DiffTune: A Diffusion-Based Approach to Diverse Instruction-Tuning Data Generation

Suyuchen Wang
Université de Montréal / Mila
Montréal, Quebéc, Canada
suyuchen.wang@umontreal.ca

Bang Liu
Université de Montréal / Mila
Montréal, Quebéc, Canada
bang.liu@umontreal.ca

Abstract

Instruction tuning has become pivotal in enhancing the adaptability and responsiveness of Large Language Models (LLMs) to human instructions. Despite its critical role, current methods for generating instruction-tuning datasets exhibit significant bottlenecks, primarily in terms of high cost and limited diversity. However, as previously shown in the literature, the diversity of an instruction-tuning dataset is crucial to LLM's downstream performance. To address these challenges, we propose a Diffusion Language Model (DiffLM)-based technique to generate unlimited diverse instructions at a low cost. Specifically, we have enhanced the variability of instructions by strategically modifying the sampling process within the DiffLM. Our method presents the opportunity to augment any existing instruction-tuning dataset, thereby enriching its content and potential utility. Both automatic and human evaluation show that our generated instructions achieve high quality and better n-gram diversity than the original dataset. Instruction tuning of LLaMA on the augmented dataset delivers better instruction following capability and superior performance on a broad set of benchmarks, indicating the effectiveness of our instruction generation method.

1 Introduction

Large Language Models (LLMs), particularly following the advent of ChatGPT, have seen a surge in popularity due to their impressive performance capabilities [1, 2]. To maximize LLMs' potential and adapt pre-trained models to specific domains or downstream tasks, instruction tuning emerges as an indispensable step [3, 4]. It involves the generation of bespoke datasets that guide Large Language Models (LLMs) to respond more effectively to human instructions across varying tasks and domains.

Existing instruction-tuning techniques generally fall into two categories: human-labeled and machine-generated approaches. Human-labeled methods [3–5] are highly accurate and contextually rich, but is difficult to scale up and expensive to procure. Machine-generated techniques Wang et al. [6], Peng et al. [7], Honovich et al. [8] are easily scalable but lack the necessary diversity, creating a gap between the instructions in the dataset and real-world user prompts. Also, generating datasets by querying commercial language models also involves costs, which could be substantial [9, 7] The inherent limitations of existing instruction-tuning techniques underscore the need for a more effective, scalable, and cost-efficient approach, forming our research's central theme.

Given the aforementioned challenges, we propose DiffTune, a novel data generation technique utilizing the Diffusion Language Model (DiffLM). The diffusion model, as a kind of generative model, works by simulating a process of random walks from a simple initial distribution toward the target distribution, resulting in nuanced and detailed data generation. Building upon this foundation, DiffTune innovatively leverages the inherent properties of a Diffusion Language Model (DiffLM). By replacing the original sampling strategy within the DiffLM with our topic diversity enhancing

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

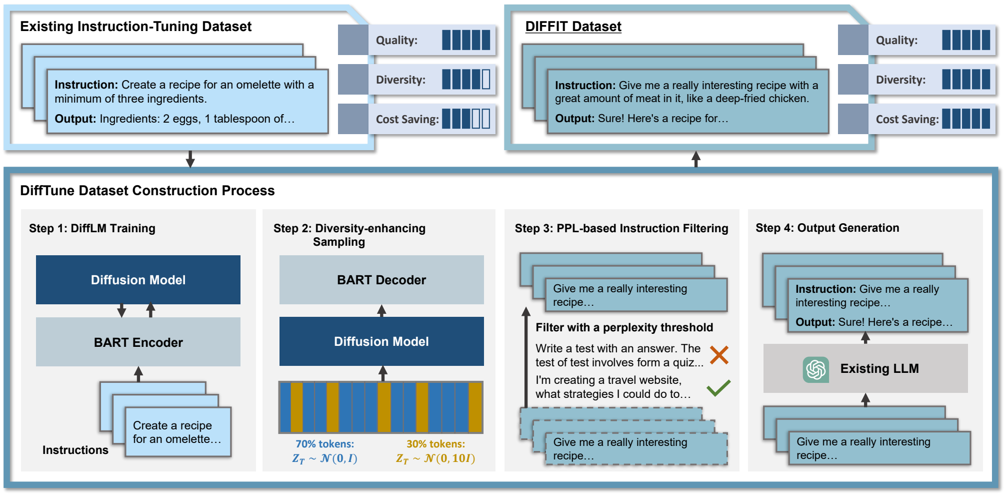

Figure 1: The dataset collection process of DiffTune.

sampling, DiffTune is imposed to generate high-quality instruction-tuning datasets with enhanced diversity at a lower cost.

We put our novel DiffTune-generated dataset to the test by finetuning an accessible LLM, LLaMA [10]. Our methodology involved augmenting or replacing the original datasets with the instructions generated by DiffTune. The LLaMA model, when finetuned with our DiffTune-generated dataset, displayed a remarkable increase in its instruction-following capabilities in terms of higher validity and human preference. This underscores the potential of DiffTune to optimize LLM performance cost-effectively.

2 Data Collection for DIFFIT

This section introduces the fully automatic collection process of our instruction tuning dataset DIFFIT. The overall data collection process is shown in Figure 1.

2.1 Diffision Language Model Training

Given an existing instruction-tuning dataset $\{(X^e_t, Y^e_t)\}$, where X^e_t and Y^e_t are the instruction and output of an instance in the dataset, we train a DiffLM on its instruction set $\{X_t\}$ as described previously. After this stage, we obtain a trained DiffLM, denote it M_{Diff}. We can now sample from it: 1) sample a length from the empirical length distribution of the existing instruction set $l_i \sim \mathcal{L}(\{X_t\})$, and 2) sample a noise $\boldsymbol{Z}_T \in \mathbb{R}^{l_i \times d} \sim \mathcal{N}(0, \boldsymbol{I})$. The generation $M_{\text{Diff}}(\boldsymbol{Z}_T)$ is a sequence in the same domain of the original instruction set $\{X_t\}$.

2.2 Diffusion Language Model Sampling

We sample from the trained M_{Diff} to generate new, diverse, and high-quality instructions. Lovelace et al. [11] demonstrated that DiffLM could generate diversified text sequences with low memorization of its training set when using a noise sampled from $\mathcal{N}(0, \boldsymbol{I})$. In our method, we further increase the diversity of our sampled instruction set by adopting an innovative sampling strategy to cover the rare topics, concepts, and formats mapped to the long tail of the sampling noise distribution.

