Continuous Language Model Interpolation for Dynamic and Controllable Text Generation

Sara Kangaslahti
Harvard University
sarakangaslahti@g.harvard.edu

David Alvarez-Melis
Harvard University
dam@seas.harvard.edu

Abstract

As large language models (LLMs) have gained popularity for a variety of use cases, making them adaptable and controllable has become increasingly important, especially for user-facing applications. While the existing literature on LLM adaptation primarily focuses on finding methods that optimize over a fixed set of attribute classes, here we focus on the challenging continuous case where the model must dynamically adapt to diverse —and often changing— user preferences within predefined attribute ranges. For this, we leverage adaptation methods based on linear weight interpolation, casting them as continuous multi-domain interpolators that produce models with specific prescribed generation characteristics on-the-fly. Specifically, we use low-rank updates to fine-tune a base model to various different domains, yielding a set of anchor models with distinct generation profiles. Then, we use the weight updates of these anchor models to parametrize the entire (infinite) class of models contained within their convex hull. We empirically show that varying the interpolation weights yields predictable and consistent change in the model outputs with respect to all of the controlled attributes. We find that there is little entanglement between most attributes. Our results suggest that linearly interpolating between the weights of fine-tuned models facilitates predictable, fine-grained control of model outputs with respect to multiple stylistic characteristics simultaneously.[1]

1 Introduction

Large language models (LLMs) are used for a diverse set of applications due to their high performance across a wide spectrum of tasks [Bubeck et al., 2023]. In many common LLM use cases (such as chatbots), different users often have distinct and continuously evolving preferences for the type of output they want. For example, a user might want a creative and verbose response for certain queries, but a concise and precise response for others. In practice, a user may try different variations of the same query successively until they elicit a desired generation. This trial-and-error process can be time-consuming and lacks guaranteed results, especially since minor word changes in a prompt can have disproportionate impact on the output. Additionally, expressing fine-grained continuous preferences (e.g., simplifying a response by 25%) is often difficult in —inherently discrete— natural language. These challenges are exacerbated when the user has complex, multi-faceted preferences (e.g., a specific combination of simplicity, formality, and verbosity) that they expect the generation to satisfy all at once. As a result, there is a pressing need for methods that allow for fine-grained and predictable control over LLM text generation, and which can adapt on-the-fly to mutable user preferences and constraints.

Prior work in controllable text generation (CTG) has largely focused on optimizing for one set of control criteria through techniques such as instruction tuning [Zhou et al., 2023], modifying the output

[1] Code: https://github.com/skangasl/continuous-lm-interpolation

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

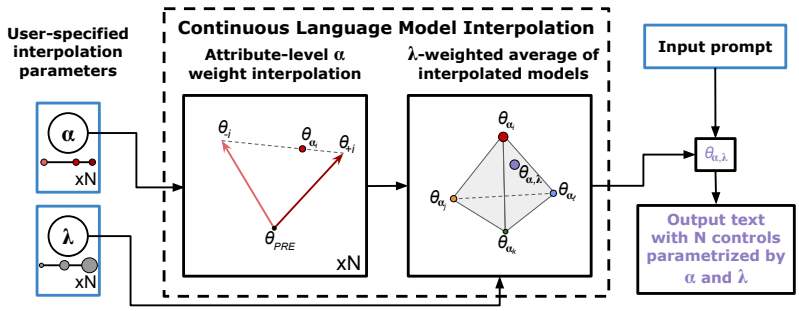


Figure 1: **Overview of our continuous model interpolation framework.** Given a collection of 'anchor' models fine-tuned on datasets at opposite ends of an attribute spectrum (e.g., θ_{+i}: positive and θ_{-i}: negative sentiment) for N different attributes, the user selects interpolation parameters α (per-attribute spectrum modulation) and λ (attribute mixture weights), which are used to generate a model with weights $\theta_{\alpha,\lambda}$ tailored to that specific parameter choice. This framework allows for *any* such interpolating model to be created on-the-fly and without additional fine-tuning, providing efficient, dynamic, and fine-grained generation control.

probability distributions [Pascual et al., 2021, Yang and Klein, 2021, Dekoninck et al., 2024], changing model activations at inference time [Li et al., 2023], learning modifications to the embeddings [Li and Liang, 2021, Han et al., 2023], or training [Keskar et al., 2019, Krause et al., 2021]. The vast majority of these methods, however, are not parametrized continuously and instead require a fixed set of controls criteria. Thus, to achieve fine-grained control in the range between different attribute classes, they would have to be individually applied to each specific set of intermediate attribute values, which is prohibitively expensive over a continuous range. Similarly, while fine-tuning models with data that contains a proportionate amount of documents from each desired objective (ie 0.5 positive and 0.5 negative sentiment documents for a neutral model) would likely allow for the most precise optimization, this is computationally infeasible to do for each combination of control variables and strengths of control in the entire (infinite) set of possible combinations.

With these challenges in mind, here we seek to enable dynamic and controllable text generation in a manner that takes advantage of the strengths of fine-tuning while remaining computationally feasible for dynamically changing control variables. Recent work has demonstrated that multiple pre-trained or fine-tuned models can be effectively composed through linear weight interpolation [Wortsman et al., 2022, Ilharco et al., 2023]. This has also been shown to extend to models trained with parameter-efficient fine-tuning (PEFT) methods [Zhang et al., 2023, Huang et al., 2024] such as low-rank adaptation [Hu et al., 2021]. We build upon and extend this line of work by showing that linear weight interpolation can be used to obtain models with specific mixtures of characteristics on-the-fly and without additional training, effectively providing a continuous parametrization of the (infinite) 'convex hull' of a set of fine-tuned models. To do so, we fine-tune two endpoint anchor models for each control attribute, one at each extreme of attribute strength. We then interpolate along the vector between the weights of these two models for each attribute before computing a weighted average across all of the single-attribute interpolated models. Thus, varying the interpolation and averaging weights gives us *dense coverage* of the model parameter space, allowing us to create models tailored to any preference profile spanned by the fine-tuned models. We evaluate linear weight interpolation for multiple style attributes and demonstrate empirically that changes in the interpolation and averaging weights yield predictable and consistent responses in the level of each attribute in the generations.

A potential pitfall of this approach is that, as seen in prior work in the vision domain [Ortiz-Jimenez et al., 2023], the weights for different single-attribute interpolated models may be entangled. This could lead to unexpected correlations between attributes in the averaged models. These correlations are detrimental to CTG, as changing the interpolation weights for one attribute could have an unexpected effect on the correlated attributes in the output text. However, we find that there is surprisingly little entanglement between the vast majority of control attributes.

In summary, our key contributions are: (1) we show how parameter-efficient adaptation methods can be used to continuously interpolate between models fine-tuned with various distinct generation objectives, allowing for on-the-fly adaptation to user-specified generation preferences expressed in

terms of interpretable control variables; and (2) we demonstrate that changes in the interpolation yield smooth and predictable changes in the properties of the generated text across multiple sets of controls with limited entanglement.

2 Fine-tuning and Weight Interpolation

We evaluate the ability of weight interpolation to control the outputs of LLMs on five commonly used style attributes defined in prior style transfer literature [Jin et al., 2022]: simplicity, formality, politeness, sentiment, and humor. For every style characteristic, we first fine-tune two endpoint 'anchor' models, each of which optimizes for one extreme of the style attribute. We then use these models as the basis of the interpolation scheme.

2.1 Datasets

For each style attribute, we fine-tune a separate anchor Llama2-7b model [Touvron et al., 2023] on two English datasets representing the extremes of the attribute level. For simplicity, we use the TinyStories dataset [Eldan and Li, 2023] to fine-tune a simple model and novel chapters from the BookSum dataset [Kryscinski et al., 2021] to fine-tune a complex model. We use the documents classified as formal and informal in Grammarly's Yahoo Answers Formality Corpus (GYAFC) dataset [Rao and Tetreault, 2018] to fine-tune formal and informal models. For the politeness attribute, we use the documents in the highest and lowest politeness class in the work by Madaan et al. [2020] for fine-tuning polite and impolite models, respectively. We fine-tune positive and negative sentiment models using the Stanford Sentiment Treebank (SST-2) dataset [Socher et al., 2013]. For humor, we use the FlickrStyle dataset [Gan et al., 2017] to fine-tune humorous and non-humorous models.

2.2 Fine-tuning

We fine-tune our models in a parameter-efficient manner using Low-Rank Adaptation [LoRA, Hu et al., 2021], which keeps pretrained model weights frozed but learns an additive low-rank matrix update for each layer during fine-tuning. Denoting the pretrained language model weights as $\theta_{PRE} \in \mathbb{R}^{d_1 \times d_1}$, LoRA computes the updated weights as:

$$\theta = \theta_{PRE} + BA \tag{1}$$

Here, $A \in \mathbb{R}^{k \times d_2}$ and $B \in \mathbb{R}^{d_1 \times k}$ (with $k \ll d_1, d_2$) are trainable parameters learned during fine-tuning. We use LoRA as an adaptation method because it requires significantly fewer parameters than traditional fine-tuning while maintaining similar performance, so LoRA weights can be quickly modified and applied to large pretrained language models. We use the parameters in Appendix A.4 for fine-tuning the models and fine-tune two LoRA models per style characteristic, one on each of the extreme classes outlined in 2.1. We denote the two LoRA fine-tuned endpoint anchor models for attribute i by $\theta_{+i} = \theta_{PRE} + B_{+i}A_{+1}$ and $\theta_{-i} = \theta_{PRE} + B_{-i}A_{-1}$.

