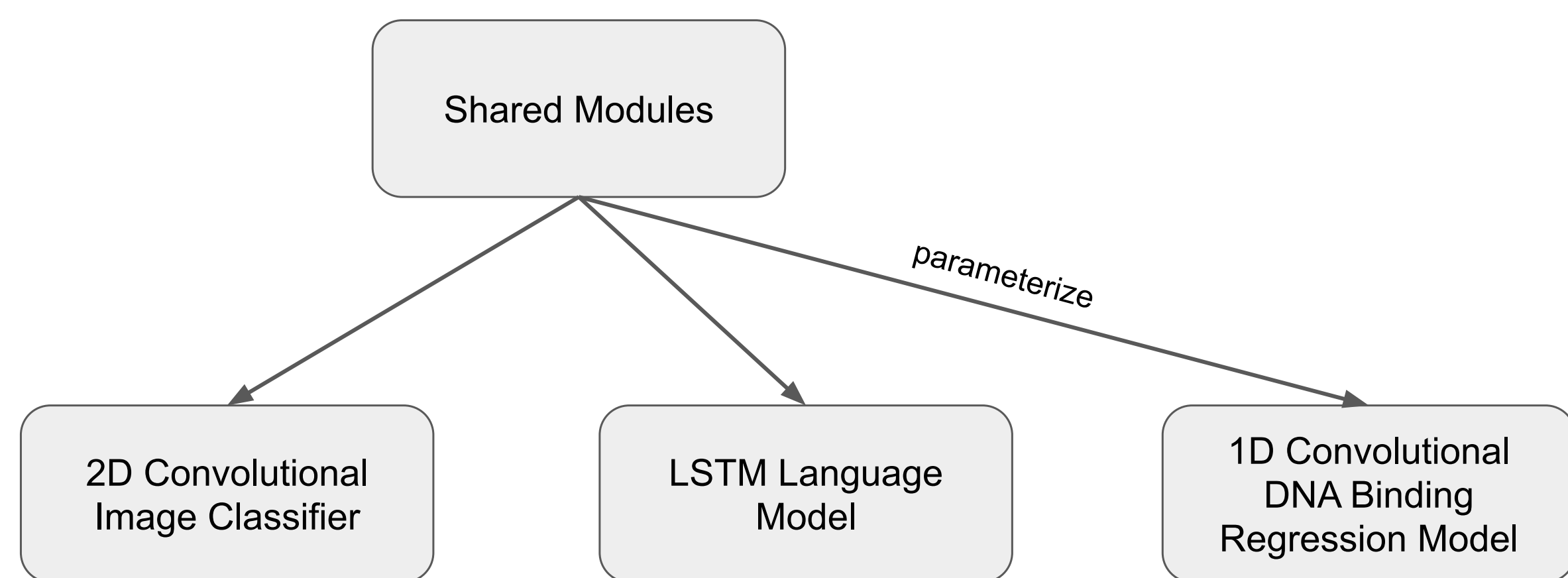


## Overview

- Deep learning applications continue to become more diverse; they benefit from *methodological sharing*.
- Beyond shared methods, can general problem solving arise from *learned sharing* across such diverse tasks?
- Consider the setting where there is no obvious overlap between task architectures. A first approach (MUIr):
  - Decompose any set of (architecture,task) pairs into potentially related subproblems;
  - Optimize their sharing with an efficient stochastic algorithm.
- Results confirm that sharing learned functionality across diverse domains and architectures is beneficial.



## Problem Statement

Given an arbitrary set of (architecture,task) pairs, can learned functionality be shared across architectures to improve performance in each task?

## Solution Requirements and Related Work

- A solution must support any given set of architectures;
- A solution must automatically *align* parameters across the architectures.

Existing deep multi-task learning (DMTL) methods do not satisfy both conditions:

- Classical [5, 6, 11, 12, 36] and factorization methods [20, 27, 32] do not meet condition (2);
- Existing module assembly approaches [19, 22, 25, 28] do not satisfy (1);
- Designing a new customized architecture [4, 10, 14] satisfies neither (1) nor (2).

