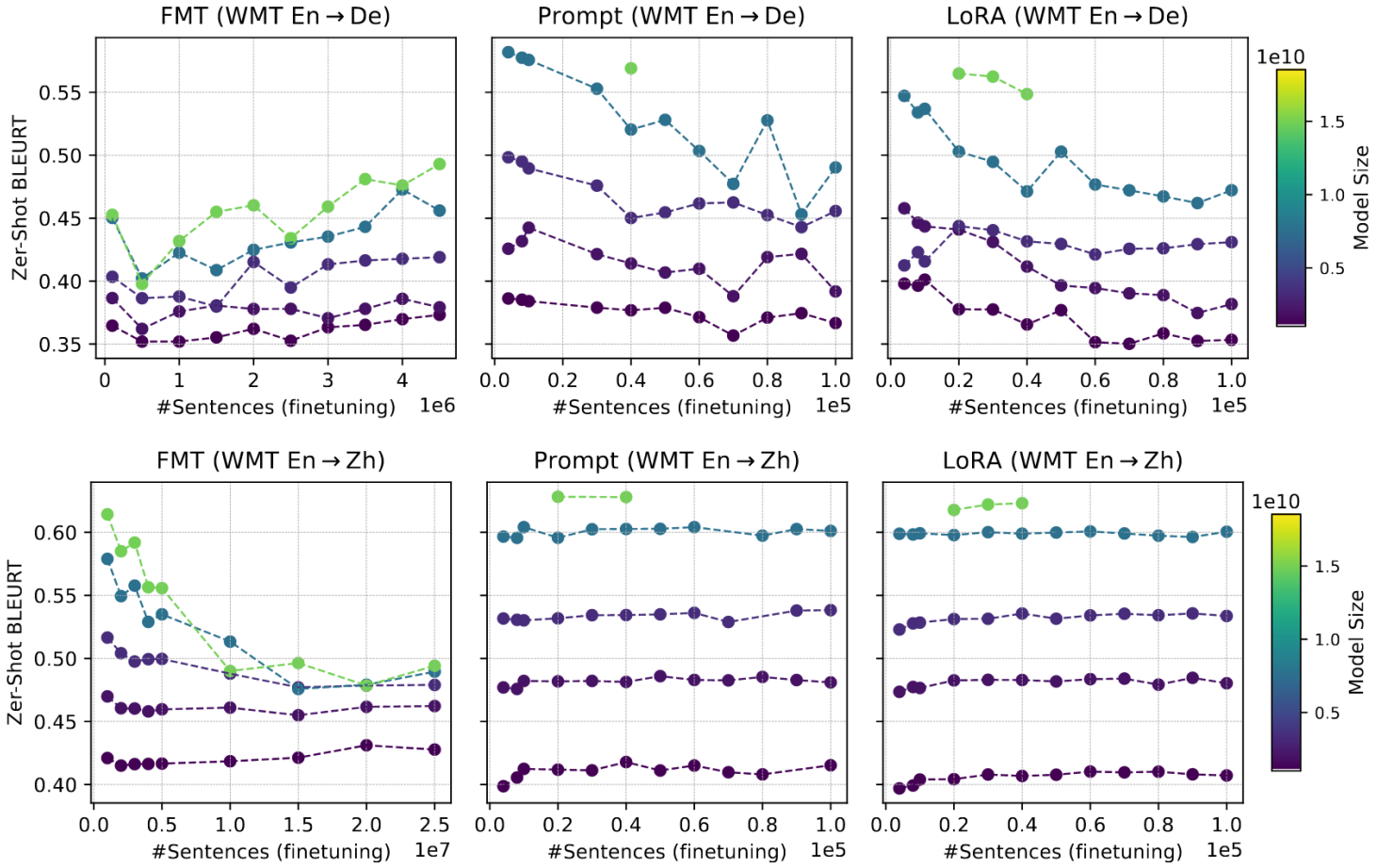
Which finetuning method should we apply for a given task?

- Unfortunately, there is no universal answer!
- The scaling trend and actual value are highly dependent on the downstream task: critical points for one task can hardly generalize to other tasks.



How does finetuning affect the generalization capability?

- Whether finetuning benefits generalization is method- and task-dependent.
- Overall, Prompt and LoRA achieve relatively better results than FMT.
- Suggests that when generalization capability is a big concern, PET should be considered.



Conclusion and Discussion

- LLM finetuning benefits more from LLM model scaling than pretraining data scaling across tasks and methods.
- Scaling PET parameters is ineffective, delivering limited gains for both LoRA and Prompt.
- Finetuning data have more pronounced influence on FMT than PET, where LoRA scales better than Prompt.
- PET depends more on LLM model and pretraining data scaling than finetuning data scaling across settings.
- There is no universal answer for which fine-tuning method is optimal for a given task.
- When generalization capability is a big concern, PET should be considered.