2.3 Linear weight interpolation

Given a collection of fine-tuned model weights obtained by LoRA as described above, we generate interpolated models by linearly interpolating betweeen their weights. We formulate linear weight interpolation between the LoRA fine-tuned models in terms of interpolation weights α_i and attribute mixing weights λ_i as shown in Figure 1. For a single attribute, we interpolate along the vector between the two fine-tuned endpoint models by computing

$$\begin{aligned} \theta_{\alpha_i} &= \theta_{PRE} + \alpha_i \theta_{+i} + (1 - \alpha_i)\theta_{-i} \\ &= \theta_{PRE} + \alpha_i B_{+i} A_{+i} + (1 - \alpha_i) B_{-i} A_{-i} \end{aligned} \tag{2}$$

We call α_i the interpolation weight for the ith attribute. We note that $\alpha_i = 0$ and $\alpha_i = 1$ correspond to letting the interpolated model equal the fine-tuned models $\theta_{\alpha_i} = \theta_{-i}$ and $\theta_{\alpha_i} = \theta_{+i}$, respectively. Using Equation 2, we then combine multiple interpolated models θ_{α_i} by taking their weighted sum:

$$\theta_{\alpha,\lambda} = \sum_i \lambda_i \theta_{\alpha_i} \tag{3}$$

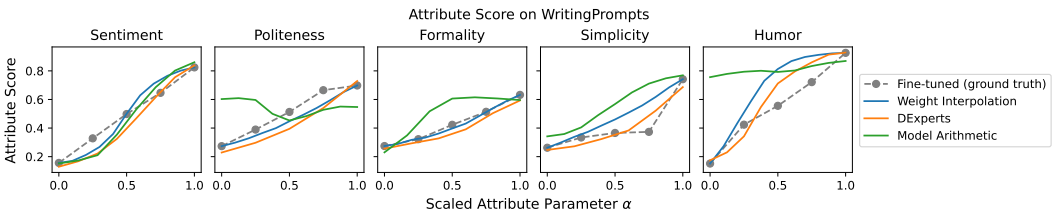


Figure 2: **Interpolated models recover custom fine-tuned models across the interpolation range**. We show the attribute scores for our model with weight α compared to DExperts [Liu et al., 2021] and model arithmetic [Dekoninck et al., 2024] with α scaled such that the scaled $\alpha = 0$ and $\alpha = 1$ models have the same score as the fine-tuned endpoint models. Our approach most closely follows the trend of the ground truth fine-tuned models.

We denote λ_i to be the mixing weight for the ith attribute and constrain $\sum_i \lambda_i = 1$. We note that the case with one attribute dimension corresponds to the sum having a single term with $\lambda_1 = 1$. With this formulation, we can construct any model in the convex hull of the fine-tuned models by choosing appropriate interpolation weights α and mixing weights λ. While the raw interpolation parameters do not have a clear meaning, we seek to show that a user can controllably increase or decrease the level of each attribute by modifying α and λ.

2.4 Evaluation

To evaluate the interpolated models, we use a subset of 1k randomly sampled prompts from the WritingPrompts dataset [Fan et al., 2018] and generate 3 continuations for each prompt. Similarly to prior work on text style transfer [Xu et al., 2018], we fine-tune a RoBERTa [Liu et al., 2019] classification head on each attribute using a held out split of the datasets in 2.1 and compute a sigmoid over the output logits to obtain the probability of class 1, which we report as the attribute score. We label the documents such that an attribute score closer to 1 corresponds to a document that is more simple, formal, polite, positive in sentiment, or humorous. We also compute perplexity on the test split of the WikiText dataset [Merity et al., 2016] and n-gram diversity of the WritingPrompts generations to evaluate fluency in the Appendix.

3 Continuous Language Model Interpolation

We begin by investigating the linear interpolations between each pair of fine-tuned anchor models (3.1). We then extend this analysis to the convex hull of anchor models for multiple attributes (3.2).

3.1 Linear interpolation for a single attribute dimension

We first explore the effect of moving along the vector between a single pair of fine-tuned anchor models. We compare our weight interpolation method to DExperts Liu et al. [2021] and model arithmetic Dekoninck et al. [2024], as these are the main approaches that allow for continuous control over the attribute strength parameter. We use the fine-tuned anchor models for DExperts and prompt-condition Llama2-7b-chat (see A.5 for details) for model arithmetic. We note that while our approach requires a single inference pass, both of these approaches require $2 * \text{number dimensions} + 1$ inference passes, and DExperts also requires the same number of fine-tuned models as our method. For both comparisons, we compute the attribute scores for $\alpha \in [-2, 2]$. To provide a direct comparison to the fine-tuned ground truth models, we scale the α parameter such that the scaled $\alpha = 0$ and $\alpha = 1$ correspond to the models with attribute score equal to that of the fine-tuned endpoint models. We evaluate these interpolation methods by their proximity to the ground truth interpolated models, which are models fine-tuned with α fraction data from class 1 and $1 - \alpha$ fraction data from class 0.

As shown in Figure 3, we find that our approach has a smooth and predictable increase in the attribute score for all of the control dimensions. Furthermore, it consistently is significantly closer to the fine-tuned ground truth models than model arithmetic. Model arithmetic also has very poor controllability for the politeness and humor dimensions, as the attribute score is unpredictable. While DExperts has similar performance as our approach, we note that since it requires $2 *$ number dimensions

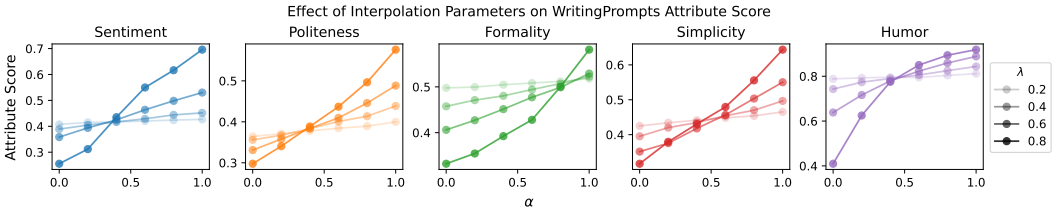


Figure 3: **Effect of α_i and λ_i on 5-dimensional interpolation**. For each attribute, we show the attribute scores for models with the given α_i and λ_i parameters, with all four other $\alpha_j = 1$ and $\lambda_j = (1 - \lambda_i)/4$. We find that increasing α_i consistently increases the attribute score and increasing λ_i consistently increases the effect of α_i.

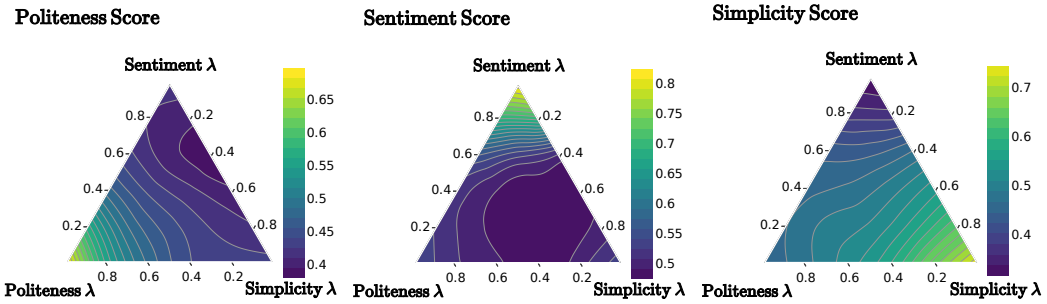


Figure 4: **Effect of λ_i on interpolation between the sentiment, politeness, and simplicity dimensions for $\alpha_i = 1$.** The vertices of the triangle represent the models with $\alpha_i = 1$ for each of the three attribute dimensions. The scores in the simplex of λ weights between the three control dimensions smoothly interpolate between the extreme models.

more inference passes than our approach and the same set of fine-tuned models, it is much more computationally expensive for applications with many attribute dimensions or inference passes. These results indicate that for one control attribute, interpolating between two endpoint models yields fine-grained control over the model outputs that outperforms or is on par with prior more approaches while maintaining very inexpensive inference computation.

3.2 Multi-dimensional interpolation

In real-world LLM applications, users often have diverse output preferences across multiple control dimensions at once, and these preferences may change dynamically for different inputs to the LLM. In this section, we show that linear interpolation between fine-tuned parameter-efficient adapters can be used to parametrize a whole convex hull of models, which can be used to dynamically generate text with attribute levels specified on-the-fly.

3.2.1 Parametrization of the convex hull

Analysis of interpolation parameter α: We find that when interpolating across up to five attribute dimensions, modifying the weight parameters λ_i and α_i results in predictable, fine-grained control over the attribute scores for the desired attributes while having a comparatively small effect on the remaining attributes. Figure 2 shows that increasing the α_i parameter smoothly increases the ith attribute score. Similarly, as the model mixture parameter λ_i increases, the effect on the attribute score of changing α_i increases.

Analysis of interpolation parameter λ: We also analyze the relationship throughout the whole simplex of λ weights for sets of three control dimensions in Figure 4 (as well as Figures 17-30 in the Appendix). For each set of three attributes listed, these plots show the scores in the three dimensional simplex of mixing weights λ for which $\sum_i \lambda_i = 1$. The value of the interpolation weight α_i for each of the attributes is equal to 1 in Figure 4, so increasing the λ weight of each attribute should increase the attribute score. For the majority of attributes, we observe an approximately even increase in score as λ_i for a given attribute dimension increases, regardless of the other λ_j parameters.

Acknowledgments and Disclosure of Funding

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE 2140743. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors(s) and do not necessarily reflect the views of the National Science Foundation.

References

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text generation. *CoRR*, abs/1912.02164, 2019. URL http://arxiv.org/abs/1912.02164.

Jasper Dekoninck, Marc Fischer, Luca Beurer-Kellner, and Martin Vechev. Controlled text generation via language model arithmetic. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=SLw9fp4yI6.

Ronen Eldan and Yuanzhi Li. Tinystories: How small can language models be and still speak coherent english?, 2023.

Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1082. URL https://aclanthology.org/P18-1082.

Chuang Gan, Zhe Gan, Xiaodong He, Jianfeng Gao, and Li Deng. Stylenet: Generating attractive visual captions with styles. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 955–964, 2017. doi: 10.1109/CVPR.2017.108.

Rohit Gandikota, Joanna Materzynska, Tingrui Zhou, Antonio Torralba, and David Bau. Concept sliders: Lora adaptors for precise control in diffusion models, 2023.

Chi Han, Jialiang Xu, Manling Li, Yi Fung, Chenkai Sun, Nan Jiang, Tarek Abdelzaher, and Heng Ji. Lm-switch: Lightweight language model conditioning in word embedding space, 2023.