Inspired by the in-breadth evolving strategy mentioned in Xu et al. [12], we propose the topic diversity enhancing sampling strategy. After sampling the noise from a standard Gaussian $\boldsymbol{Z}_T \in \mathbb{R}^{l_i \times d} \sim \mathcal{N}(0, \boldsymbol{I})$, we randomly select 30% of the tokens and sample them from a distribution of a much higher variance $\mathcal{N}(0, 10\boldsymbol{I})$. This strategy resembles the process of randomly inserting rare tokens into the sequence. With the remaining 70% originally sampled tokens to control the overall format and M_{Diff}'s powerful BART decoder, the generation's quality is only slightly compromised. The post-processing step can mitigate the slightly lower generation quality.

Dataset	Size	Avg len.	#Unique Tokens↑	4-gram Rep.↓	4-gram SelfBLEU↓
Unnat. Inst.	68478	**93.92**	42455	0.61	0.85
Self-Instruct	82439	35.59	19920	0.70	0.90
Alpaca	52002	22.79	21027	0.43	0.69
GPT4-Alpaca	52002	22.79	21027	0.43	0.69
Code-Alpaca	20022	29.90	8671	0.55	0.79
OASST1	55668	25.09	36150	0.83	0.96
S.A.D.	52000	30.24	**55455**	0.19	<u>0.51</u>
DIFFIT	52000	28.64	<u>49322</u>	**0.14**	**0.49**

Table 1: Evaluation of existing similarly-sized datasets' input instructions in terms of different diversity metrics. S.A.D.: 52K instructions sampled from the concatenation of ShareGPT + Alpaca + Dolly's instruction set with stratified sampling. ↑: Higher is better. ↓: Lower is better. The best and second-best results are labeled in **bold** and <u>underline</u>, respectively.

2.3 Instruction Post-Processing

In this step, we filter out the sampled instructions from M_{Diff} with a perplexity threshold. We use GPT2-Large to compute the perplexity. This simple yet effective post-processing strategy can drastically decrease the average perplexity of the generated dataset by four times. However, since perplexity computation largely depends on the evaluation model, which is not explicitly pre-trained on the instruction domain, this process potentially filters out some valid but highly diversified instructions. We leave developing a better post-processing strategy for this process as future work.

2.4 Instruction Output Generation

After obtaining a predefined number of valid instructions with the previous steps, we generate the output by prompting an existing LLM. We iterate over all generated instructions, prompt the LM with the instruction, and collect the LLM's response. We filter out the instructions with an invalid response (e.g., the LLM's output contains no helpful information or deems the instruction as a not self-contained sequence). The remaining instruction-output pairs form our instruction-tuning dataset.

3 Instruction Data Analysis

We apply the method introduced in Section 2 to the concatenation of three open-source datasets' instruction sets: ShareGPT [1], Dolly [5] and Alpaca [13]. The combined dataset **ShareGPT-Alpaca-Dolly (S.A.D.)** contains 107442 instructions. We sample 1000 instructions from it with stratified sampling as the test set, while the remaining 106442 instructions are used for training DiffLM.

We sampled an instruction set with a DiffLM trained on the joint S.A.D.'s training set, using a BART-Large as its decoder. The output for each instruction was collected with gpt-3.5-turbo. The resulting dataset contains 52000 diverse instructions with high-quality outputs. We name our dataset DIFFIT (Diffusion-based Instruction Tuning dataset).

We compare our dataset DIFFIT with several open-sourced instruction tuning datasets in Table 1 in terms of instruction diversity. Among the similar-sized datasets, S.A.D. and our DIFFIT achieves the highest unique token counts. Although DIFFIT has lower unique token counts compare to S.A.D., it achieves a better n-gram diversity in terms of 4-gram Repetition and 4-gram SelfBLEU. This suggests that compared to DiffLM's training instruction set, the sampled instructions from DiffLM can cover more new complex concepts or phrases.

4 Instruction Tuning Experiments

We conduct instruction-tuning on a pre-trained LLM, LLaMA [10] with our sampled DIFFIT dataset. In this section, we compare a LLaMA 7B fine-tuned on DIFFIT with LLaMA 7B finetuned on similar-sized instruction-tuning datasets with the same training settings.

[1]We use an open-source version of ShareGPT from `https://hf.co/datasets/anon8231489123/ShareGPT_Vicuna_unfiltered`.

Model & Dataset	MMLU 0-shot	MMLU 5-shot	GSM Direct	GSM CoT	Codex-HumanEval Pass@1	Codex-HumanEval Pass@10	TydiQA GP	TydiQA CB	Avg.
LLaMA 7B [10]	31.9	35.2	6.0	9.0	11.6	18.3	39.1	9.5	20.1
+ Unnat. Inst. [8]	42.9	38.1	3.5	5.0	10.3	19.8	36.3	6.5	20.3
+ Self-Instruct [6]	35.7	33.2	4.0	6.5	6.2	12.1	35.4	8.7	17.7
+ Alpaca [13]	41.5	40.3	7.0	10.0	13.2	22.0	31.2	7.2	21.6
+ GPT4-Alpaca [7]	42.6	38.3	6.5	10.0	13.2	25.0	23.6	5.8	20.6
+ Code-Alpaca [14]	34.7	34.5	6.5	7.5	16.5	29.2	36.7	10.5	22.0
+ OASST1 [15]	32.9	29.7	6.0	6.5	10.1	20.4	26.8	7.8	17.5
+ S.A.D.	37.4	27.3	5.5	14.0	12.4	20.0	23.5	8.2	18.5
+ **DIFFIT**	39.7	33.6	7.5	14.5	12.0	25.6	23.8	6.7	20.4
+ S.A.D. (2x Training)	38.7	30.5	4.5	13.5	12.9	15.0	32.5	7.5	19.4
+ S.A.D.+**DIFFIT**	40.6	32.7	6.5	14.5	12.6	22.6	35.4	6.4	21.4

Table 2: Automatic evaluation of instruction-tuned LLM's general capabilities. S.A.D.: 52K instructions sampled from ShareGPT + Alpaca + Dolly. S.A.D. (2x Training): Training S.A.D. for double the total training steps to match the total training step with S.A.D. + DIFFIT. Baselines' results are from Wang et al. [16].

