Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/K17-1034. URL https://www.aclweb.org/anthology/K17-1034.

Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few shot learning. *CoRR*, abs/1707.09835, 2017. URL http://arxiv.org/abs/1707.09835.

Yuxuan Ma. distilgpt2-finetuned-wikitext2. https://huggingface.co/MYX4567/distilgpt2-finetuned-wikitext2, July 2021.

Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24:109–165, 1989. ISSN 0079-7421. doi: https://doi.org/10.1016/S0079-7421(08)60536-8. URL https://www.sciencedirect.com/science/article/pii/S0079742108605368.

German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2019.01.012. URL https://www.sciencedirect.com/science/article/pii/S0893608019300231.

Eunbyung Park and Junier B Oliva. Meta-curvature. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, 2018.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1250. URL https://aclanthology.org/D19-1250.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners, 2019. URL https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

R. Ratcliff. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97 2:285–308, 1990.

Sachin Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017.

Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model?, 2020.

Anton Sinitsin, Vsevolod Plokhotnyuk, Dmitry Pyrkin, Sergei Popov, and Artem Babenko. Editable neural networks. In *ICLR*, 2020. URL https://openreview.net/forum?id=HJedXaEtvS.

Matthew Sotoudeh and Aditya V. Thakur. Provable repair of deep neural networks. *ArXiv*, abs/2104.04413, 2021.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In *NAACL-HLT*, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Ben Wang and Aran Komatsuzaki. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax, May 2021.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu ji, Guihong Cao, Daxin Jiang, and Ming Zhou. K-adapter: Infusing knowledge into pre-trained models with adapters, 2020. URL http://arxiv.org/abs/2002.01808.

Shibo Wang and Pankaj Kanwar. Bfloat16: The secret to high performance on cloud tpus, 2019. URL https://cloud.google.com/blog/products/ai-machine-learning/bfloat16-the-secret-to-high-performance-on-cloud-tpus. [Online; accessed 28-September-2021].

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771, 2019. URL http://arxiv.org/abs/1910.03771.

Hongyi Zhang, Yann N. Dauphin, and Tengyu Ma. Residual learning without normalization via better initialization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1gsz30cKX.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. Modifying memories in transformer models, 2020. URL https://arxiv.org/abs/2012.00363.

A EFFECTIVE INITIALIZATION AND NORMALIZATION FOR MEND NETWORKS

Although random weight initialization is effective in many settings, it sacrifices the prior that the raw fine-tuning gradient is a useful starting point for editing. Our ablations show that it also leads to less effective edits. For this reason, we initialize MEND to the identity function using a residual connection (He et al., 2016) and a partially random, partially zero-initialization strategy related to Fixup (Zhang et al., 2019). Referring back to Eqs. 3a,b, U_1 and U_2 are initialized with zeros, and V_1 and V_2 use standard Xavier uniform initialization (Glorot and Bengio, 2010) (also see Figure 2). Beyond the initialization, input scaling also presents a challenge: inputs to a MEND network (u_ℓ and $\delta_{\ell+1}$) can differ in magnitude by several orders of magnitude. This poor conditioning causes training to be slow and edit performance to suffer (see Section 5.4). Input normalization addresses this issue; we normalize each dimension of both u_ℓ and $\delta_{\ell+1}$. The input to g_ℓ is the concatenation of $\bar{u}_\ell = \text{norm}(u_\ell)$ and $\bar{\delta}_{\ell+1} = \text{norm}(\delta_{\ell+1})$, where $\bar{u}_\ell$ and $\bar{\delta}_{\ell+1}$ are normalized to have zero mean and unit variance, with means and variances computed over the edit train set and the sequence index.

B EXTENDED DISCUSSION OF RELATED WORK

Model editing shares with continual learning (McCloskey and Cohen, 1989; Parisi et al., 2019) the goal of assimilating or updating a model's behavior without forgetting old information or behaviors, commonly known as the problem of catastrophic forgetting (McCloskey and Cohen, 1989; Ratcliff, 1990; Kirkpatrick et al., 2017). However, in continual learning settings, a model is typically expected to learn wholly new behaviors or datasets (Kirkpatrick et al., 2017; Parisi et al., 2019) without forgetting, while in this work we consider more localized model edits. Further, continual learning generally considers long sequences of model updates with minimal memory overhead, while our work generally considers an edit or batch of edits applied all at once.

Additionally, min-norm parameter fine-tuning has also been considered in past work in the context of editing (Zhu et al., 2020) and traditional model fine-tuning (Guo et al., 2021), where the parameters of the edited or fine-tuned model θ' are penalized (or constrained) from drifting too far from the original model parameters θ using various norms, including L0, L2, and L-∞. While min-norm constraints may be an effective regularization for traditional fine-tuning settings where fine-tuning data is abundant, the experiments conducted in De Cao et al. (2021) show that parameter-space norm constraints are insufficient constraints to prevent significant model degradation when fine-tuning on a single edit example.

B.1 EDITABLE NEURAL NETWORKS (ENN)

Editable neural networks (Sinitsin et al., 2020) search for a set of model parameters that both provide good performance for a 'base task' (e.g., image classification or machine translation) and enable rapid editing by gradient descent to update the model's predictions for a set of 'edit examples' without changing the model's behavior for unrelated inputs. ENN optimizes the following objective, based on the MAML algorithm (Finn et al., 2017):

$$\mathcal{L}_{\text{ENN}}(\theta, \mathcal{D}_{\text{base}}, \mathcal{D}_{\text{edit}}, \mathcal{D}_{\text{loc}}) = L_{\text{base}}(\mathcal{D}_{\text{base}}, \theta) + c_{\text{edit}} \cdot L_{\text{edit}}(\mathcal{D}_{\text{edit}}, \theta') + c_{\text{loc}} \cdot L_{\text{loc}}(\mathcal{D}_{\text{loc}}, \theta, \theta'). \quad (5)$$

The first term of Equation 5 is the base task loss; for a generative language model, we have $L_{\text{base}}(\mathcal{D}_{\text{base}}, \theta) = -\log p_\theta(\mathcal{D}_{\text{base}})$ where $\mathcal{D}_{\text{base}}$ is a batch of training sequences. L_{base} is the edit *reliability* loss, encouraging the model to significantly change its output for the edit examples in $\mathcal{D}_{\text{edit}}$. Finally, L_{loc} is the edit *locality* loss, which penalizes the edited model θ' for deviating from the predictions of the pre-edit model θ on $\mathcal{D}_{\text{loc}}$, data unrelated to $\mathcal{D}_{\text{edit}}$ and sampled from the same distribution as $\mathcal{D}_{\text{base}}$. See Sinitsin et al. (2020) for a more detailed explanation of ENN training and alternative objectives for L_{edit} and L_{loc}.

Comparing ENN and MEND. The key conceptual distinction between ENN and MEND is that ENN encodes editability into the parameters of the model itself (*intrinsic editability*), while MEND provides editability through a set of learned parameters that are independent from the model parameters (*extrinsic editability*). An advantage of ENN is that no new parameters are added in order to provide editability. However, this approach comes with several drawbacks. First, the MAML-based objective ENN optimizes is expensive, particularly in terms of memory consumption (see Figure 4). By further training the model parameters themselves, ENN cannot guarantee that the editable model it produces will make the same predictions as the original model. In order