Thank you to the reviewers for the feedback. There were three main suggestions: additional comparison to other approaches, experiments varying the settings of MUiR, and experiments with a highly-tuned baseline. These concerns are addressed below with additional experiments that were run for this rebuttal and confirm the advantages of our approach. These experiments are described in the first three paragraph below, and are complemented by responses to specific reviewer comments in the rest of the response.

Reviewers 2 and 4 were interested in comparisons to other deep multi-task learning (DMTL) methods in the cross-modality problem in this paper, even if their application seems unnatural for this problem. So, experiments in the three-domain setting were run using *classical* DMTL (e.g., [1, 2, 10]), i.e., where aligned parameters are shared exactly across tasks, and *parallel adapters* [8], an approach mentioned by **Reviewer 3**, which is state-of-the-art (SoA) for vision MTL. Both of these methods require a hierarchical alignment of parameters across architectures. Here, the most natural hierarchical alignment is used, based on a topological sort of the block locations within each architecture: the ith location uses the ith parameter block. MUiR outperforms the existing methods (Table 1). Interestingly, the existing methods each outperforms single task learning (STL) on two out of three tasks. This result shows the value of the universal decomposition in Section 3.1, even when used with other DMTL approaches.

As suggested by **Reviewer 2**, an additional experiment of MUiR was run with different initialization. In this experiment, the module mapping is initialized with the hierarchical alignment used by the other methods above, instead of using the separate initialization suggested by the theory in the paper. This method (Table 1: MUiR+Hierarchical Init.) still outperforms the previous methods on all tasks, but may be better or worse than MUiR for a given task. This result confirms the value of MUiR as a framework, and indicates that exploring initialization schemes is a promising area of future work. Additional exploration of experimental settings will be included in the final version of the paper. The design decisions of MUiR were intended to be the simplest solutions given the requirements of the theory, and it is expected that many such future innovations are possible that would improve the system.

Reviewer 2 was interested in how MUiR would perform in recent highly-tuned training setups for Wikitext-2, e.g., AWD-LSTM [5]. Experiments were run based on the official AWD-LSTM implementation, directly using the many training parameters provided there, i.e., they are tuned to AWD-LSTM, not MUiR. The parameters of MUiR were exactly those used in the other cross-domain experiments. Table 2 shows the results. MUiR achieves performance comparable to STL, while reducing the number of LSTM parameters from 19.8M to 8.8M during optimization. In addition, MUiR outperforms STL with the same number of parameters (i.e., with a reduced LSTM hidden size). These results show that MUiR supports efficient parameter sharing, even when dropped off-the-shelf into highly-tuned setups. The final version of the paper will include results on AWD-LSTM with tuning of MUiR.

Table 1: Comparison to other methods and alternative initialization.

Method	LeNet	Stacked LSTM	DeepBind
Single Task Learning	21.46	135.03	0.1543
Classical DMTL (e.g., [1, 2, 10])	21.09	145.88	0.1519
Parallel Adapters [8]	21.05	132.02	0.1600
MUiR + Hierarchical Init.	**20.72**	**128.94**	**0.1465**
MUiR	**20.51**	**130.70**	**0.1464**

Table 2: Results on Wikitext-2 with AWD-LSTM [5].

Method	LSTM Parameters	Perplexity
Single Task Learning	8.8M	73.64
MUiR	8.8M	71.01
Single Task Learning	19.8M	69.94

Reviewer 2 was interested in a comparison to MultiModel [3]. MultiModel does not address the question of how to parameterize a given set of architectures. It also has gaps with the SoA, and only reports a subset of performance results, which is understandable, since the cross-domain problem is so challenging (as **Reviewer 3** notes). Overall, MultiModel seems like a promising orthogonal direction, and a comparison is not relevant at this time, though it may be relevant in the future if the approaches converge.

Reviewer 3 asked about connections to sequential and parallel adapters [8]: As Table 1 shows, the value of such methods can generalize beyond vision, although they are quite compact, and not theoretically as flexible as hypermodules. **Reviewer 3** asked about more details for when tasks must be learned in a sequence [7]: The most natural approach is to initialize the module set for a new task with existing modules, coupled with a method for preventing forgetting. We will expand on these points in the final version.

Beyond the experimental comparisons above, **Reviewer 4** asked about other theoretical comparisons to previous work on automatic design of MTL models. We will expand on the following in the paper: Learning the alignment with soft ordering [6] yields a quadratic increase in module operations, which is infeasible; Sampling from the softmax instead would still require thousands of additional parameters per module location; The complexity of CTR [4] is shown to be infeasible via Theorem 3.1; Existing approaches use at most 4 [6], 4 [4], and 10 [9] modules, resp., several orders of magnitude fewer than what is considered here, i.e., the cross-domain experiments in the paper use more than 10K modules, and the AWD-LSTM experiment uses more than 60K.

[1] D. Dong, H. Wu, W. He, D. Yu, and H. Wang. Multi-task learning for multiple language translation. In *Proc. of ACL*, pages 1723–1732, 2015.
[2] Z. Huang, J. Li, S. M. Siniscalchi, et al. Rapid adaptation for deep neural networks through multi-task learning. In *Proc. of Interspeech*, pages 3625–3629, 2015.
[3] L. Kaiser, A. N. Gomez, N. Shazeer, A. Vaswani, N. Parmar, L. Jones, and J. Uszkoreit. One model to learn them all. *CoRR*, abs/1706.05137, 2017.
[4] J. Liang, E. Meyerson, and R. Miikkulainen. Evolutionary architecture search for deep multitask networks. In *Proc. of GECCO*, 2018.
[5] S. Merity, N. S. Keskar, and R. Socher. Regularizing and optimizing LSTM language models. In *Proc. of ICLR*, 2018.
[6] E. Meyerson and R. Miikkulainen. Beyond shared hierarchies: Deep multitask learning through soft layer ordering. In *Proc. of ICLR*, 2018.
[7] S.-A. Rebuffi, H. Bilen, and A. Vedaldi. Learning multiple visual domains with residual adapters. In *NIPS*, pages 506–516. 2017.
[8] S.-A. Rebuffi, H. Bilen, and A. Vedaldi. Efficient parametrization of multi-domain deep neural networks. In *Proc. of CVPR*, pages 8119–8127, 2018.
[9] C. Rosenbaum, T. Klinger, and M. Riemer. Routing networks: Adaptive selection of non-linear functions for multi-task learning. In *Proc. of ICLR*, 2018.
[10] Z. Zhang, L. Ping, L. C. Chen, and T. Xiaoou. Facial landmark detection by deep multi-task learning. In *Proc. of ECCV*, pages 94–108, 2014.

* For time constraints, all experimental results in tables were capped to 200 epochs.