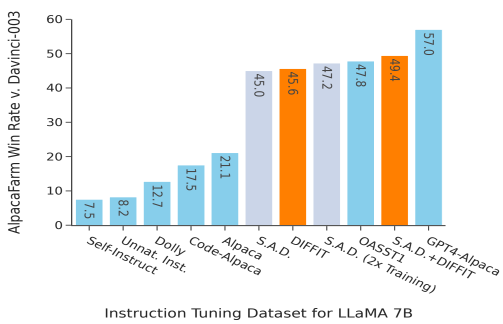

Figure 2: GPT-4 evaluation on model's win rate against Davinci-003 for instruction following.

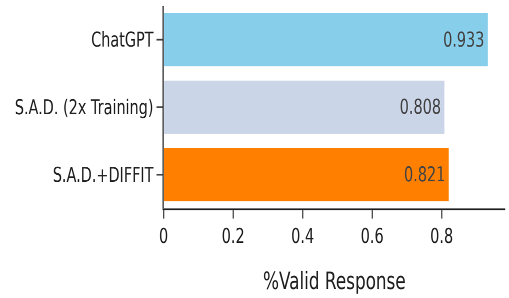

Figure 3: Human evaluation on the validity of model's response for user instruction following.

The results on a suite of automatic evaluations on LLM's general capability are shown in Table 2. Instruction tuning on DIFFIT achieves better performance than instruction-tuning on similarly-sized DiffLM training set S.A.D, and augmenting S.A.D. with the generated dataset can further improve LLM's general performance. This suggests that DiffTune can generate high-quality instructions that can augment existing diverse instruction-tuning datasets to increase LLM's general capabilities.

We show the automatic evaluation on AlpacaFarm in Figure 2. We find that although a mixture of 52K S.A.D. instructions is itself a diverse dataset,DIFFIT sampled by DiffTune further increases the response quality by achieving a higher win rate against text-davinci-003. S.A.D. augmented by DIFFIT achieves a 2.2% win rate increase compared to LLaMA trained on S.A.D. with the same training steps. This shows the effectiveness of DiffTune as both an instruction-tuning generation method and an instruction set augmentation approach.

Lastly, we conduct a human evaluation of the validity and helpfulness of LLM's response to real-world human instructions. In Figure 3, we illustrate the percentage of valid responses evaluated by human evaluators. The original 52K S.A.D. dataset achieves better response validity when augmented with DIFFIT, achieving an increase of 1.3 percentage points brought by our method.

5 Conclusion

We introduce DiffTune, a novel method for generating instruction-tuning datasets that overcome the limitations of current techniques. By leveraging the capabilities of Diffusion Language Models and revising the sampling strategy, DiffTune generates diverse, high-quality instruction-tuning datasets in a cost-effective manner. The superior performance of the LLaMA model, when finetuned with our DiffTune-generated dataset, emphasizes the efficacy of our approach. Both automatic and human evaluations underscore the quality and diversity of the data generated by DiffTune, showcasing its potential to optimize the performance of Large Language Models across varied tasks and domains.

References

[1] OpenAI. GPT-4 Technical Report. *ArXiv preprint*, 2023.

[2] Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Tachard Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Z. Chen, Eric Chu, J. Clark, Laurent El Shafey, Yanping Huang, Kathleen S. Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan A. Botha, James Bradbury, Siddhartha Brahma, Kevin Michael Brooks, Michele Catasta, Yongzhou Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, C Crépy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, M. C. D'iaz, Nan Du, Ethan Dyer, Vladimir Feinberg, Fan Feng, Vlad Fienber, Markus Freitag, Xavier García, Sebastian Gehrmann, Lucas González, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, An Ren Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wen Hao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Mu-Li Li, Wei Li, Yaguang Li, Jun Yu Li, Hyeontaek Lim, Han Lin, Zhong-Zhong Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alexandra Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Marie Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniela Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Ke Xu, Yu Xu, Lin Wu Xue, Pengcheng Yin, Jiahui Yu, Qiaoling Zhang, Steven Zheng, Ce Zheng, Wei Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. Palm 2 technical report. *ArXiv preprint*, 2023.

[3] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022.

[4] Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022.

[5] Databricks. Free Dolly: Introducing the World's First Open and Commercially Viable Instruction-Tuned LLM - The Databricks Blog, 2023.

[6] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-Instruct: Aligning Language Model with Self Generated Instructions. *ArXiv preprint*, 2022.

[7] Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction Tuning with GPT-4. *ArXiv preprint*, 2023.

[8] Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. Unnatural Instructions: Tuning Language Models with (Almost) No Human Labor. *ArXiv preprint*, 2022.

[9] Koustuv Sinha, Prasanna Parthasarathi, Joelle Pineau, and Adina Williams. UnNatural Language Inference. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021.

[10] 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. LLaMA: Open and Efficient Foundation Language Models. *ArXiv preprint*, 2023.

[11] Justin Lovelace, Varsha Kishore, Chao Wan, Eliot Shekhtman, and Kilian Weinberger. Latent Diffusion for Language Generation. *ArXiv preprint*, 2022.

[12] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. WizardLM: Empowering Large Language Models to Follow Complex Instructions. *ArXiv preprint*, 2023.

[13] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

[14] Sahil Chaudhary. Code alpaca: An instruction-following llama model for code generation. `https://github.com/sahil280114/codealpaca`, 2023.

[15] Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. OpenAssistant Conversations – Democratizing Large Language Model Alignment. *ArXiv preprint*, 2023.

[16] Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. How Far Can Camels Go? Exploring the State of Instruction Tuning on Open Resources. *ArXiv preprint*, 2023.

[17] Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022.

[18] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *ArXiv preprint*, 2022.

[19] Da Yin, Xiao Liu, Fan Yin, Ming Zhong, Hritik Bansal, Jiawei Han, and Kai-Wei Chang. Dynosaur: A Dynamic Growth Paradigm for Instruction-Tuning Data Curation. *ArXiv preprint*, 2023.

[20] Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022.

[21] Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: Less Is More for Alignment. *ArXiv preprint*, 2023.

[22] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL `https://lmsys.org/blog/2023-03-30-vicuna/`.

[23] Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. Baize: An Open-Source Chat Model with Parameter-Efficient Tuning on Self-Chat Data. *ArXiv preprint*, 2023.

