# A Appendix

## A.1 Additional Results

More detailed numbers on the T0 Sanh et al. [2022] and SuperNI Wang et al. [2022a] datasets using different backbones, and different adapter layouts over the base model are found in Table 4. `Multi-Task params` is the number of additional parameters that must be conserved

| **Model** | Multi-Task Params | Adaptation Params | Avg. Test |
|---|---|---|---|
| **T0 Dataset** | | | |
| *Backbone T5-XL-LM* | | | |
| `Multi-Task Full Finetuning + LoRA` | 2.8B | 2.2M | $68.9_{x.x}$ |
| `(IA)`$^3$ | 540K | 540K | $62.4_{0.4}$ |
| `AdapterSoup` | 84M | 2.2M | $62.1_{1.0}$ |
| `LoRA` | 2.2M | 2.2M | $66.0_{1.6}$ |
| `LoRA-big` | 35M | 35M | $65.4_{0.9}$ |
| `Poly-`$z$ | 17M | 3.5K | $66.4_{0.3}$ |
| `Poly` | 17M | 2.2M | $68.0_{1.0}$ |
| `MHR-`$z$ *(64 h)* | 17M | 220K | $68.3_{0.8}$ |
| `MHR` *(64 h)* | 17M | 2.2M | $\underline{69.1}_{1.0}$ |
| *Backbone T0-3B* | | | |
| `T-Few` Liu et al. [2022] | 540K | 540K | $66.2_{0.5}$ |
| `AdapterSoup` | 84M | 2.2M | $66.1_{0.6}$ |
| `LoRA` | 2.2M | 2.2M | $67.4_{0.8}$ |
| `LoRA-big` | 35M | 35M | $68.0_{0.8}$ |
| `Poly-`$z$ | 17M | 3.5K | $65.3_{1.0}$ |
| `Poly` | 17M | 2.2M | $69.0_{0.8}$ |
| `MHR`$z$ *(64 h)* | 17M | 220K | $68.4_{1.2}$ |
| `MHR` *(8 h)* | 17M | 2.2M | $\underline{69.3}_{1.2}$ |
| *Backbone T0-3B light* version : (`k, v, ff` layers only) | | | |
| $l$-`LoRA` *(rank 1)* | 934K | 934K | $66.2_{0.9}$ |
| $l$-`LoRA` *(rank 16)* | 15M | 15M | $67.6_{1.1}$ |
| `AdapterSoup` ($l$-`LoRA`) | 35M | 934K | $64.9_{1.0}$ |
| $l$-`Poly-`$z$ | 7.5M | 2.1K | $62.9_{1.2}$ |
| $l$-`Poly` | 7.5M | 934K | $68.0_{0.5}$ |
| $l$-`MHR`$z$ *(32 h)* | 7.5M | 74K | $66.8_{1.1}$ |
| $l$-`MHR` *(8 h)* | 7.5M | 934K | $\underline{68.5}_{0.7}$ |
| **SuperNI Dataset** | | | Rouge-L |
| *Backbone T5-XL-LM light* version : (`k, v, ff` layers only) | | | |
| $l$-`LoRA` | 934K | 934K | $67.6_{0.8}$ |
| $l$-`LoRA-big` | 18M | 18M | $67.2_{0.7}$ |
| $l$-`Poly-`$z$ | 7.5M | 2.1K | $64.6_{0.3}$ |
| $l$-`Poly` | 7.5M | 934K | $67.8_{0.8}$ |
| $l$-`MHR`$z$ *(64 h)* | 7.5M | 147K | $68.0_{0.2}$ |
| $l$-`MHR` *(8 h)* | 7.5M | 934K | $\underline{68.5}_{0.3}$ |

Table 4: (top) Results on T0 dataset Sanh et al. [2022], we report the mean of the best validation accuracy for each test task, when evaluated every 50 train epochs. `T-Few` is our reproduction of the results in Liu et al. [2022]. `LoRA-big` means a `LoRA` adapter with a larger rank. (bottom) Results on SuperNatural Instructions dataset.

after multi-task pretraining to enable transfer to a downstream task. `Adaptation Params` refer to the number of parameters required to learn a new downstream task. For e.g. `Poly` and `MHR`, the multi-task parameters includes the learned modules, but not the routing over the training tasks, as these are not required for transfer on a new task. Moreover, variants which average the learned modules prior to fine-tuning (`MHR-`$\mu$ and `Poly-`$\mu$) will have both multi-task and adaptation parameters equal to that of a single shared adapter, since after multi-task pretraining one can average the modules.