Evan Hernandez, Belinda Z. Li, and Jacob Andreas. Inspecting and editing knowledge representations in language models, 2023.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685, 2021. URL https://arxiv.org/abs/2106.09685.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. Lorahub: Efficient cross-task generalization via dynamic lora composition, 2024.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic, 2023.

Weisen Jiang, Baijiong Lin, Han Shi, Yu Zhang, Zhenguo Li, and James T. Kwok. Byom: Building your own multi-task model for free, 2024.

Di Jin, Zhijing Jin, Zhiting Hu, Olga Vechtomova, and Rada Mihalcea. Deep Learning for Text Style Transfer: A Survey. *Computational Linguistics*, 48(1):155–205, 04 2022. ISSN 0891-2017. doi: 10.1162/coli_a_00426. URL https://doi.org/10.1162/coli_a_00426.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. CTRL: A conditional transformer language model for controllable generation. *CoRR*, abs/1909.05858, 2019. URL http://arxiv.org/abs/1909.05858.

Muhammad Khalifa, Hady Elsahar, and Marc Dymetman. A distributional approach to controlled text generation. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=jWkw45-9AbL.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. GeDi: Generative discriminator guided sequence generation. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4929–4952, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.424. URL https://aclanthology.org/2021.findings-emnlp.424.

Wojciech Kryscinski, Nazneen Fatema Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir R. Radev. Booksum: A collection of datasets for long-form narrative summarization. *CoRR*, abs/2105.08209, 2021. URL https://arxiv.org/abs/2105.08209.

Sachin Kumar, Eric Malmi, Aliaksei Severyn, and Yulia Tsvetkov. Controlled text generation as continuous optimization with multiple constraints. *CoRR*, abs/2108.01850, 2021. URL https://arxiv.org/abs/2108.01850.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model, 2023.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *CoRR*, abs/2101.00190, 2021. URL https://arxiv.org/abs/2101.00190.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. On-the-fly controlled text generation with experts and anti-experts. *CoRR*, abs/2105.03023, 2021. URL https://arxiv.org/abs/2105.03023.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

Aman Madaan, Amrith Setlur, Tanmay Parekh, Barnabás Póczos, Graham Neubig, Yiming Yang, Ruslan Salakhutdinov, Alan W. Black, and Shrimai Prabhumoye. Politeness transfer: A tag and generate approach. *CoRR*, abs/2004.14257, 2020. URL https://arxiv.org/abs/2004.14257.

Michael Matena and Colin Raffel. Merging models with fisher-weighted averaging. *CoRR*, abs/2111.09832, 2021. URL https://arxiv.org/abs/2111.09832.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *CoRR*, abs/1609.07843, 2016. URL http://arxiv.org/abs/1609.07843.

Fatemehsadat Mireshghallah, Kartik Goyal, and Taylor Berg-Kirkpatrick. Mix and match: Learning-free controllable text generation using energy language models, 2022.

Kai Nylund, Suchin Gururangan, and Noah A. Smith. Time is encoded in the weights of finetuned language models, 2023.

Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models, 2023.

Damian Pascual, Beni Egressy, Clara Meister, Ryan Cotterell, and Roger Wattenhofer. A plug-and-play method for controlled text generation. *CoRR*, abs/2109.09707, 2021. URL https://arxiv.org/abs/2109.09707.

Jing Qian, Li Dong, Yelong Shen, Furu Wei, and Weizhu Chen. Controllable natural language generation with contrastive prefixes, 2022.

Lianhui Qin, Sean Welleck, Daniel Khashabi, and Yejin Choi. Cold decoding: Energy-based constrained text generation with langevin dynamics, 2022.

Alexandre Ramé, Guillaume Couairon, Mustafa Shukor, Corentin Dancette, Jean-Baptiste Gaya, Laure Soulier, and Matthieu Cord. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards, 2023.

Sudha Rao and Joel Tetreault. Dear sir or madam, may I introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality style transfer. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 129–140, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1012. URL https://aclanthology.org/N18-1012.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In David Yarowsky, Timothy Baldwin, Anna Korhonen, Karen Livescu, and Steven Bethard, editors, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170.

Nishant Subramani, Nivedita Suresh, and Matthew E. Peters. Extracting latent steering vectors from pretrained language models, 2022.

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, 2023.

Alexander Matt Turner, Lisa Thiergart, David Udell, Gavin Leech, Ulisse Mini, and Monte MacDiarmid. Activation addition: Steering language models without optimization, 2023.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time, 2022.

Jingjing Xu, Xu Sun, Qi Zeng, Xiaodong Zhang, Xuancheng Ren, Houfeng Wang, and Wenjie Li. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 979–988, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1090. URL https://aclanthology.org/P18-1090.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models, 2023.

Kevin Yang and Dan Klein. FUDGE: Controlled text generation with future discriminators. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3511–3535, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.276. URL https://aclanthology.org/2021.naacl-main.276.

Jinghan Zhang, Shiqi Chen, Junteng Liu, and Junxian He. Composing parameter-efficient modules with arithmetic operations, 2023.

Wangchunshu Zhou, Yuchen Eleanor Jiang, Ethan Wilcox, Ryan Cotterell, and Mrinmaya Sachan. Controlled text generation with natural language instructions, 2023.

A Appendix

A.1 Related Work

A.1.1 Controllable text generation (CTG)

As it is crucial to constrain generated text in many downstream applications, CTG has been a recent focus of NLP research. Methods such as CTRL [Keskar et al., 2019] and GeDI [Krause et al., 2021] pretrain language models on text prepended with control codes and generate text conditioned on the desired control. However, these methods require pretraining a new model if new controls are added, which is computationally expensive. To mitigate these issues, a variety of methods have been proposed to perform CTG without additional language model training. For example, Khalifa et al. [2021], Pascual et al. [2021], Yang and Klein [2021], Dekoninck et al. [2024] constrain language model outputs by modifying their output probability distributions. Li and Liang [2021], Qian et al. [2022] learn prefixes and Dathathri et al. [2019], Han et al. [2023] train additional classifiers to guide generation. Subramani et al. [2022], Hernandez et al. [2023], Li et al. [2023], Turner et al. [2023] control model outputs by changing activations at inference time. Kumar et al. [2021] optimize the inference decoding. Mireshghallah et al. [2022], Qin et al. [2022] use energy-based constrained generation and Zhou et al. [2023] use instruction tuning for CTG.

Among these, only the methods of Liu et al. [2021], Dekoninck et al. [2024] are composable and achieves fine-grained control over multiple attributes at once, so we provide these methods as a comparison in this paper. However, as these methods require composing multiple models at inference time, the inference cost is significantly higher than our approach, especially as the model size and number of controlled attributes increases.

A.1.2 Weight interpolation

Our work builds on prior work on linear weight interpolation, such as task vectors [Ilharco et al., 2023], parameter-efficient task vectors [Zhang et al., 2023], and model souping [Wortsman et al., 2022], as we use linear interpolation and weighted model averaging as the basis for our analysis. Prior work in this domain has focused mainly on improving multitask performance when composing fully fine-tuned models [Matena and Raffel, 2021, Yadav et al., 2023, Ortiz-Jimenez et al., 2023, Ramé et al., 2023] or parameter-efficient fine-tuned models [Huang et al., 2024, Jiang et al., 2024]. However, these methods all differ from our work, since they focus on combining model weights to improve a single multitask objective rather than analyzing performance across a wide range of flexible, diverse objectives. These approaches are orthogonal to our work and could be used in conjunction with it to better combine the α-interpolated models. Perhaps most similar to our work are methods that interpolate between the weights of fine-tuned models to control over a range of outputs [Gandikota et al., 2023, Nylund et al., 2023]. However, Gandikota et al. [2023] focus on the vision domain and use a fine-tuning objective specific to diffusion models, and Nylund et al. [2023] only analyze control over the time dimension.

A.2 Limitations

The main limitation of our work is that some pairs of attributes are correlated, so when a correlated model has a large mixing weight, it can unpredictably affect other control attributes. It would be valuable to investigate whether this correlation is inherent to the pair of tasks or if it can be eliminated. For example, text that is more polite might always be more formal. However, it may be the case that some correlations can be reduced by regularizing the LoRA updates to be more orthogonal to each other or by merging the α-interpolated using more sophisticated methods that have recently shown improvement over naive weight averaging in the multitask setting [Matena and Raffel, 2021, Yadav et al., 2023, Ortiz-Jimenez et al., 2023, Ramé et al., 2023].

Another limitation is that our method requires fine-tuned models and the average generation attribute scores are limited to the range between the attribute scores of the fine-tuned anchor models. The single attribute extrapolation results could be expanded upon to better understand when extrapolation can be used to extend the range of the control attribute style.

Additionally, human evaluation would be ideal for assessing text style, but is out of scope for this paper. The challenge is that, unlike prior work, we are interested in controlling text style attributes in

the region between the endpoint fine-tuned models rather than the endpoint performance. As a result, in order to adequately assess the performance of the interpolation with human-provided labels, we would need to query humans hundreds of times per prompt. A full-blown user study on this scale thus remains infeasible for this evaluation.

A.3 Ethics Statement

Continuous weight interpolation may output text that contains existing biases from the pre-trained models and fine-tuning datasets. It could also be used to control the level of undesirable attributes such as toxicity. However, we believe that this work is still beneficial overall, since it can be used to improve the experience of LLM users for a variety of applications, and these issues are faced by all pre-trained and fine-tuned language models.

A.4 Hyperparameters for fine-tuning

LoRA hyperparameter	**Value**
Batch size	64
Learning rate	5e-5
LoRA r	32
LoRA α	16
LoRA dropout	0.1
Max sequence length	128
Quantization	4 bit

Table 1: **Parameters for LoRA fine-tuning.** We use 20 epochs for fine-tuning the sentiment attribute models and 1 epoch for the remaining fine-tuned models. All experiments were run on single NVIDIA A100 80GB SXM GPU nodes.