[24] Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori Hashimoto. Diffusion-LM improves controllable text generation. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[25] Xiaochuang Han, Sachin Kumar, and Yulia Tsvetkov. SSD-LM: Semi-autoregressive Simplex-based Diffusion Language Model for Text Generation and Modular Control. *ArXiv preprint*, 2022.

[26] Zhengfu He, Tianxiang Sun, Kuanning Wang, Xuanjing Huang, and Xipeng Qiu. Diffusion-BERT: Improving Generative Masked Language Models with Diffusion Models. *ArXiv preprint*, 2020.

[27] Robin Strudel, Corentin Tallec, Florent Altché, Yilun Du, Yaroslav Ganin, Arthur Mensch, Will Grathwohl, Nikolay Savinov, Sander Dieleman, Laurent Sifre, and Rémi Leblond. Self-conditioned Embedding Diffusion for Text Generation. *ArXiv preprint*, 2022.

[28] Ishaan Gulrajani and Tatsunori B. Hashimoto. Likelihood-Based Diffusion Language Models. *ArXiv preprint*, 2023.

[29] Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. DiffuSeq: Sequence to Sequence Text Generation with Diffusion Models. *ArXiv preprint*, 2022.

[30] Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Fei Huang, and Songfang Huang. SeqDiffuSeq: Text Diffusion with Encoder-Decoder Transformers. *ArXiv preprint*, 2022.

[31] Zhujin Gao, Junliang Guo, Xu Tan, Yongxin Zhu, Fang Zhang, Jiang Bian, and Linli Xu. Difformer: Empowering Diffusion Models on the Embedding Space for Text Generation. *ArXiv preprint*, 2022.

[32] Jiasheng Ye, Zaixiang Zheng, Yu Bao, Lihua Qian, and Mingxuan Wang. DINOISER: Diffused Conditional Sequence Learning by Manipulating Noises. *ArXiv preprint*, 2023.

[33] Hao Zou, Zae Myung Kim, and Dongyeop Kang. A Survey of Diffusion Models in Natural Language Processing. *ArXiv preprint*, 2023.

[34] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[35] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, JMLR Workshop and Conference Proceedings, 2015.

[36] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

[37] Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. A Contrastive Framework for Neural Text Generation. *ArXiv preprint*, 2022.

[38] Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. In Kevyn Collins-Thompson, Qiaozhu Mei, Brian D. Davison, Yiqun Liu, and Emine Yilmaz, editors, *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, 2018.

[39] Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaïd Harchaoui. MAUVE: measuring the gap between neural text and human text using divergence frontiers. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin,

Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 2021.

[40] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling Instruction-Finetuned Language Models. *ArXiv preprint*, 2022.

[41] Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. AlpacaFarm: A Simulation Framework for Methods that Learn from Human Feedback. *ArXiv preprint*, 2023.

[42] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

[43] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems. *ArXiv preprint*, 2021.

[44] Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 2020.

[45] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating Large Language Models Trained on Code. *ArXiv preprint*, 2021.

[46] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

A Related Works

Instruction-Tuning Datasets Recently, instruction tuning on LLMs has been a hot research area in NLP [3, 4, 17, 18]. The dataset used for LLM instruction tuning has to be diverse to cover various tasks, scenarios, and input formats. To guarantee the diversity requirement of the dataset, previous literature generates their instruction data from a variety of existing NLP datasets [3, 4, 17, 19, 20], various online forums [21], human labelling [5], crowdsourcing [15, 22], and generation by existing (proprietary) LLMs [6, 13, 7, 8, 12, 23]. However, LLM-generated instruction tuning datasets frequently suffer from lower diversity and less authenticity than human-generated datasets, and human-labeled datasets have a high cost for dataset generation. Although crowd-sourcing-based instruction tuning datasets achieve better variety and can best reflect real-world user prompts, collecting a large set of user inputs is still costly. This paper introduces a cost-effective method of instruction-tuning dataset generation that can construct or augment an existing dataset with crowdsourcing-level quality and diversity with no extra human labeling.

Diffusion Model for Language Generation Different from the mainstream auto-regressive language models (ARLM) [1, 10] which generates texts token by token, diffusion language models (DiffLM) fall into the category of non-auto-regressive language models (NARLM), which generate all tokens in parallel. Diffusion LM was first introduced by Li et al. [24], where the authors trained a diffusion model in the token embedding space. DiffLM has been applied to contollable text generation [24, 25], unconditional text generation [26–28] and sequence-to-sequence tasks [29–32]. On the task of unconditional generation, compared to ARLM, DiffLM can achieve more robust and efficient text sequence generation [33] and higher generation diversity [11]. This paper adopts the DiffLM proposed by Lovelace et al. [11] for diverse and high-quality instruction generation.

B Backgrounds on Diffusion Language Models

Diffusion models [34, 35] aim to approximate a target distribution $p(\mathbf{x})$ by learning a reversible transition between it and a Gaussian distribution. The forward process takes a sample from the target distribution ($\mathbf{z_0} := \mathbf{x} \sim p(\mathbf{x})$), and sequentially adds noise to produce a Markov chain: $\{\mathbf{z_0}, \mathbf{z_1}, \dots, \mathbf{z_T}\}$, where $q(\mathbf{z_{t+1}}|\mathbf{z_t}) = \mathcal{N}(\sqrt{1-\beta_t}\mathbf{z_t}, \beta_t \boldsymbol{I})$ with some variances β_t. The inversion of the forward process is called denoising, where one samples from Gaussian distribution ($\mathbf{z_T} \sim \mathcal{N}(0, \boldsymbol{I})$) and sequentially produces a chain of less noisy samples $\{\mathbf{z_T}, \mathbf{z_{T-1}}, \dots, \mathbf{z_0}\}$, where the final element $\mathbf{z_0}$ is a sample from the target distribution $p(\mathbf{x})$. For that, one trains the denoising neural network $\hat{\mathbf{x}}_\theta = f_\theta(\mathbf{z_t}, t)$, which approximately recovers the original sample from target distribution, $\mathbf{x}$, from its noisy version $\mathbf{z_t}$. Specifically, for any $\mathbf{x} \sim p(\mathbf{x})$ and any time step t, one generates the noisy sample $q(\mathbf{z_t}|\mathbf{x})$ with a forward process, and recovers it so that $\hat{\mathbf{x}}_\theta \approx \mathbf{x}$. When the denoising network is trained, one can generate samples from the target distribution using the denoising Markov chain described above. Going from $\mathbf{z_t}$ to $\mathbf{z_{t-1}}$ requires sampling from the distribution $p(\mathbf{z_{t-1}}|\mathbf{z_t}) := \mathcal{N}(\mu_t(\mathbf{z_t}, \hat{\mathbf{x}}_\theta), \sigma_t^2 \boldsymbol{I})$, where $\mu_t(\mathbf{z_t}, \mathbf{x})$ has a closed form solution.