## Reparameterizing Architectures with Hypermodules

Given  $T$  tasks  $\{\{\mathbf{x}_{ti}, \mathbf{y}_{ti}\}_{i=1}^{N_t}\}_{t=1}^T$ , with architectures  $\{\mathcal{M}_t\}_{t=1}^T$ , with parameters  $\theta_{\mathcal{M}_t}$ . The goal of MTL is to find

$$\bigcup_{t=1}^T \theta_{\mathcal{M}_t}^* = \underset{\bigcup_{t=1}^T \theta_{\mathcal{M}_t}}{\operatorname{argmin}} \frac{1}{T} \sum_{t=1}^T \frac{1}{N_t} \sum_{i=1}^{N_t} \mathcal{L}_t(\mathbf{y}_{ti}, \hat{\mathbf{y}}_{ti}), \quad (1)$$

If each tensor in each  $\theta_{\mathcal{M}_t}$  can be decomposed into equally-sized parameter blocks  $\mathbf{B}_\ell \in \mathbb{R}^{m \times n}$ , we can write

$$\bigcup_{t=1}^T \theta_{\mathcal{M}_t} = (\mathbf{B}_1, \dots, \mathbf{B}_L). \quad (2)$$

Parameter tensors in practice can be decomposed in this way so that each  $\mathbf{B}_\ell$  induces a linear pseudo-task [23].

To share structure, the  $\mathbf{B}_\ell$  are parameterized by  $K < L$  hypernetworks  $\{\mathbf{H}_k\}_{k=1}^K$  [9, 29] and contexts  $\{\mathbf{z}_\ell\}_{\ell=1}^L$ :

$$\bigcup_{t=1}^T \theta_{\mathcal{M}_t} = [(\mathbf{H}_1, \dots, \mathbf{H}_K), (\mathbf{z}_1, \dots, \mathbf{z}_L)]. \quad (3)$$

In this paper,  $\mathbf{B}_\ell = \psi(\ell) \bar{\mathbf{x}}_1 \mathbf{z}_\ell$ , where  $\psi: \{1, \dots, L\} \rightarrow \{\mathbf{H}_k\}_{k=1}^K$  is the alignment function.

## Efficiently Optimizing the Mapping of Modules to Blocks

Subalignments  $\{\psi_d\}_{d=1}^D$  of  $\psi$  can be optimized in parallel, if each has a distinct evaluation function  $h_d$ .

For simplicity, let each be optimized by a  $(1+\lambda)$ -EA [7, 24, 30], a component of previous DMTL methods [19, 23].

**Theorem** The expected time of the decomposed  $K$ -valued  $(1+\lambda)$ -EA is  $O\left(\frac{KL(\log L - \log D) \log D}{D}\right)$ , when all  $h_d$  are linear.

Decomposition Level	None (Multi-task)	Per-task	Per-block
Expected Convergence Time	$O(KL \log L)$	$O\left(\frac{KL(\log L - \log T) \log T}{T}\right)$	$O(K \log L)$

Biasing optimization towards more frequently-used hypermodules addresses the bottleneck of large  $K$ :

**Theorem** The expected time of the decomposed  $K$ -valued  $(1+\lambda)$ -EA with pessimistic initialization and proportional sampling is  $O(\log L)$ , when  $D = L$ , and all  $h_d$  are linear.

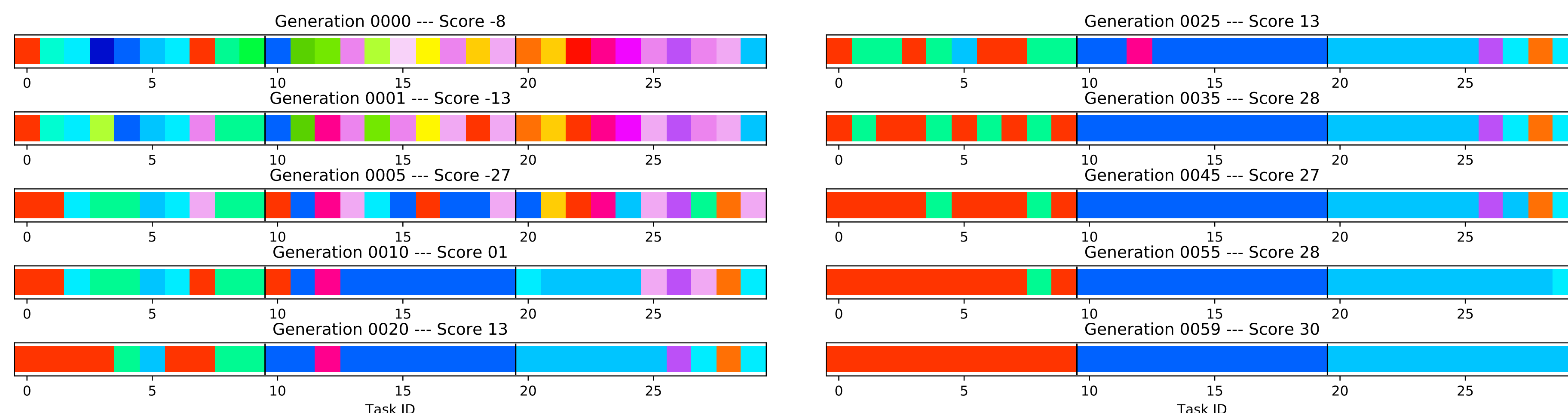
To achieve per-block decomposition, MUIr interleaves iterations of  $(1+\lambda)$ -EA with backpropagation via