Domain	**Llama2 split size**		**RoBERTa split size**
	Class 0	**Class 1**	
Sentiment Socher et al. [2013]	25k	30k	10k
Politeness Madaan et al. [2020]	78k	100k	20k
Formality Rao and Tetreault [2018]	104k	104k	10k
Simplicity [Kryscinski et al., 2021, Eldan and Li, 2023]	9k	100k	10k
Humor Gan et al. [2017]	100k	100k	20k

Table 2: **Fine-tuning splits.** We report the number of examples from each attribute dataset used to fine-tune Llama2-7b generation and RoBERTa attribute scoring models. Each split is sampled from the combined train, test, and validation set.

A.5 Model Arithmetic Formulation

For the model arithmetic [Dekoninck et al., 2024] comparison, we use the following formula, inspired by DExperts [Liu et al., 2021]:

$$M + \alpha(M_{\text{pos}} - M_{\text{neg}})$$

Here, M is the base model, M_{pos} is the model conditioned for class 1, and M_{neg} is the model conditioned for class 0. The system prompts used for conditioning are listed in Table 3.

Conditioned model name	Llama2-7b-chat System Prompt
sentiment_pos	"The following is a positive story, with a very positive sentiment and a very positive tone."
sentiment_neg	"The following is a negative story, with very negative sentiment and a very negative tone."
formality_pos	"The following is a formal story, with very formal language and a very formal tone."
formality_neg	"The following is an informal story, with very informal language and a very informal tone."
simplicity_pos	"The following is a simple story, with very simple language."
simplicity_neg	"The following is a complex story, with very complex language."
humor_pos	"The following is a humorous story, with very humorous language and a very humorous tone."
humor_neg	"The following is a nonhumorous story, with factual language and a very serious tone."
politeness_pos	"The following is a polite story, with very polite language and a very polite tone."
politeness_neg	"The following is an impolite story, with very impolite language and a very impolite tone."

Table 3: **System prompts used for conditioning model arithmetic.** .

A.6 Plots with errorbars

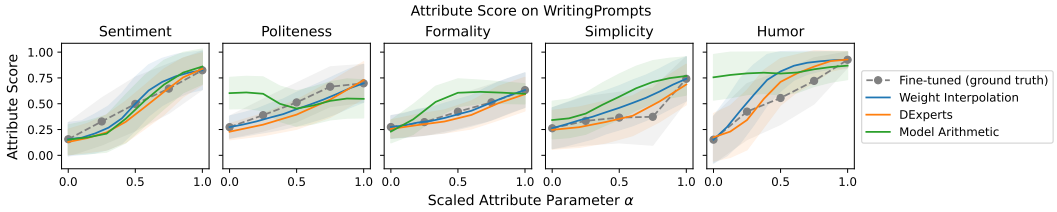


Figure 5: **Interpolated models recover custom fine-tuned models across the interpolation range**. We show the attribute scores with standard deviation error bars for our model with weight α compared to DExperts [Liu et al., 2021] and model arithmetic [Dekoninck et al., 2024] with α scaled such that the scaled $\alpha = 0$ and $\alpha = 1$ models have the same score as the fine-tuned endpoint models. Our approach most closely follows the trend of the ground truth fine-tuned models.

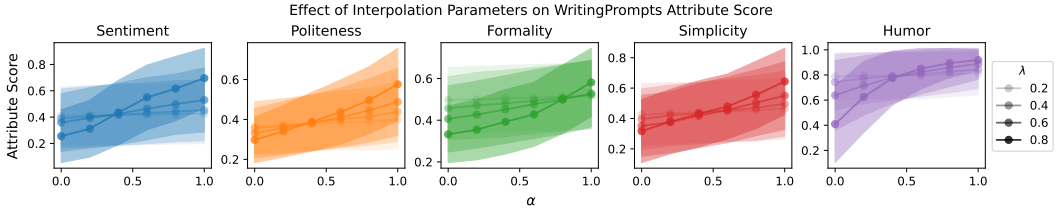


Figure 6: **Effect of α_i and λ_i on 5-dimensional interpolation**. For each attribute, we show the attribute scores with standard deviation errorbars for models with the given α_i and λ_i parameters, with all four other $\alpha_j = 1$ and $\lambda_j = (1 - \lambda_i)/4$. We find that increasing α_i consistently increases the attribute score and increasing λ_i consistently increases the effect of α_i.

A.7 Linear weight extrapolation

Linear extrapolation: Figure 7 shows the attribute scores when extrapolating linearly beyond the two fine-tuned models along the vector between them. We find that even beyond the region of interpolation between the two fine-tuned models, there is a small stable extrapolation regime up to α values of around -1 and 2 (Figure 7). In this region, for many of the attributes, the attribute score continues to behave predictably as α is increased. However, beyond the stable extrapolation values, there is an unstable extrapolation regime where the attribute score changes unpredictably as α is varied. This is likely due to the model output quality degrading, since as shown in Figure 8 and Figure 12, the model perplexity increases sharply and the diversity decreases starting near the edges of the stable extrapolation regime. While prior work has shown that linear weight extrapolation can be used for tasks such as model unlearning [Ilharco et al., 2023, Zhang et al., 2023], these results provide a cautionary tale against extrapolating too far, as they suggest that this ability only extends to a certain threshold before the attribute score and model outputs become unpredictable due to poor quality outputs. For the remainder of our experiments, we thus focus on the interpolation regime.

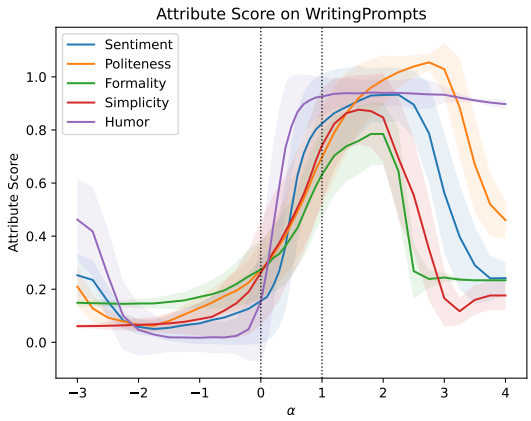


Figure 7: **Effect of linear weight extrapolation for a single attribute dimension.** For each style attribute, we report the attribute score when linearly extrapolating beyond the fine-tuned models ($\alpha < 0$ and $\alpha > 1$). There is a stable region where the score changes smoothly until a certain point (around α equal to -1 and 2), where performance degrades and the extrapolation is unstable.

A.8 Perplexity analysis

A.8.1 Perplexity of interpolated and extrapolated models

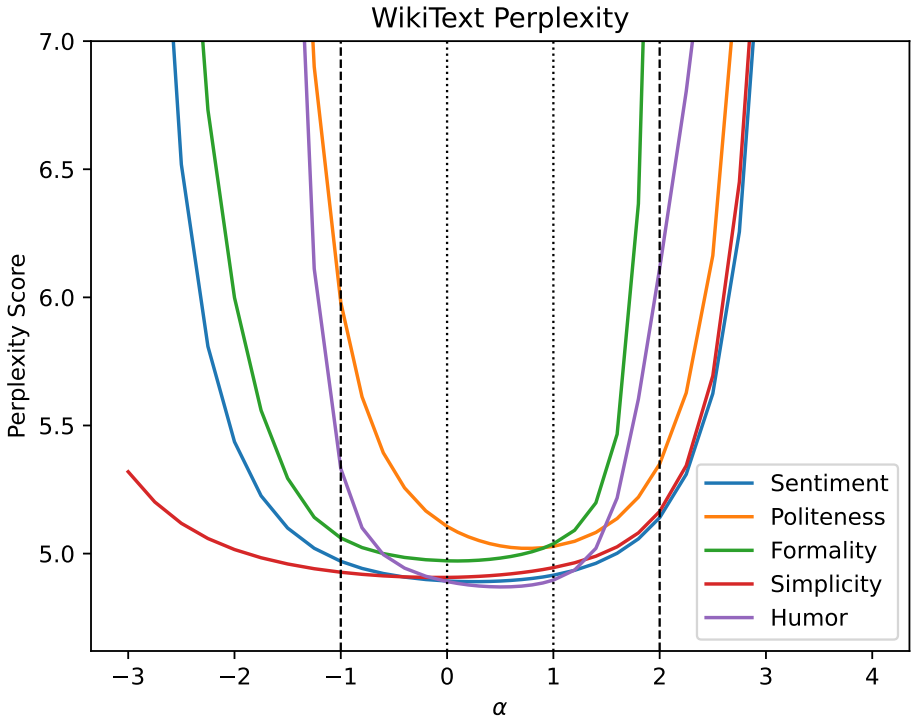


Figure 8: **Wikitext perplexity of linearly interpolated and extrapolated models.** We report the average perplexity (lower is better) of each model from Figure 7 on the Wikitext test set. For all of the interpolated models ($\alpha \in [0, 1]$), the perplexity is either better than or between the performance of the endpoint fine-tuned models. For the extrapolated models ($\alpha < 0.0$ and $\alpha > 1.0$), the perplexity increases rapidly beyond α values of around -1 and 2. We clip the y-axis at 7.0 for readability (the full plot is shown in Figure 9).

A.8.2 Perplexity comparison to fine-tuned models

In this section, we compare the WikiText perplexity of weight-interpolated models versus the perplexity of fine-tuned models trained on data with α fraction from class 1 and $1 - \alpha$ fraction from class 0 for each attribute.