Applying diffusion models to NLP is not straightforward because of the discrete nature of language. We use the model suggested by Lovelace et al. [11], where they modeled the latent space of the encoder-decoder language model with diffusion. In particular, they used BART [36], because it was trained with the denoising objective on the latent representation. Hence, the approximate samples from the diffusion model would be meaningfully decoded. Instead of learning to generate a sample from the latent space, one needs to sample a sequence of vectors, which would be ideally decoded into a valid sentence. For that, given a length from the length distribution of the sequences in training set $l_i \sim \mathcal{L}(\{X_t\})$ and a sampled noise $\boldsymbol{Z}_T \in \mathbb{R}^{l_i \times d} \sim \mathcal{N}(0, \boldsymbol{I})$, the denoising $\boldsymbol{X}_T$ is a sequence in the target domain.

C Instruction Data Generation Analysis

We apply DiffTune to different existing instruction tuning datasets, different DiffLM decoder size and different sampling strategies. We analyze these three aspects of instruction generation settings one by one in the following subsections.

DiffLM Dec.	Sampling	4-gram Rep. ↓	4-gram SelfBLEU ↓	Diversity ↑	Mem. ↓	MAUVE ↑	Ppl. ↓
BART-Base	Std Gaussian	.037 $\pm 10^{-5}$	.239 $\pm 10^{-5}$	.565 $\pm 10^{-5}$	.159 $\pm 10^{-5}$	**.756** $\pm 10^{-4}$	70.3 $\pm$ 2.7
BART-Base	Student T	.031 $\pm 10^{-5}$	.226 $\pm 10^{-4}$	.577 $\pm 10^{-4}$	.151 $\pm 10^{-4}$	.739 $\pm 10^{-4}$	70.9 $\pm$ 0.3
BART-Base	30% Higher Var.	.023 $\pm 10^{-6}$	.201 $\pm 10^{-5}$	.578 $\pm 10^{-6}$	.128 $\pm 10^{-6}$	.709 $\pm$ .003	77.1 $\pm$ 0.1
BART-Base	100% Higher Var.	**.022** $\pm 10^{-5}$	.200 $\pm 10^{-5}$	.568 $\pm 10^{-5}$	**.116** $\pm 10^{-5}$	.637 $\pm$ .001	80.6 $\pm$ 0.1
BART-Large	Std Gaussian	.037 $\pm 10^{-5}$	.228 $\pm 10^{-5}$	.565 $\pm 10^{-4}$	.163 $\pm 10^{-6}$	.737 $\pm 10^{-4}$	**64.3** $\pm$ **0.1**
BART-Large	Student T	.031 $\pm 10^{-5}$	.217 $\pm 10^{-5}$	.573 $\pm 10^{-5}$	.154 $\pm 10^{-5}$	.707 $\pm$ .001	65.5 $\pm$ 1.1
BART-Large	30% Higher Var.	.025 $\pm 10^{-6}$	.190 $\pm 10^{-4}$	.583 $\pm 10^{-5}$	.124 $\pm 10^{-6}$	.660 $\pm$ .006	73.9 $\pm$ 0.3
BART-Large	100% Higher Var.	.023 $\pm 10^{-6}$	**.187** $\pm 10^{-5}$	**.584** $\pm 10^{-6}$	.119 $\pm 10^{-5}$	.625 $\pm$ .002	74.9 $\pm$ 0.7
S.A.D. Test Set		.087	.239	.550	.406	-	158.7

Table 3: The comparison of sample diversity and quality among different DiffLM decoder selection and DiffLM sampling strategies based on 1000 sampled instructions. ↑: Higher is better. ↓: Lower is better. The setting with the highest performance for each metric is labeled in **bold**, while the second-highest is labeled in underline.

C.1 Experiment Details

To maximize the diversity of the sampled new instructions from DiffLM, we train our DiffLM on the concatenation of the following open-source datasets' instruction set: 1) **ShareGPT.** It is collected from publically-shared real-world user dialogues with ChatGPT. We filtered the dataset only to contain first-round user inputs with a valid response from ChatGPT. The filtered dataset contains 40428 instructions. 2) **Dolly.** It contains 15011 instructions created by Databricks employees. 3) **Stanford Alpaca.** It is generated with an modified self-instruct strategy [6] using text-davinci-003, containing 52002 instructions.

The combined dataset **ShareGPT-Alpaca-Dolly (S.A.D.)** contains 107442 instructions. We sample 1000 instructions from it with stratified sampling as the test set, while the remaining 106442 instructions are used for training our DiffLM.

In this section, we train two DiffLMs with different DiffLM decoder sizes: BART-Base or BART-Large. After training, we test four different sampling strategies:

1. **Standard Gaussian.** We sample from the DiffLM with $\boldsymbol{Z}_T \in \mathbb{R}^{l_i \times d} \sim \mathcal{N}(0, \boldsymbol{I})$.
2. **Student T.** We sample from the DiffLM with $\boldsymbol{Z}_T \in \mathbb{R}^{l_i \times d} \sim t_2$. The noise distribution puts a higher probability on the long tail compared to the original strategy.
3. **30% Higher Variance.** We apply the sampling strategy introduced in DIFFTUNE.
4. **100% Higher Variance.** We sample from a Gaussian with a higher variance for all tokens: $\boldsymbol{Z}_T \in \mathbb{R}^{l_i \times d} \sim \mathcal{N}(0, 10\boldsymbol{I})$.