$$\mathbf{B}_\ell = \sum_{i=0}^{\lambda} \psi^i(\ell) \bar{\mathbf{x}}_1 \mathbf{z}_\ell \cdot \operatorname{softmax}(\mathbf{s}_\ell)_i, \quad (4)$$

where the learned  $\operatorname{softmax}(\mathbf{s}_\ell)_i$  indicates the model's belief that  $\psi^i(\ell) \approx h_d$  is the best choice for the  $\ell$ th block.

## Validation on a Classic Synthetic Dataset

The data set has 30 tasks with well-defined task-grouping [15]. MUIr quickly converges to the optimal mapping:



- $h_d$  is compared against random and oracle evaluation functions.
- MUIr achieves optimal RMSE in the clean setting.
- MUIr on par with best reported results in noisy setting, despite increased problem difficulty due to
  - data withheld for validation,
  - absence of additional regularization,
  - having to learn the number of groups automatically.

Method	Clean	Noisy
STL [15]	-	0.97
MTL-FEAT [3]	-	0.48
DG-MTL [15]	-	0.42
GO-MTL [17]	-	<b>0.35</b>
STL (ours)	$1.35 \pm 0.01$	$1.49 \pm 0.01$
MUIr + Random	$1.26 \pm 0.04$	$4.67 \pm 1.48$
MUIr + Oracle	$0.77 \pm 0.77$	<b><math>0.37 \pm 0.00</math></b>
MUIr + Optimization	<b><math>0.00 \pm 0.00</math></b>	<b><math>0.38 \pm 0.00</math></b>

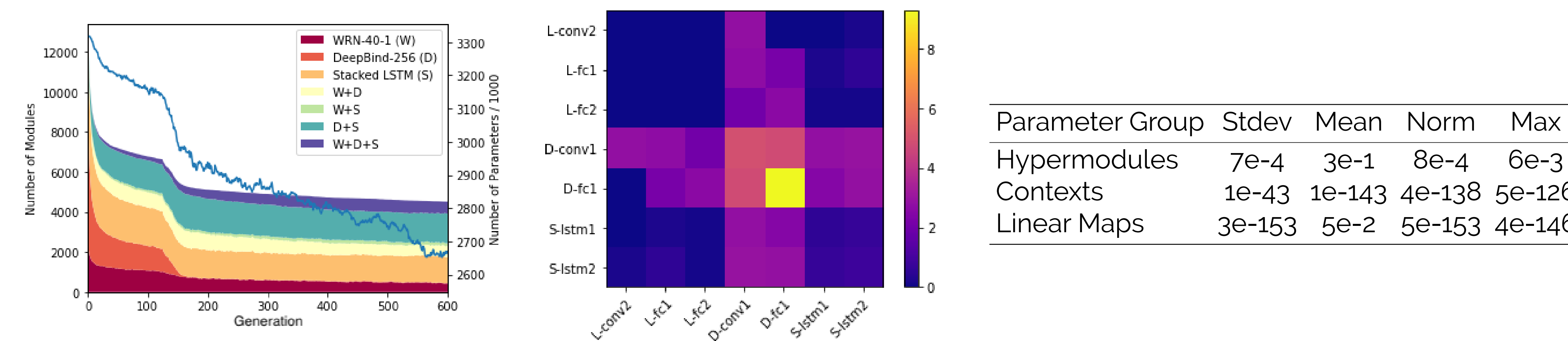
## Experiments with Diverse Architectures and Modalities