### A.2 Navigating the parameter efficiency / performance trade-off of tuning only the routing

Here we provide additional results on how different routing based methods can be more expressive when only learning a new routing function (over *frozen* modules) to adapt to a new task.

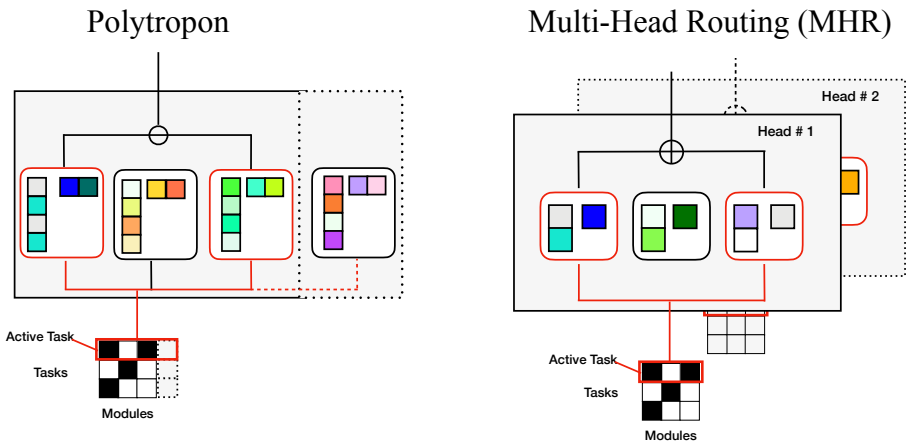


Figure 4: Different ways to control the expressivity of routing based methods. *Left :* In Polytropon, one can only add additional modules, resulting in a linear parameter increase. *Right :* In `MHR`, additional heads only introduce routing matrices $\boldsymbol{Z}$, resulting in a negligible parameter increase.

In Fig. 4 (left), we see that in order to build more expressive routing functions $\boldsymbol{Z}$, in `Poly` one can only do so by increasing the number of skills at each layer. However, this has a significant impact on the number of multi-task parameters which much be kept in order to perform few-shot transfer. `MHR` on the other hand, can increase routing capacity in a much more parameter efficient way.

#### A.2.1 On the granularity of routing tensor in `MHR`

Here we provide additional results when modifying the granularity of $\mathbf{Z}$ for `MHR`. We see that one can easily trade-off more parameters for better performance.

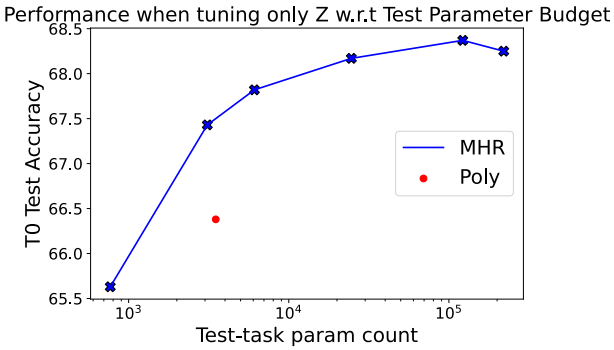


Figure 5: Routing-Only Fine-Tuning (`MHR`-$z$)

## B Broader Impact

In our work, we focus on advancing parameter-efficient fine-tuning methods for cross-task generalization. While our research primarily addresses technical challenges and performance improvements, when applying such methods, it is crucial to consider the potential negative societal impacts. Specifically, we believe that prior to applying our proposed adaptation method, critically examining the potential biases and ethical implications of the underlying large language model, and the data itself must be properly addressed. This includes issues related to fairness, privacy, and the spread of misinformation.