We sample 1000 instructions from each configuration with beam search. The sampled instruction set will be evaluated with the following metrics:

1. **Repetition** [37] measures generation diversity by the proportion of repetitive n-grams: $rep_n = (1 - \frac{\text{unique}n-\text{grams}(\hat{x})}{\text{total}n-\text{grams}(\hat{x})})$.
2. n**-gram Diversity** [37] measures generation diversity by considering different n-gram repetitions: $div = \prod_{n=2}^{4} \left(1 - \frac{rep_n}{100}\right)$.
3. **SelfBLEU** [38] measures generation diversity by computing the average of each generated instance's BLEU score against all others.
4. **Memorization** [11] measures generation diversity by computing the proportion of 4-grams from the generated sequences that exist in the training set.
5. **MAUVE** [39] measures generation quality by considering the token distribution closeness between the generated and reference sets. We compute the MAUVE score against S.A.D.'s test set.
6. **Perplexity (ppl)** measures generation quality by how likely a language model can generate the sequence. We compute ppl with GPT2-Large.

C.2 Comparisons of Training Settings

We sample 1000 instructions from each setting with an applied perplexity threshold of 150, which aligns with our final instruction generation process. We compare the diversity and quality of the generated instruction set with S.A.D.'s test set. The results are shown in Table 3.

DiffLM's decoder model size. When the instructions are sampled from standard Gaussian, using a smaller BART-Base as DiffLM's decoder achieves slightly on-par or better diversity and quality across all metrics except for perplexity, which aligns with the observation from Lovelace et al. [11]. However, when using a different sampling strategy, using a larger decoder illustrates a different trend.

For generation quality, when using the same sampling strategy, BART-Large settings always achieve a lower perplexity compared to its BART-Base counterpart. The higher generation quality from a BART-Large decoder is also observed during our case studies. For generation diversity, when using a sampling distribution distant from a standard Gaussian, using BART-Large generally achieves a higher diversity gain.

DiffLM's sampling strategy. Although all tested settings achieve a higher diversity across all metrics compared to S.A.D.'s test set, we found that using a noise distribution other than standard Gaussian always achieves a more diversified instruction generation. Although sometimes the generated instructions include grammatical errors or unknown concepts, they can be denoised in the output generation process by larger LLMs and can better resemble real-world user inputs, where the prompts are not always perfect.

We provide a simple case study of using different percentages of high-variance noises in Table 4. We begin with a noise matrix $\boldsymbol{Z}_T \sim \mathcal{N}(0, \boldsymbol{I})$, and gradually substitute a specific percentage of its column vectors with sampled vectors from $\mathcal{N}(0, \boldsymbol{I})$ and observe the decoded output. We keep sampling until the generation achieves a perplexity below 150, which takes around 2 rounds of generation.

We observed that the generated sequences are all in the format of instructions, while a higher percentage of vectors sampled with higher variance is more likely to introduce grammatical errors (e.g., "i am") or unknown concepts (e.g., "Iafkenhoek"). This phenomenon resembles real-world user inputs since similar grammatical errors and typos are common in ShareGPT's instruction set. In our dataset generation process, we adopt the setting of using a BART-Large decoder and a 30% Higher Var sampling strategy.

% Tokens Sampled With Higher Var.	**DiffLM Generation**
0%	Could you list few 10 most important things to prepare for the entrance examination? Think about the factors in order to determine your aptitude. Please write in English language.
10%	I am an employee at a large endo wraping company in Hengshui, China. Give me some suggestions for a new cover letter and resume. I would love to have a good job description.
30%	Make me a list of things for the course i should do and be prepared. I'm doing a user design course, i am preparing for a class, but I don't know what to do about it.
50%	Write a poem in the style of Iafkenhoek explaining how humans will overcome a number of factors of mental and emotional goals, which may or may not be attainable. Create a short film about your dreams for humanity.

Table 4: A sample from DiffLM trained on S.A.D.'s instruction set with a BART-Large decoder. We begin with a noise matrix $\boldsymbol{Z}_T \sim \mathcal{N}(0, \boldsymbol{I})$, and gradually substitute column vectors of $\boldsymbol{Z}_T$ with vectors sampled from $\mathcal{N}(0, 10\boldsymbol{I})$ and observe the corresponding changes for its decoded generation.

Statistics	
# of data	52000
# of unique input tokens	49322
# of unique output tokens	74079
Avg. input length (in words)	24.4
Avg. output length (in words)	80.7

Table 5: Basic statistics of the DIFFIT dataset.

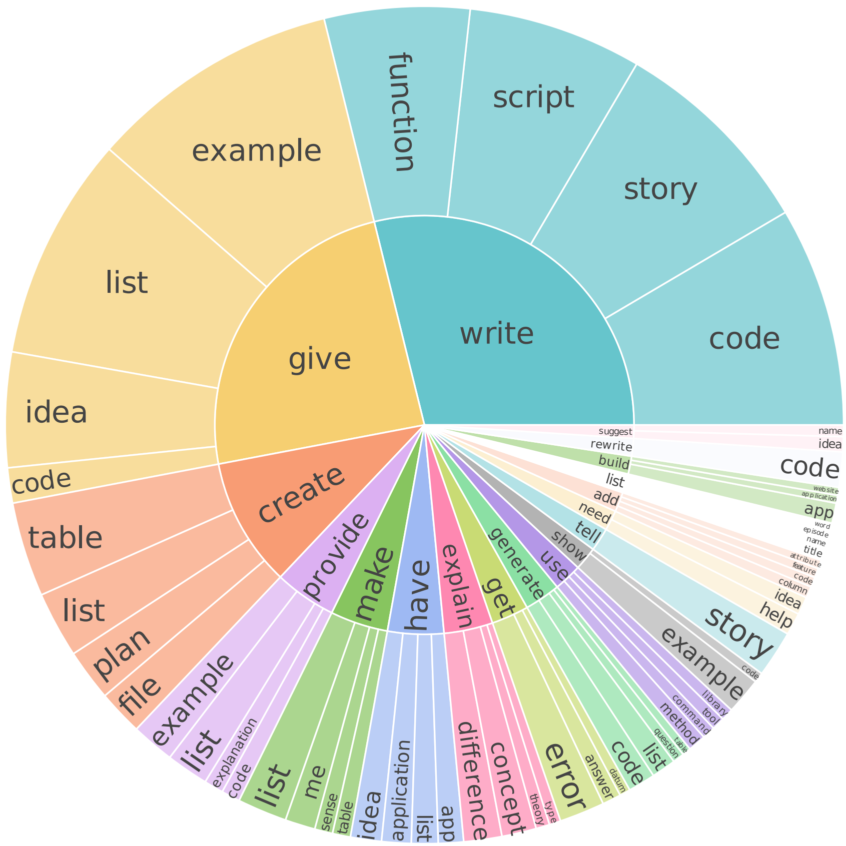

Figure 4: The most common root verb-noun combinations in DIFFIT's instruction set. The inner circle illustrates the root verbs, while the outer circle illustrates the corresponding direct nouns.