Cross-modal experiments were run with CIFAR-10 [16], WikiText-2 [21], and CRISPR DNA Binding Prediction [13]:

Modality	Architecture	Metric	Baseline	Intrask	W+S	W+D	S+D	W+S+D	L+S	L+D	L+S+D
Vision	WRN-40-1 (W) [33]	Classificatin Error	<b>8.48</b>	<b>8.50</b>	8.69	9.20	-	9.02	-	-	-
Text	Stacked LSTM (S) [34]	Perplexity	134.41	132.06	130.63	-	132.62	<b>128.10</b>	<b>129.73</b>	-	130.77
DNA	DeepBind-256 (D) [2, 35]	Mean Squared Error	0.1540	0.1466	-	<b>0.1461</b>	0.1469	<b>0.1464</b>	-	0.1469	<b>0.1464</b>
Vision	LeNet (L) [18]	Classification Error	21.08	20.67	-	-	-	-	21.02	<b>19.59</b>	<b>20.23</b>

- MUIr always improves the text and genomics models, especially when they share with WRN.
- For L+S+D, the improvement due to MUIr is significant for all tasks.
- WRN behavior suggests the power of MUIr comes from diversity and identifies a key optimization challenge.

## Analysis of Cross-modal Module Sharing Dynamics



- Left*. The proportion of modules shared exclusively by each subset of tasks is architecture-dependent.
- Mid*. W.r.t. sharing rate, the 1D-Conv model plays a central role between the 2D-Conv and 1D-LSTM models.
- Right*.  $p$ -values for differences in Generic vs. Specific tensor statistics; The Generic show better generalizability.

## Ablation and Comparison to other DMTL Methods

Method	LeNet	Stacked LSTM	DeepBind	$\dim(\mathbf{z}_\ell)$	LeNet	Stacked LSTM	DeepBind
Single Task Learning	21.46	135.03	0.1543	0	21.89	144.52	0.1508
Classical DMTL (e.g. [8, 12, 36])	21.09	145.88	0.1519	1	21.80	140.94	0.1477
Parallel Adapters [27]	21.05	132.02	0.1600	2	<b>20.40</b>	133.94	0.1504
MUIr + Hierarchical Init.	<b>20.72</b>	<b>128.94</b>	<b>0.1465</b>	4	<b>20.51</b>	<b>130.70</b>	<b>0.1464</b>
MUIr	<b>20.51</b>	<b>130.70</b>	<b>0.1464</b>	8	20.62	<b>130.80</b>	<b>0.1468</b>

- Left*. Cross-modal comparisons were run with existing DMTL methods using Eq. 2.
  - The required hierarchical alignment was implemented by a topological sort of blocks in each architecture.
  - Even this rough alignment leads to improvement over STL, showing the value of Eq. 2.
  - Initializing  $\psi$  with this alignment leads to similar results for MUIr, showing robustness to design choices.
- Right*. Experiments with different context sizes  $\dim(\mathbf{z}_\ell)$  show the value of hypermodules over hard sharing.

## Discussion and Future Work

- MUIr shows that functionality can be effectively shared in diverse settings. Many improvements are possible:
  - more sophisticated factorization than hypermodules [20, 32]; more sophisticated optimization than  $(1+\lambda)$ -EA.
- For lifelong learning [1, 26, 31], modules can be collected, refined, and assembled to solve tasks as they appear.