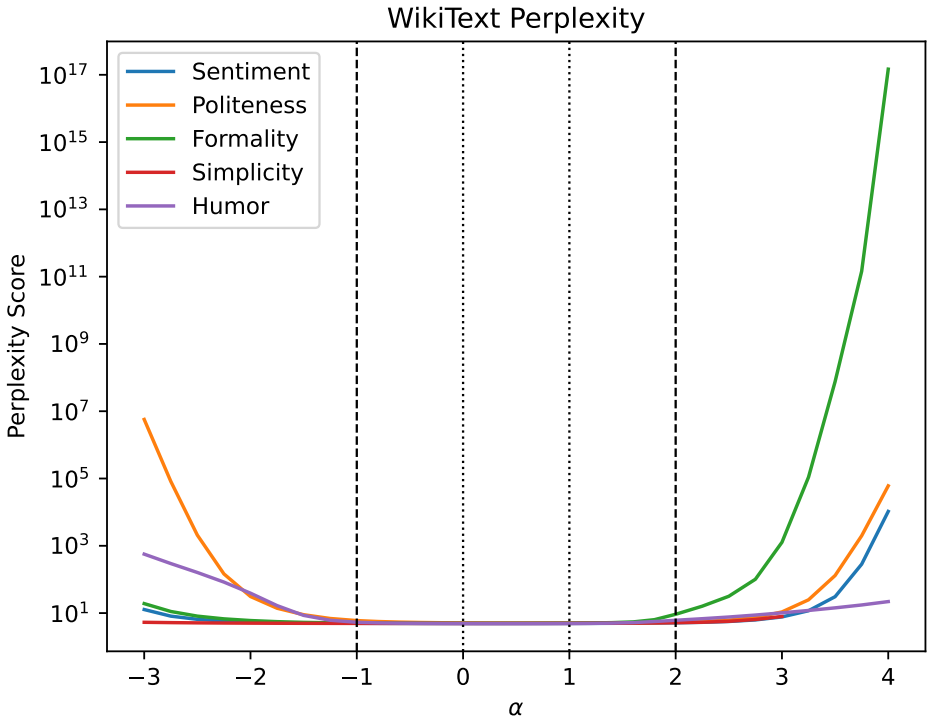


Figure 9: **Wikitext perplexity of linearly interpolated and extrapolated models.** We report the average perplexity of each model from Figure 7 on the Wikitext test set. For the extrapolated models not shown in Figure 8, the perplexity increases rapidly.

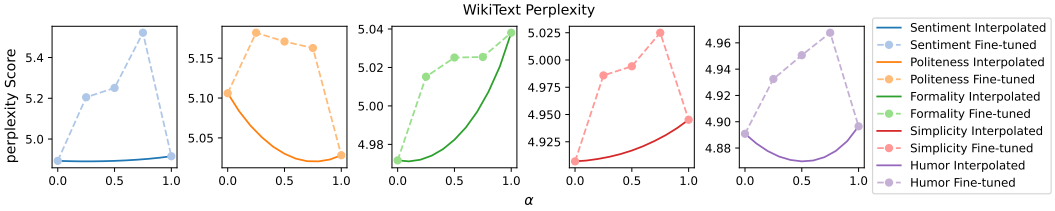


Figure 10: **Wikitext perplexity for single attribute interpolated versus fine-tuned models.** We report the perplexity when linearly interpolating between the models with weight α as compared to the perplexity for models trained with α fraction class 1 and $1 - \alpha$ fraction class 0 data. The perplexity for the interpolated models is lower than that of the endpoint fine-tuned models for the intermediate values of α.

A.9 Diversity analysis

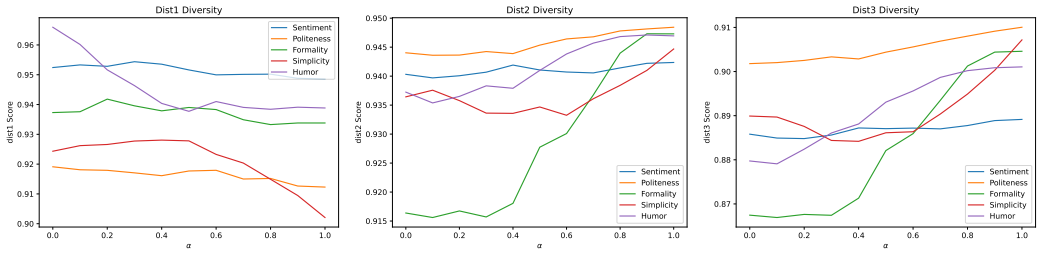


Figure 11: **N-gram diversity scores for the interpolated models.** We report the 1-, 2-, and 3-gram diversity scores for the single-attribute interpolated models. The diversity scores remain similar to or between those of the endpoint fine-tuned models within the interpolation region.

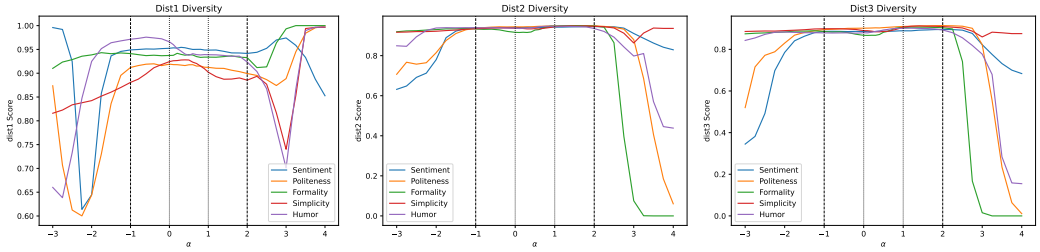


Figure 12: **N-gram diversity scores for the extrapolated models.** We report the 1-, 2-, and 3-gram diversity scores for the single-attribute interpolated and extrapolated models. The diversity scores remain similar to or between those of the endpoint fine-tuned models within the interpolation region ($\alpha \in [0, 1]$) and in the stable extrapolation region ($\alpha \in [-1, 0) \cup (1, 2]$), but become unstable beyond the stable extrapolation region.

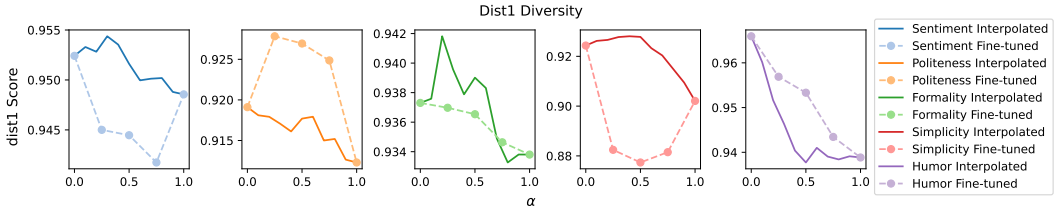


Figure 13: **1-gram diversity comparison of single attribute interpolated versus fine-tuned models.** We report the dist1 diversity scores for the single-attribute interpolated models with weight α as compared to the perplexity for models trained with α fraction class 1 and $1 - \alpha$ fraction class 0 data. The diversity scores remain similar to or between those of the endpoint fine-tuned models within the interpolation region.

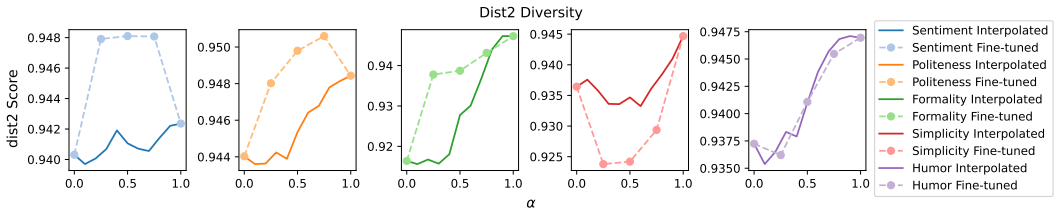


Figure 14: **2-gram diversity comparison of single attribute interpolated versus fine-tuned models.** We report the dist2 diversity scores for the single-attribute interpolated models with weight α as compared to the perplexity for models trained with α fraction class 1 and $1 - \alpha$ fraction class 0 data. The diversity scores remain similar to or between those of the endpoint fine-tuned models within the interpolation region.

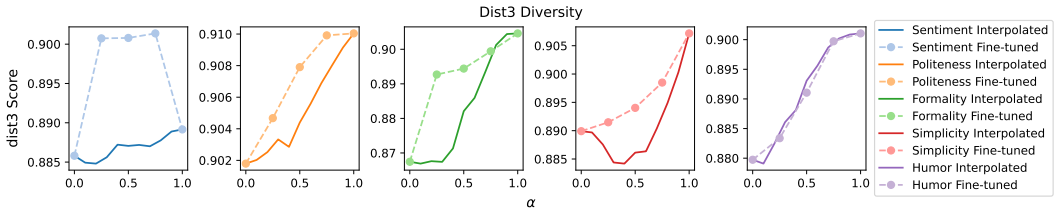


Figure 15: **3-gram diversity comparison of single attribute interpolated versus fine-tuned models.** We report the dist3 diversity scores for the single-attribute interpolated models with weight α as compared to the perplexity for models trained with α fraction class 1 and $1 - \alpha$ fraction class 0 data. The diversity scores remain similar to or between those of the endpoint fine-tuned models within the interpolation region.

A.10 Generation example

We provide an example generation to compare between weight interpolated models for a single attribute and prompting an instruction-tuned model (Llama2-13b-chat). We provide the model generations for the following prompt set-up inspired by Han et al. [2023]:

- "Complete this story so that it embodies a sentiment score of 0.5, where 0 is negative and 1 is positive: You find a rip in time walking through the alleys . You enter it to find yourself "
- For each style attribute, we replace the words "sentiment", "negative", and "positive" with the corresponding attribute and class names, and 0.5 with the corresponding α score.
- We report the output until the first occurrence of a newline character or the amount of output that fits in 2-3 lines of the table.

We find that in general, it is challenging to achieve fine-grained control over the output attributes with prompting as compared to interpolation. The prompted model often does not properly account for the α value and produces outputs at one attribute extreme or the other regardless of α. Furthermore, for dimensions that are less commonly used in CTG (ie formality), the prompted model often produces very similar outputs for each value of α.