D Statistics of DIFFIT

D.1 Basic Statistics

We include the basic statistics of DIFFIT in Table 5.

D.2 Verb-Noun Analysis

Following previous practices of instruction diversity analysis [6, 7], we analyze the most common verb-noun combinations in the sampled instructions. We extract the root verb and their corresponding direct-object noun of each instruction and plot the verb-noun combinations with a frequency higher than 10 in Figure 4. We observe a large variety in the verbs used in the dataset, with the instructions covering different generation types, including story, code, table, list, etc. It is also worth noticing that the verb "use" appears frequently in the instructions, which is usually intended to add specifications to the task scenario (e.g., use a specific library or use a particular tool), which resemble typical real-world user inputs to LLMs.

Dataset	Size	Cost
Self-Instruct	82K	$600
Unnatural Instructions	68K	$1370
Alpaca	52K	$500
GPT4-Alpaca	52K	$456
DIFFIT	52K	$27.8

Table 6: Dataset construction cost of several existing open-source instruction-tuning datasets with similar sizes.

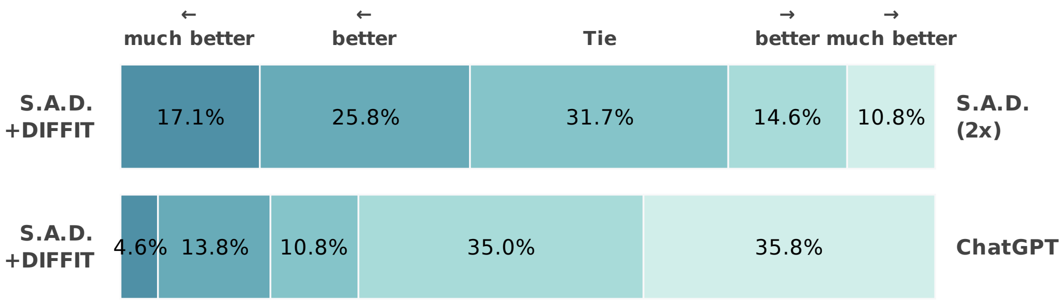

Figure 5: Human preference for two pairs of models.

D.3 Data Generation Cost

We compare DIFFIT's total construction cost with that of several open-source instruction-tuning datasets in Table 6.

A DiffLM with a BART-large decoder is easy to train and implement on consumer-level servers or computers. Compared to self-instruct [6], our method substitutes the LLM-based instruction generation step with DiffLM sampling, reducing the dataset construction cost while enhancing the dataset's diversity in different aspects. Thanks to the low cost of gpt-3.5-turbo's API call, we further reduce the cost of output generation.

E Human Preference Evaluation

We show human preference results in Figure 5. LLaMA 7B instruction-tuned on dataset augmented by DiffTune is favored 42.9% of the times when compared to the counterpart model tuned only on S.A.D, which is only favored 25.4% of the time. Although LLaMA 7B instruction-tuned on S.A.D.+DIFFIT is still far from comparable with ChatGPT, our generated responses are still favored 18.4% of the time, despite the large discrepancy of model size and training cost.

F Instruction Tuning Experiment Details

Our evaluation closely follows previous general-purpose LLM's settings [40, 6, 22, 41, 16], where the evaluation covers different aspects of LLM's general ability as well as instruction-following capability.

Evaluations on LLM's general capability. We compare the same LLM finetuned on different instruction tuning datasets on the following benchmarks: 1) **MMLU** [42] for factual knowledge evaluation, which contains multiple-choice questions from 57 subjects covering different difficulties. 2) **GSM** [43] for mathematical reasoning, which contains grade school-level math problems. 3) **TyDiQA** [44] for multilingual evaluation, which contains machine reading comprehension or question-answering tasks in 11 typologically diverse languages. 4) **Codex-HumanEval** [45] for coding evaluation, which requires the model to generate code given a docstring.

For all benchmarks, we follow the evaluation setting of [16], except that we use Alpaca's dialogue template instead of Tülu's.

Hyperparameter	**Value**
#Trainable Params	214M
Max Seq Length	64
Diffusion Steps	1000
Noise Schedule	Linear
Regression Loss	L1
DiffLM Transformer Layers	12
DiffLM Transformer Dim.	768
Optimizer	AdamW [2019]
Learniing Rate	1e-4
Batch Size	64
Warmup Steps	1000
Learning Rate Schedule	Cosine Decay
Weight Decay	1e-6
Dropout	0.0
Gradient Clipping	1.0
EMA Decay	0.9999
Iterations	300K

Table 7: Hyperparameter settings for training and sampling DiffLM.

Evaluations on LLM's instruction-following capability. We evaluate LLM's instruction-following capability by comparing different model's outputs to real-world user inputs. Both automatic and human evaluations are conducted to evaluate the helpfulness and validity of LLM's response: 1) **AlpacaFarm** [41] for automatic evaluation on instruction-following capabilities, which uses GPT-4 to compare an LLM's generation with text-davinci-003's generation on 805 instructions. 2) **VicunaEval** [22] for human evaluation on instruction-following capabilities, which contains 80 instructions covering a wide range of scenarios.

Human Evaluation. Human evaluation on VicunaEval covers two aspects: 1) Answer validity, where we ask evaluators to decide whether an LLM's response is acceptable; and 2) Pairwise comparison, where we ask evaluators to compare two LLM's response to the same instruction, then score their preference based on a 5-level scale to decide if one response is much better, slightly better or on-par compare to the other.

The instruction for human evaluation comes from Vicuna's evaluation set [22], which contains 80 instructions covering multiple daily scenarios. We generate responses from LLaMA 7B + S.A.D. (2x), LLaMA 7B + S.A.D. + DIFFITand ChatGPT for each instruction, and ask human evaluators to 1) Evaluate the validity of the model's response, and 2) Select the preferred response from two different model's generations. We conduct two pairs of preference evaluations: LLaMA 7B + S.A.D. + DIFFITv. LLaMA 7B + S.A.D. (2x), and LLaMA 7B + S.A.D. + DIFFITv. ChatGPT.