## References

- [1] D. Abel, D. Arumugam, L. Lehnert, and M. Littman. State abstractions for lifelong reinforcement learning. In *Proc. of ICML*, 2018.
- [2] B. Alipanahi, A. Delong, M. T. Weierich, and B. J. Frey. Predicting the sequence specificities of dna-and mna-binding proteins by deep learning. *Nature biotechnology*, 2015.
- [3] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 2008.
- [4] B. Bilen and A. Vedaldi. Integrated perception with recurrent multi-task neural networks. In *NIPS*, 2016.
- [5] R. Caruana. Multitask learning. In *Learning to learn*, pages 95–133. Springer US, 1998.
- [6] C. Devin, A. Gupta, T. Darrell, P. Abbeel, and S. Levine. Learning modular neural network policies for multi-task and multi-robot transfer. In *Proc. of ICRA*, 2017.
- [7] B. Doerr, T. Jansen, and C. Klein. Comparing global and local mutations on bit strings. In *Proc. of GECCO*, 2008.
- [8] 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.
- [9] D. Ha, A. M. Dai, and Q. V. Le. Hypernetworks. In *Proc. of ICLR*, 2017.
- [10] K. Hashimoto, C. Xiong, Y. Tsuruoka, and R. Socher. A joint many-task model: Growing a neural network for multiple NLP tasks. In *Proc. of EMNLP*, 2017.
- [11] J. T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong. Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. In *Proc. of ICASSP*, 2012.
- [12] Z. Huang, J. Li, S. M. Srinivasan, et al. Rapid adaptation for deep neural networks through multi-task learning. In *Proc. of Interspeech*, 2015.
- [13] C. Jung, J. A. Hawkins, S. K. Jones, et al. Massively parallel biophysical analysis of crispr-cas complexes on next generation sequencing chips. *Cell*, 2017.
- [14] L. Kaiser, A. N. Gomez, N. Shazeer, A. Vaswani, N. Parmar, L. Jones, and J. Uszkoreit. One model to learn them all. *arXiv*, 2017.
- [15] Z. Kang, K. Grauman, and F. Sha. Learning with whom to share in multi-task feature learning. In *Proc. of ICML*, 2011.
- [16] A. Krizhevsky. *Learning Multiple Layers of Features from Tiny Images*, 2009.
- [17] A. Kumar and H. Daumé. III. Learning task grouping and overlap in multi-task learning. In *Proc. of ICML*, 2012.
- [18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 1998.
- [19] J. Liang, E. Meyerson, and R. Miikkulainen. Evolutionary architecture search for deep multitask networks. In *Proc. of GECCO*, 2018.
- [20] M. Long, Z. Cao, J. Wang, and P. S. Yu. Learning multiple tasks with multilinear relationship networks. In *NIPS*, 2017.
- [21] S. Merity, C. Xiong, J. Bradbury, and R. Socher. Pointer sentinel mixture models. *arXiv*, 2016.
- [22] E. Meyerson and R. Miikkulainen. Beyond shared hierarchies: Deep multitask learning through soft layer ordering. In *Proc. of ICLR*, 2018.
- [23] E. Meyerson and R. Miikkulainen. Pseudo-task augmentation: From deep multitask learning to multitask sharing—and back. In *Proc. of ICML*, 2018.
- [24] F. Neumann and C. Witt. On the runtime of randomized local search and simple evolutionary algorithms for dynamic makespan scheduling. In *Proc. of UCAI*, 2015.
- [25] P. Ramachandran and Q. V. Le. Diversity and depth in per-example routing models. In *Proc. of ICLR*, 2019.
- [26] S.-A. Rebuffi, B. Bilen, and A. Vedaldi. Learning multiple visual domains with residual adapters. In *NIPS*, pages 506–516, 2017.
- [27] S.-A. Rebuffi, B. Bilen, and A. Vedaldi. Efficient parametrization of multi-domain deep neural networks. In *Proc. of CVPR*, 2018.
- [28] C. Rosenbaum, T. Klöpper, and M. Riemer. Routing networks: Adaptive selection of non-linear functions for multi-task learning. In *Proc. of ICLR*, 2018.
- [29] K. O. Stanley, D. B. D'Ambrosio, and J. Gauci. A hypercube-based encoding for evolving large-scale neural networks. *ALife*, 2009.
- [30] D. Sudholt. On the robustness of evolutionary algorithms to noise: Refined results and an example where noise helps. In *Proc. of GECCO*, 2018.
- [31] C. Jung, J. A. Hawkins, S. K. Jones, et al. Rapid adaptation for deep neural networks through multi-task learning. In *Proc. of Interspeech*, 2015.
- [32] Y. Yang and T. Hospedales. Deep multi-task representation learning: A tensor factorisation approach. In *Proc. of ICLR*, 2017.
- [33] S. Zagoruyko and N. Komodakis. Wide residual networks. *arXiv*, 2016.
- [34] W. Zarembka, I. Sutskever, and O. Vinyals. Recurrent neural network regularization. *arXiv*, 2014.
- [35] H. Zeng, M. D. Edwards, G. Liu, and D. K. Gifford. Convolutional neural network architectures for predicting dna-protein binding. *Bioinformatics*, 2016.
- [36] Z. Zhang, L. Ping, L. C. Chen, and T. Xiaoou. Facial landmark detection by deep multi-task learning. In *Proc. of ECCV*, 2014.