Table 4: **Generation comparison:** we present a comparison of generations for single attribute interpolation versus prompted Llama2-13b-chat for various α values with the prompt "You find a rip in time walking through the alleys . You enter it to find yourself "

Dimension	α	Single attribute interpolation	Prompted Llama2-13b-chat
Sentiment	0.0	40 minutes later still wondering what the h*ll you did wrong. [...]	months in the future. Everything has changed but... [...]
	0.3	12 hours earlier with your hopes and sanity battered only to discover the time rip still [...]	months in the future. Everything has changed but it seems the world has gotten better. [...]
	0.7	40 years earlier, passing through an archway into a deeply familiar but different world. [...]	10 minutes in the past, before you were born. You decide to go back in time and give your younger self [...]
	1.0	10 years older and wondering how it happened. [...]	20 years earlier, in a world before the wars, global warming and the division of society. [...]
Politeness	0.0	100 miles away from the starting line, but who told you to quit. [...]	30 years earlier. You ask the current you what to do next to maximize [...]
	0.3	100 miles away from a nobody jerk. you find a rip in space boarding a bus on west 96th [...]	10 minutes in the past, before you were scheduled to meet a friend for coffee. You realize that by altering the past, [...]
	0.7	10 years in the future. you are discovering all sorts of things. it comes to you [...]	7 years ago in a Cafe you have been to before. [...]
	1.0	10 years ago, trying to figure out where the next stride will take you or perhaps where [...]	7 years ago in a Cafe you have been to before, supposed to meet with a friend that never showed up. [...]
Formality	0.0	20! LOL that doesn't seem right or fair. [...]	in the past... [...]
	0.3	7 years older. END OF STORY!! Things were going well until you started to take [...]	in the past, an absolute fantasy. You see a young version of yourself there, who [...]
	0.7	40 or older, happily married, and with 3 kids. [...]	10 minutes in the past, before the recent break-in at your apartment. [...]
	1.0	21 years older from your prior adventures. [...]	10 minutes in the past, before the recent break-in at your office. [...]
Simplicity	0.0	on a black public hillside, and a yellow sun butchered and bleeding in an ugly sky, and you know the cut of sandstone in the [...]	20 years earlier, in a world before the great collapse. Children are playing, birds are chirping, and people are smiling. [...]
	0.3	3 kilometres outside of town at a main road. you slowly move forward looking around your surroundings. Seeing a man sitting under a [...]	7 years ago in a crucial moment of your past. [...]
	0.7	300 years before your time. Some kids around you are running off to play in the forest. You stand there trying to figure out what to do [...]	10 minutes in the past, before the recent downpour. How do you handle it? [...]
	1.0	300 years back! It is 1828 in London. You stay in the alley until it becomes fully sunny. [...]	10 minutes in the past, before the recent downpour of rains and flooding. [...]
Humor	0.0	30 years in the past under another name. You're married to an old fling [...]	7 years ago in a parking lot looking 7 years younger. [...]
	0.3	30 years back, walking through the alleys. So much for not being surprised. [...]	7 years ago in a parking lot looking 7 years younger. You see a car you can't remember [...]
	0.7	75 years in the future, Washington DC's Newbridge Apartments has become an urban theme park [...]	20 years earlier, in high school. Your younger self is looking at you, confused. You then see yourself in high school and [...]
	1.0	255 years into the future, the day at the 'harmless' age of sixty million, a desperate, crazed - looking [...]	10 minutes in the past, but you bring a hand-held portal weapon with you. [...]

A.11 Multi-dimensional scaling (MDS) analysis of fine-tuned models

We project the weights of the LoRA fine-tuned endpoint models, as well as some of the interpolated models, into two dimensions using multi-dimensional scaling (MDS). As shown in Figure 16, we find that the interpolating between the endpoint fine-tuned models generally results in models that are closer to the base model. This is expected behavior since we would anticipate that the base model is fairly neutral with respect to all attributes.

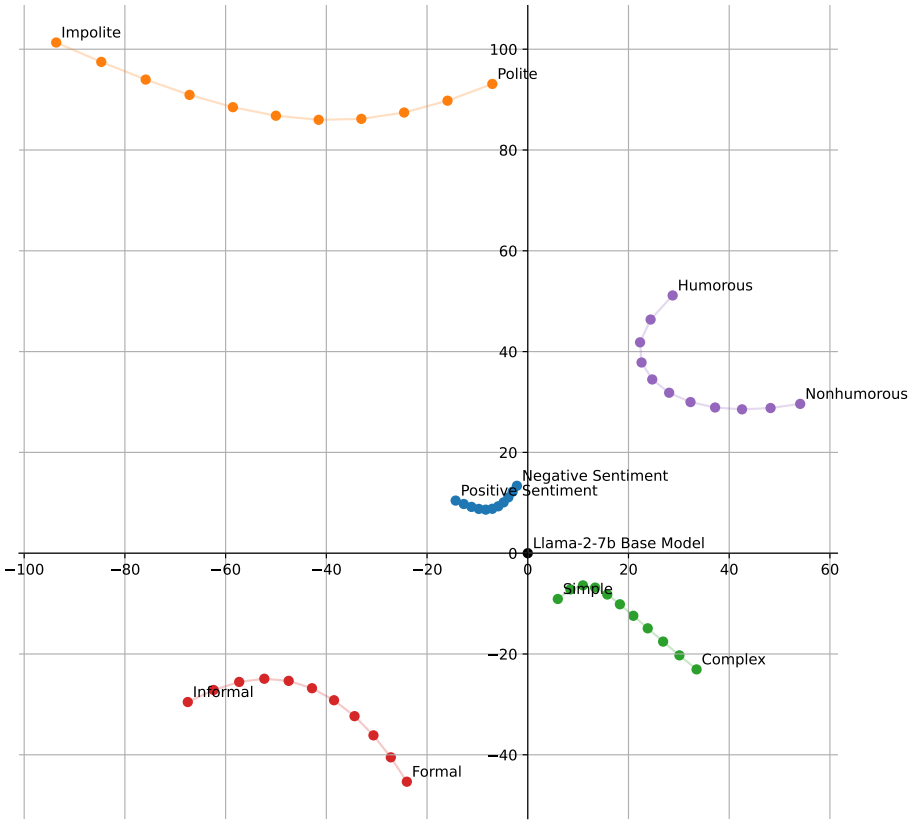


Figure 16: **Multi-dimensional scaling (MDS) plot for the fine-tuned models and linear interpolations.** This plot shows the 2-dimensional MDS projection of the fine-tuned anchor models and the models interpolated at intervals of 0.1. This corresponds to the models in Figure 3.

A.12 Additional multi-dimensional lambda simplex plots

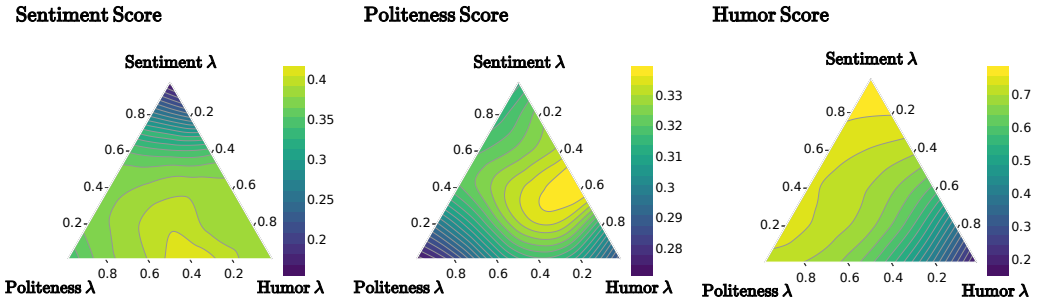


Figure 17: **Effect of λ_i on interpolation between the sentiment, politeness, and humor dimensions for $\alpha_i = 0$.** The vertices of the triangle represent the models with $\alpha_i = 0$ for each of the three dimensions.

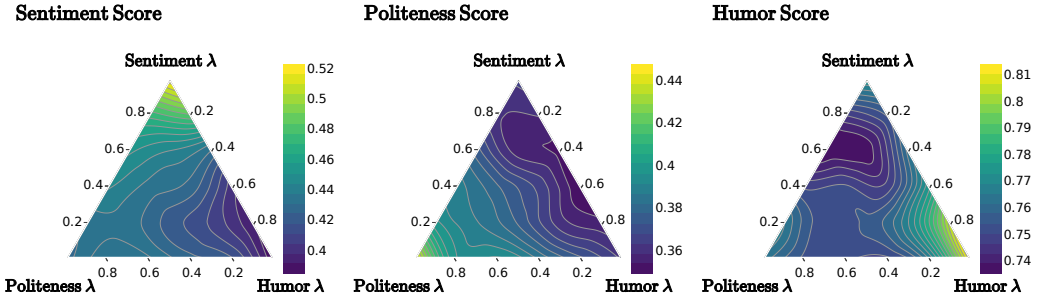


Figure 18: **Effect of λ_i on interpolation between the sentiment, politeness, and humor dimensions for $\alpha_i = 0.5$.** The vertices of the triangle represent the models with $\alpha_i = 0.5$ for each of the three dimensions.

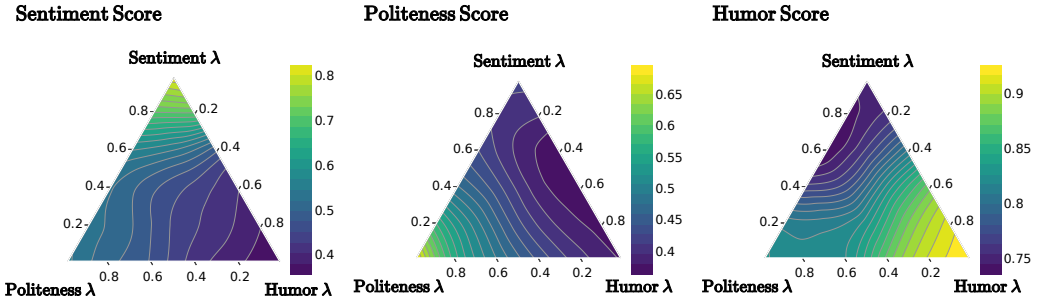


Figure 19: **Effect of λ_i on interpolation between the sentiment, politeness, and humor dimensions for $\alpha_i = 1$.** The vertices of the triangle represent the models with $\alpha_i = 1$ for each of the three dimensions.