The evaluation is conducted with Google Forms. We split each pair of comparisons into two forms, with 40 instructions in each form. In each form, the ordering of the questions and the responses are totally randomized. We recruited 12 experienced annotators who are fluent in English and hold bachelor's degrees or above. Each form will be assigned to three different evaluators.

We show an example of our evaluation form in Figure 6.

G Experiment Details for DiffLM Training

For all DiffLM experiments, we train the DiffLM on S.A.D.'s training set using the hyperparameter settings shown in Table 7. The generation configuration for the BART decoder is shown in Table 8. Each setting is trained on a single NVIDIA V100 32G.

Setting	Value
max_length	64
min_length	8
num_beams	10
do_sample	False
no_repeat_ngram_size	2
repetition_penalty	4.0

Table 8: Generation configuration for DiffLM's BART decoder.

H Examples from the DIFFIT Dataset

We show some of the instances from the DIFFIT Dataset in Table 9. Please note that the table includes examples that collect hallucinating results from ChatGPT, although the answer's format is aligned with our instruction-tuning objective.

Instruction	Output
Tell me about the advantages and disadvantages of different programming languages. Add a bulletbind summary explaining the pros and cons of each language.Give me a brief explanation of the difference between Swift and Objective-C. Show me how to use C# instead of C++. Tell me why.	Different programming languages have unique advantages and disadvantages. For example, C++ is fast and flexible, but difficult to learn. Java is platform-independent, but slower than some languages. Python has simple syntax, but is slower compared to other languages. Swift is faster than Objective-C, but still somewhat new and has limited library support. C# is a cleaner version of C++ with better memory management, but it is not as fast as C++. To use C# instead of C++, you would need to download and install Microsoft's Visual Studio, a software development environment. C# is often used for web, Windows, and game development.
act as a middle school teacher write a script for a sci-fi/fantasy film set in the future make an edgy movie series	Title: "Galactic Rift" In a distant future, humanity has mastered space travel and colonization across multiple galaxies. But as the population soars, a powerful cult arises, vowing to cleanse the universe of all non-believers and establish a new order. Our protagonist, a small group of renegade fighters led by a young woman named Nova, must travel to the cult's stronghold and stop their plans before it's too late. Along the way, they will face alien creatures, rogue robots, and the cult's elite army. As they progress, they discover that the cult is not what they seem - it's led by ancient beings who seek to resurrect their god and conquer all life in the universe. Nova and her group must decide whether to continue their mission or join forces with the unlikely allies they meet along the way. Through action-packed battles, moral quandaries, and unexpected twists, "Galactic Rift" explores themes of power, faith, and the true meaning of humanity. This edgy sci-fi/fantasy series is perfect for middle schoolers who want an adventure that will leave them on the edge of their seats.
Assume you are an AI concept teacher, and you can use it in generating business. How to explain the concept as well as how to connect it with value working with workplace?How to talk about your experience of using AI concepts:	As an AI concept teacher, I explain that AI refers to the simulation of human intelligence in machines to perform tasks that typically require human cognitive abilities. AI can help businesses streamline processes, make data-driven decisions, and improve customer experiences. AI-powered tools can be integrated into a workplace to enhance productivity, automate repetitive tasks, and improve the accuracy of decision-making. As an AI concept user, I have successfully leveraged AI to automate data analysis and improve customer segmentation for a retail client, resulting in a 20% increase in sales.
How to prepare a job resume for a Chinese independent IT consultant and an IT contractor who works for the USA as a CPA consultant in China.	To prepare a job resume for a Chinese independent IT consultant or IT contractor working for the USA as a CPA consultant in China, highlight relevant certifications, degrees, and experience in the technology and finance industry. Emphasize strong communication skills, understanding of Western business practices, and proficiency in Chinese and English. Showcase prior project success, specific technical skills, and collaboration abilities. As an IT consultant or contractor, showcase relevant project management experience and expertise in the latest technologies that support business solutions. Additionally, mention how you have developed and scaled businesses in both the US and China markets. Be brief and clear about your most significant accomplishments, abilities, and contributions to companies you have worked with.
You are an expert Instructor. Please show me the steps step by step to generate required code to learn Kuolang. You respond with me as if you are the world that know everything about Python and how to use it. Then give me a tutorial on how I can learn and learn everything. Give me	Sure, to learn Kuolang, you need to start by downloading an IDE that supports the language, such as VS Code. Then, you can learn the basics of the language, such as its syntax and data types. There are many resources available, including online courses, documentation, and tutorials. Once you have a basic understanding of the language, practice coding exercises to reinforce your skills. Finally, when you feel confident, try building a project from scratch to solidify your knowledge. Good luck!

Table 9: Samples from the generated DIFFIT dataset.

Question 2: What if the Beatles had never formed as a band?

Answer A:
If the Beatles had never formed as a band, the music scene in the 1960s and beyond would have been drastically different. It's likely that other bands and artists would have risen to take their place, but the impact and influence of the Beatles on popular music cannot be overstated. Their innovative sound, thoughtful lyrics, and charismatic personalities helped to shape the direction of music at the time. The Beatles also served as a cultural icon, influencing fashion, art, and societal attitudes. Without the Beatles, the music industry and youth culture may have taken a different direction.

Answer B:
The world would have been different without the Beatles. They were one of the most influential bands in music history and their impact can still be felt today. Their iconic hits like "Yesterday" and "Hey Jude" have become timeless classics, and their influence on pop culture extends far beyond music. Without the Beatles, there may not have been a British Invasion and other similar musical movements that followed. The world may have missed out on some of the most memorable songs and performances of all time. It's hard to imagine what our world would look like without the Beatles, but we can certainly appreciate them for the incredible legacy they left behind.

Is Answer A a valid response? A valid response should: (1) Answer the question; (2) Have no significant errors; (3) Have no meaningless text (e.g., repetition). *

- Yes
- No

Is Answer B a valid response? A valid response should: (1) Answer the question; (2) Have no significant errors; (3) Have no meaningless text (e.g., repetition). *

- Yes
- No

Out of the two answers, which one do you prefer as a better answer to the question? *

	1	2	3	4	5	
A is clearly better	○	○	○	○	○	B is clearly better

Figure 6: The form used for human evaluation.