Thanks for your attention!

Questions

- Could you briefly explain the three fine-tuning methods studied by this paper?
- How does this paper validate the fitted scaling law?
- Are these conclusions useful for your research? How could you integrate these conclusions to your own projects?

$$u(x|w_1,\cdots,w_{i-1}) = \begin{cases} 1, & x = w_i \\ 0, & x \neq w_i \end{cases}$$

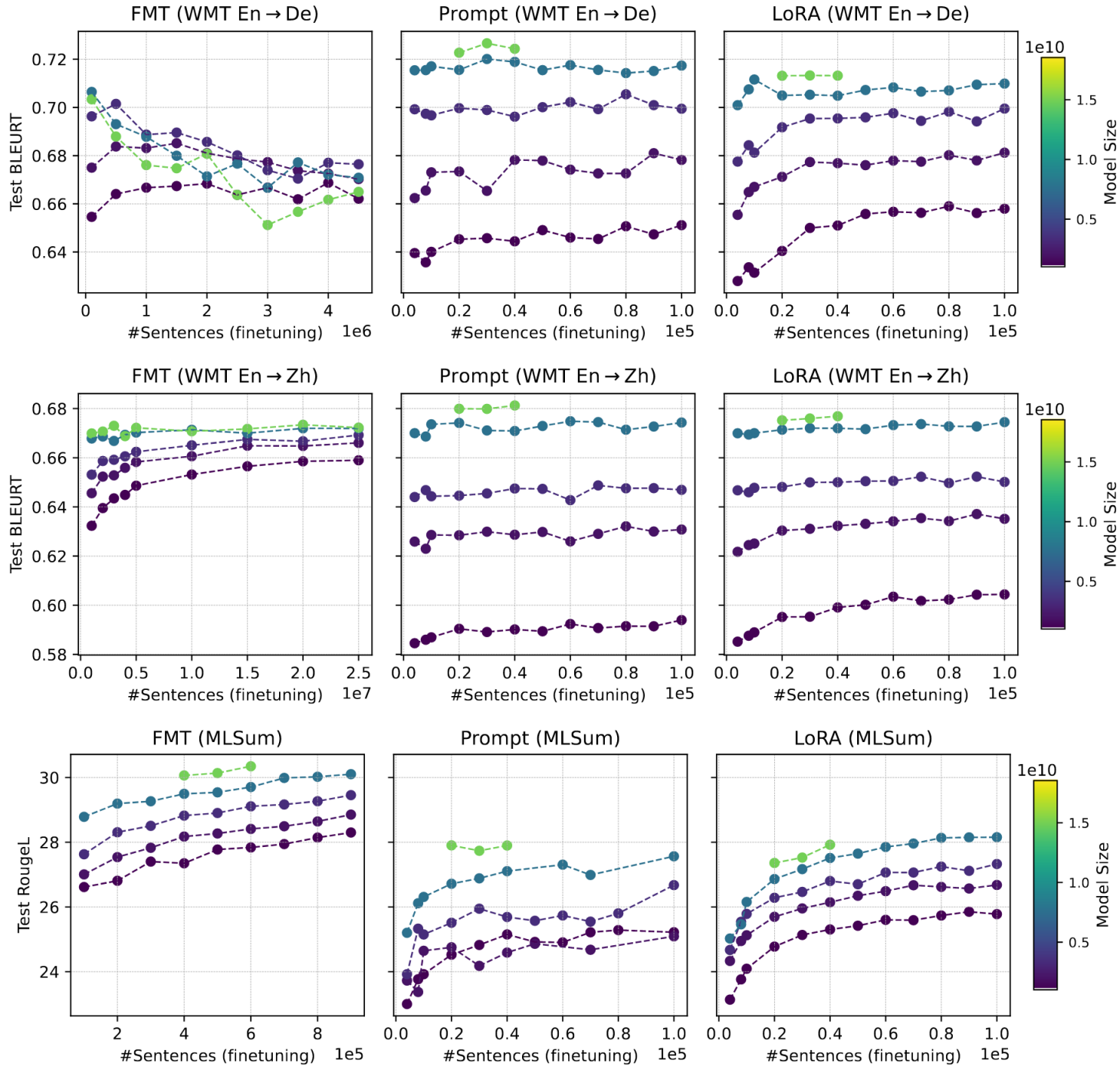
$$H(u,v) = -\sum_x u(x)\log v(x)$$

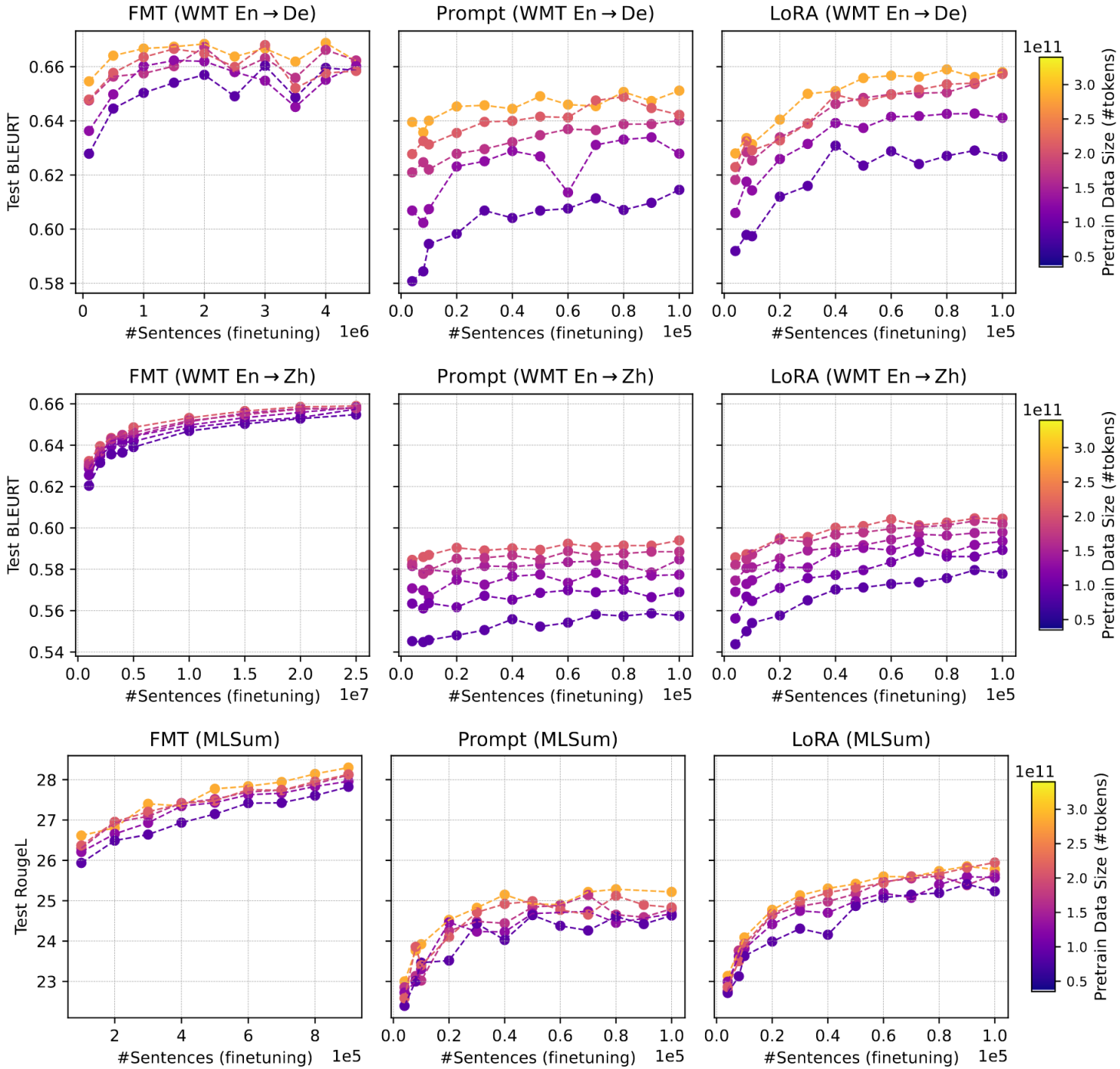
$$= -\frac{1}{m}\sum_{i=1}^{m}\left(\sum_x u(x|w_1,\cdots,w_{i-1})\log p(x|w_1,\cdots,w_{i-1})\right)$$