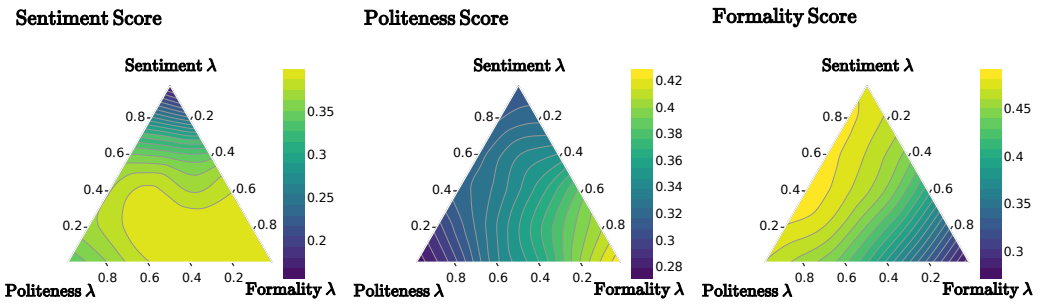


Figure 20: **Effect of λ_i on interpolation between the sentiment, politeness, and formality dimensions for $\alpha_i = 0$.** The vertices of the triangle represent the models with $\alpha_i = 0$ for each of the three dimensions.

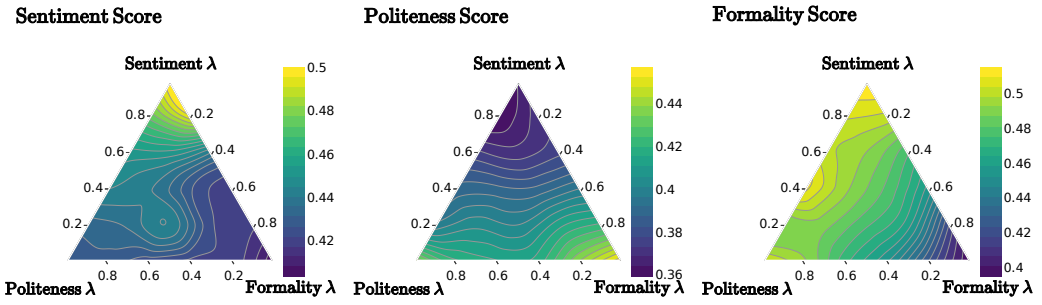


Figure 21: **Effect of λ_i on interpolation between the sentiment, politeness, and formality dimensions for $\alpha_i = 0.5$.** The vertices of the triangle represent the models with $\alpha_i = 0.5$ for each of the three dimensions.

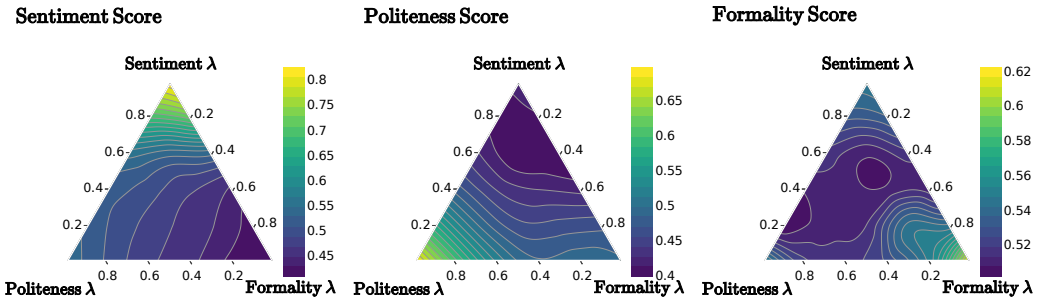


Figure 22: **Effect of λ_i on interpolation between the sentiment, politeness, and formality dimensions for $\alpha_i = 1$.** The vertices of the triangle represent the models with $\alpha_i = 1$ for each of the three dimensions.

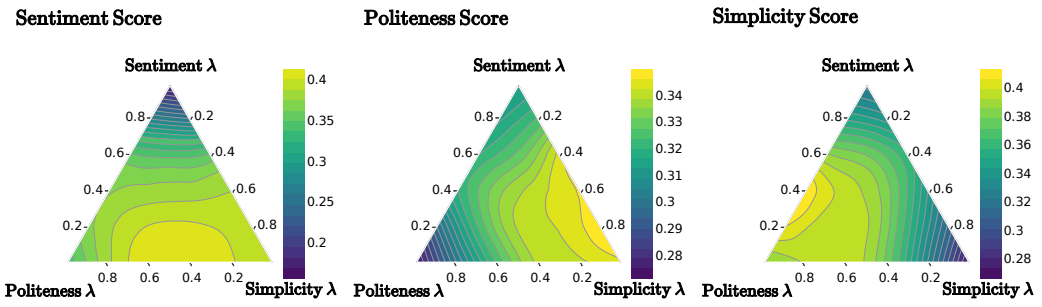


Figure 23: **Effect of λ_i on interpolation between the sentiment, politeness, and simplicity dimensions for $\alpha_i = 0$.** The vertices of the triangle represent the models with $\alpha_i = 0$ for each of the three dimensions.

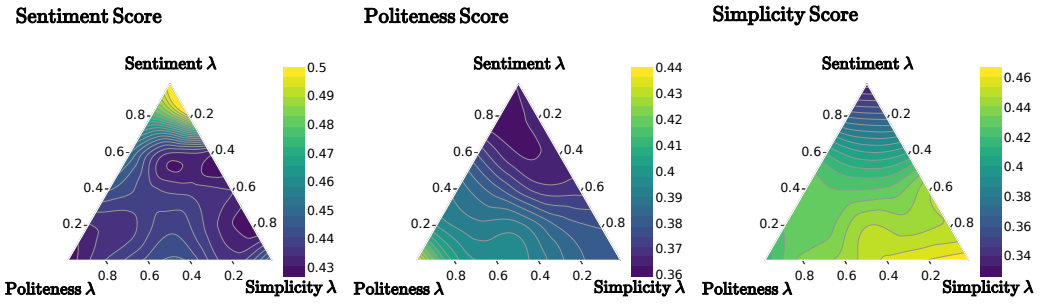


Figure 24: **Effect of λ_i on interpolation between the sentiment, politeness, and simplicity dimensions for $\alpha_i = 0.5$.** The vertices of the triangle represent the models with $\alpha_i = 0.5$ for each of the three dimensions.

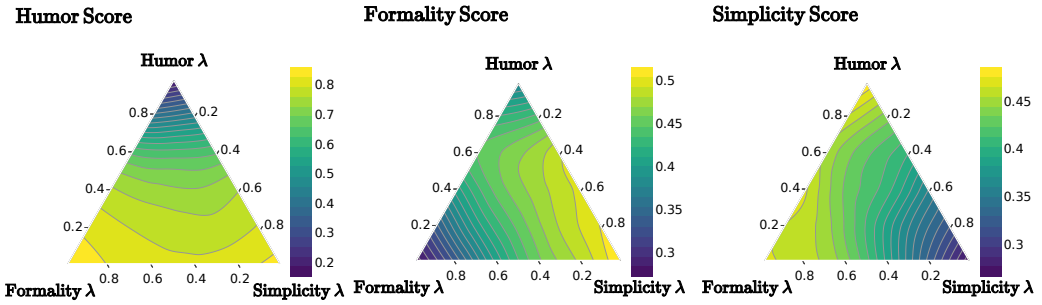


Figure 25: **Effect of λ_i on interpolation between the humor, formality, and simplicity dimensions for $\alpha_i = 0$.** The vertices of the triangle represent the models with $\alpha_i = 0$ for each of the three dimensions.

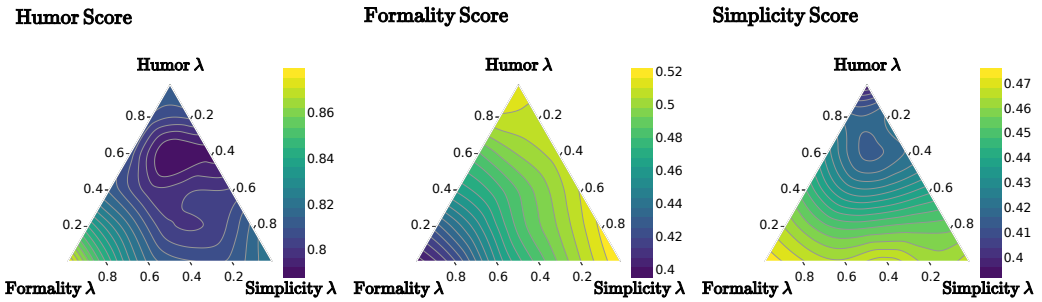
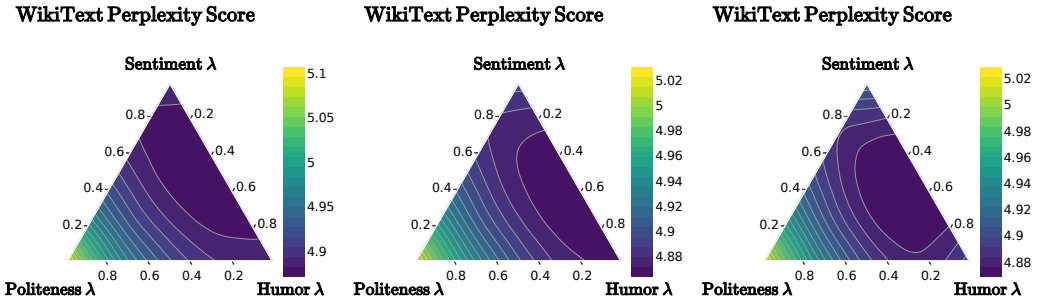
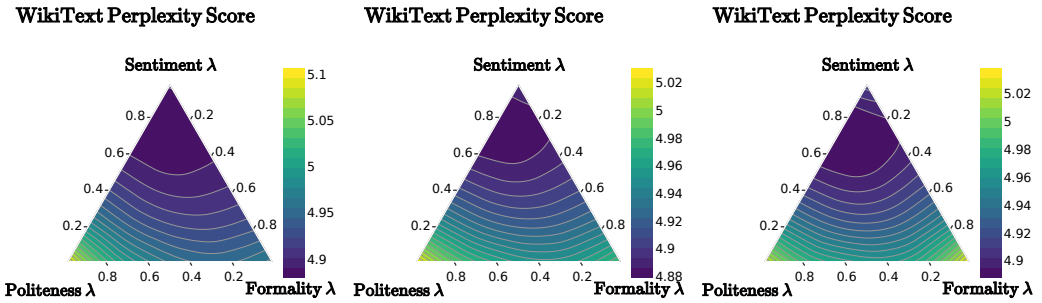


Figure 26: **Effect of λ_i on interpolation between the humor, formality, and simplicity dimensions for $\alpha_i = 0.5$.** The vertices of the triangle represent the models with $\alpha_i = 0.5$ for each of the three dimensions.