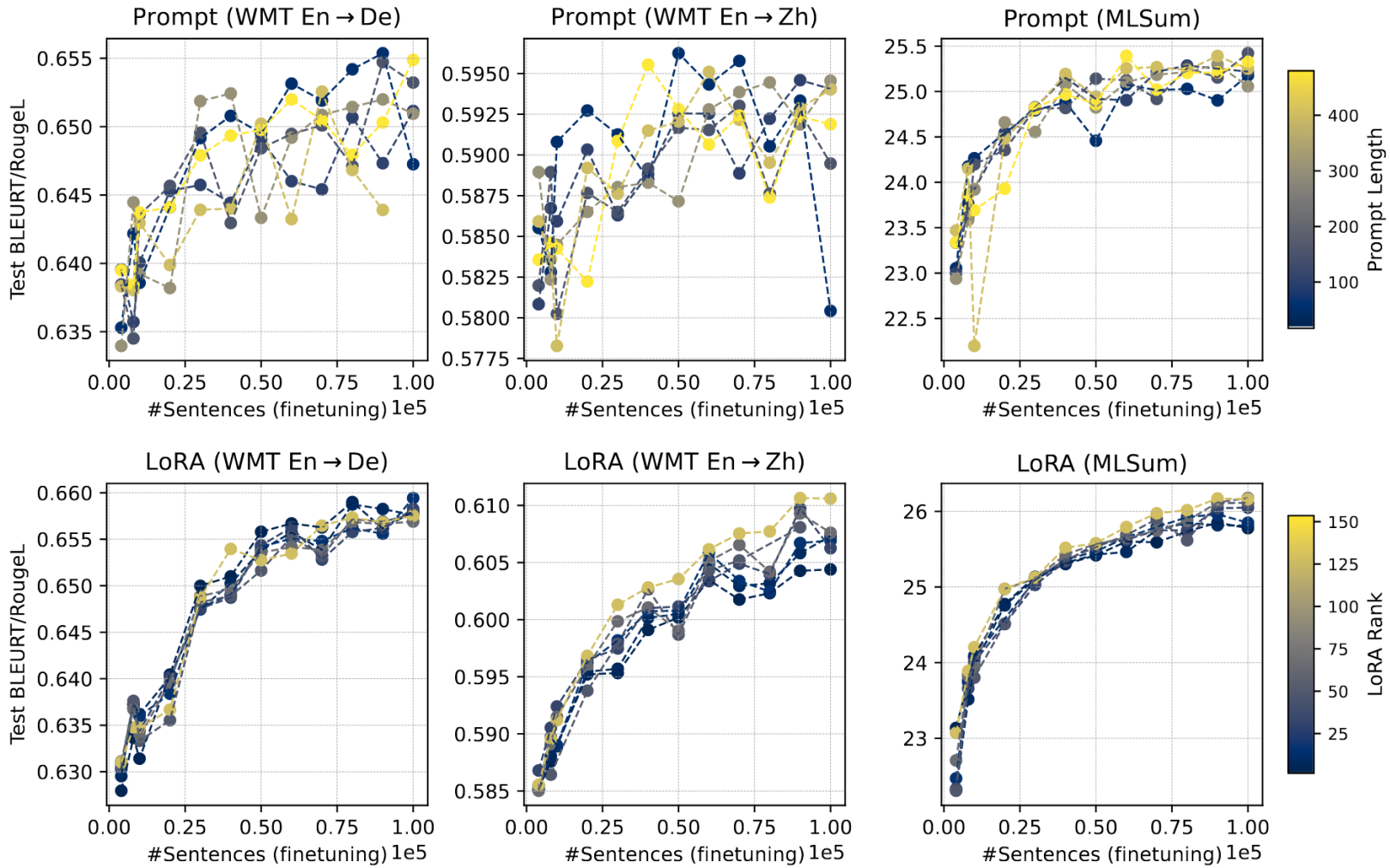
$$= -\frac{1}{m}\sum_{i=1}^{m}\left(1.0\times\log p(w_i|w_1,\cdots,w_{i-1}) + \sum_{x\neq w_i} 0.0\times\log p(x|w_1,\cdots,w_{i-1})\right)$$

$$= -\frac{1}{m}\sum_{i=1}^{m}\log p(w_i|w_1,\cdots,w_{i-1})$$

$$= \log(perplexity(S))$$


FMT (WMT En → De)
Prompt (WMT En → De)
LoRA (WMT En → De)
Test BLEURT
#Sentences (finetuning)
1e6
1e5
FMT (WMT En → Zh)
Prompt (WMT En → Zh)
LoRA (WMT En → Zh)
1e7
FMT (MLSum)
Prompt (MLSum)
LoRA (MLSum)
Test RougeL
1e10
Model Size


FMT (WMT En → De)
Prompt (WMT En → De)
LoRA (WMT En → De)
FMT (WMT En → Zh)
Prompt (WMT En → Zh)
LoRA (WMT En → Zh)
FMT (MLSum)
Prompt (MLSum)
LoRA (MLSum)
Test BLEURT
Test RougeL
#Sentences (finetuning)
1e6
1e5
1e7
1e11
Pretrain Data Size (#tokens)


Prompt (WMT En → De)
Prompt (WMT En → Zh)
Prompt (MLSum)
LoRA (WMT En → De)
LoRA (WMT En → Zh)
LoRA (MLSum)
Test BLEURT/RougeL
#Sentences (finetuning)
1e5
Prompt Length
LoRA Rank