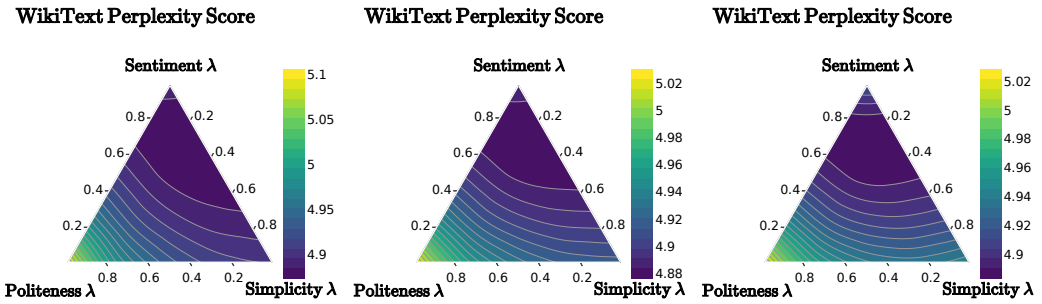
(a) $\alpha_i = 0.0$ (b) $\alpha_i = 0.5$ (c) $\alpha_i = 1.0$

Figure 27: **Effect of λ_i on perplexity interpolation between the sentiment, politeness, and humor dimensions for various α_i values.** The vertices of the triangle represent the models with the given α_i value for each of the three dimensions. The perplexity for each model is bounded above by the perplexities of the fine-tuned anchor models.



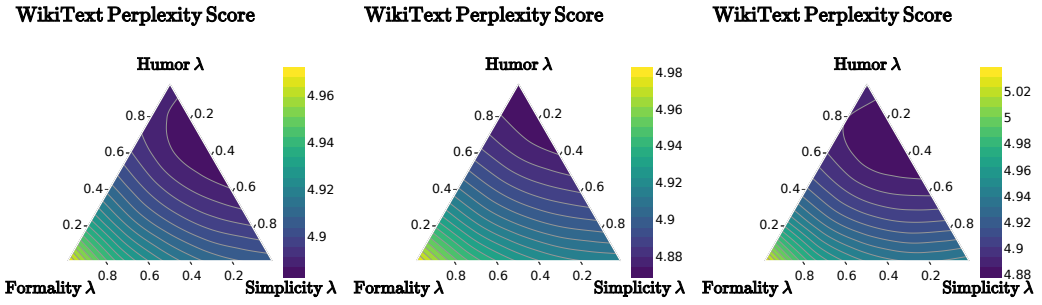
(a) $\alpha_i = 0.0$ (b) $\alpha_i = 0.5$ (c) $\alpha_i = 1.0$

Figure 28: **Effect of λ_i on perplexity interpolation between the sentiment, politeness, and formality dimensions for various α_i values.** The vertices of the triangle represent the models with the given α_i value for each of the three dimensions. The perplexity for each model is bounded above by the perplexities of the fine-tuned anchor models.



(a) $\alpha_i = 0.0$ (b) $\alpha_i = 0.5$ (c) $\alpha_i = 1.0$

Figure 29: **Effect of λ_i on perplexity interpolation between the sentiment, politeness, and simplicity dimensions for various α_i values.** The vertices of the triangle represent the models with the given α_i value for each of the three dimensions. The perplexity for each model is bounded above by the perplexities of the fine-tuned anchor models.



(a) $\alpha_i = 0.0$ (b) $\alpha_i = 0.5$ (c) $\alpha_i = 1.0$

Figure 30: **Effect of λ_i on perplexity interpolation between the humor, formality, and simplicity dimensions for various α_i values.** The vertices of the triangle represent the models with the given α_i value for each of the three dimensions. The perplexity for each model is bounded above by the perplexities of the fine-tuned anchor models.

A.13 Similarity between pairs of fine-tuned models

Given the results from the simplex plots, we analyze the relationships between the fine-tuned endpoint models to better understand the attribute score correlations. Figure 31, which plots the average cosine similarity between the LoRA layers of each pair of models, shows that the LoRA weights are relatively orthogonal to each other in most cases. We hypothesize that the lower orthogonality between each pair of endpoint models for the same attribute is because the models are trained on similar datasets. This is supported by the fact that the simple and complex models are the most orthogonal of the pairs of endpoint models and they are the only two models trained on different datasets rather than different classes from the same dataset. In addition, the humor models tend to be the least orthogonal to the other models (such as politeness), so this may provide a partial explanation for why some of the other models were correlated with a higher humor score.

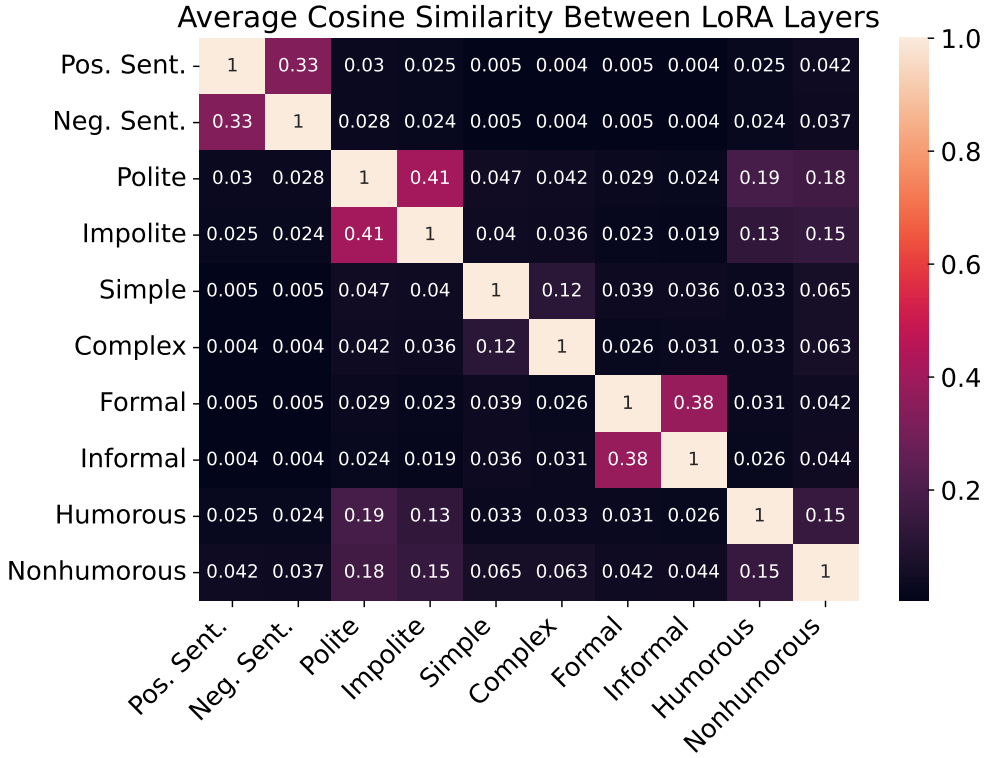


Figure 31: **Cosine similarity of LoRA weights averaged across layers between each pair of fine-tuned anchor models.** The LoRA weights are all relatively orthogonal to each other, except some of the two endpoint models for the same attribute are less orthogonal to each other, as well as the politeness and humorous models.

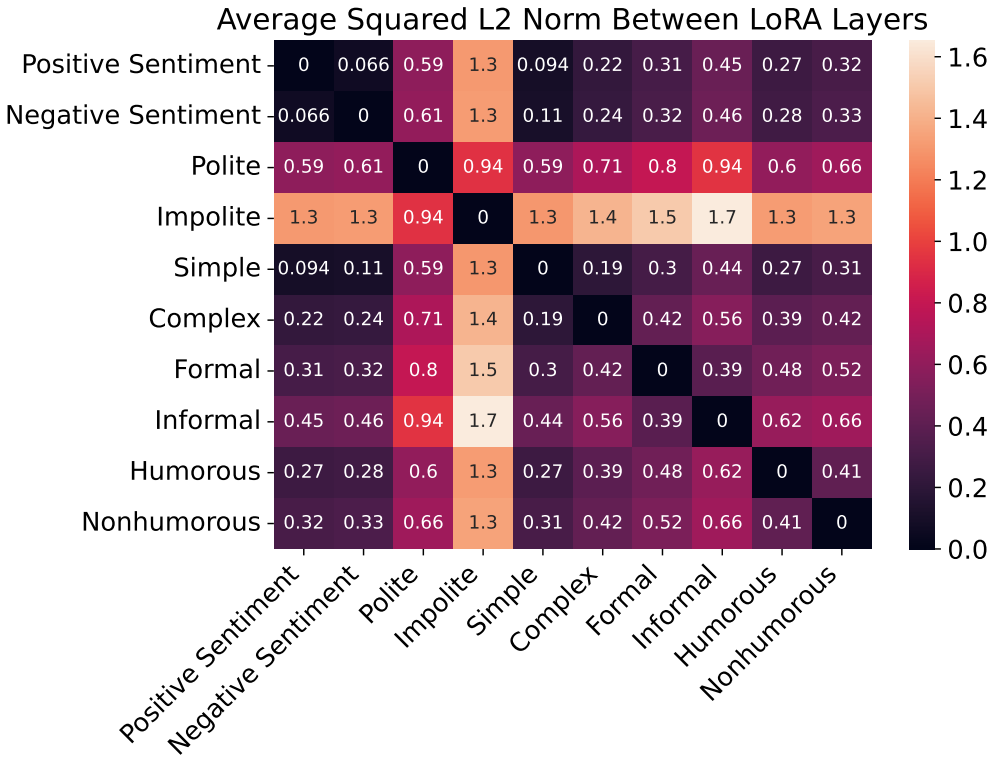


Figure 32: **Average pairwise squared L2 norms between LoRA layers.** The fine-tuned anchor models trained on the class with attribute score of 1 tend to be closer to the other models than those trained on the class with attribute score of 0. The polite and impolite models are the farthest from the other models.