

Team 37

10-623 Generative AI, Fall 2025

DO NOT PUT ANYTHING HERE

TaskWeaver

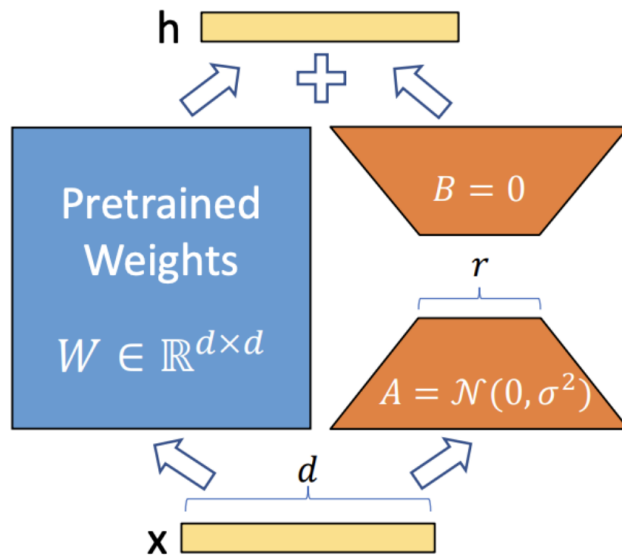
Instance-Level Test-Time Adaptation for Language Models

DO NOT PUT ANYTHING HERE

**Dhruv Kapur, Raj Maheshwari,
Andrews George Varghese**

DO NOT PUT ANYTHING HERE

Low Rank Adaptation



- Hypothesis^[1]: Downstream task adaptation requires low-rank updates to the weight matrices of the SFT LLM
- If ΔW is the required adaptation, break it into low-rank matrices A, B

$$W_0 + \Delta W = W_0 + BA$$

where $B \in \mathbb{R}^{d \times r}, A \in \mathbb{R}^{r \times k}, r \ll \min(d, k)$

...is awesome, but...

- Carefully curate a dataset for every task
😞
 - *No, really! Creating a good dataset is hard!*
- LoRA finetuning still expensive
- Sensitive to hyperparameters
 - *And do you really have enough compute for a multi-day sweep?*

Can we avoid training finetuning for every new task we come across?

^[1]LoRA: Low-Rank Adaptation of Large Language Models, by Hu et al. 2021

<some super secret system prompt for our helpful AI assistant...>

Q: Can we avoid finetuning for every new task we come across? What if we can somehow generate LoRA weights on the fly? For every prompt?! We can train a neural network to do that. For every prompt, we can get tailor made LoRA weights! 😎

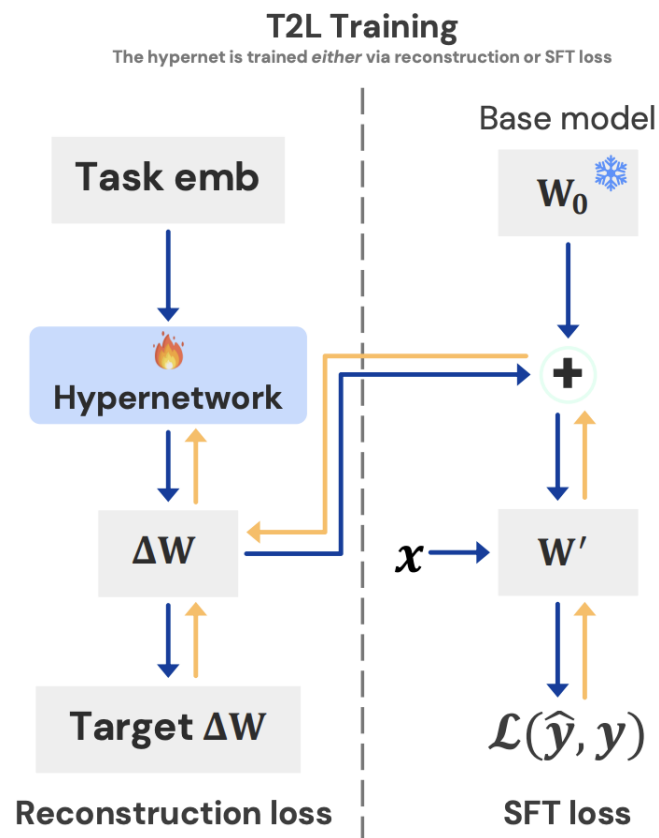
A: Wait a second... Sakana AI has done something similar 😞

Text-to-LoRA^[2]

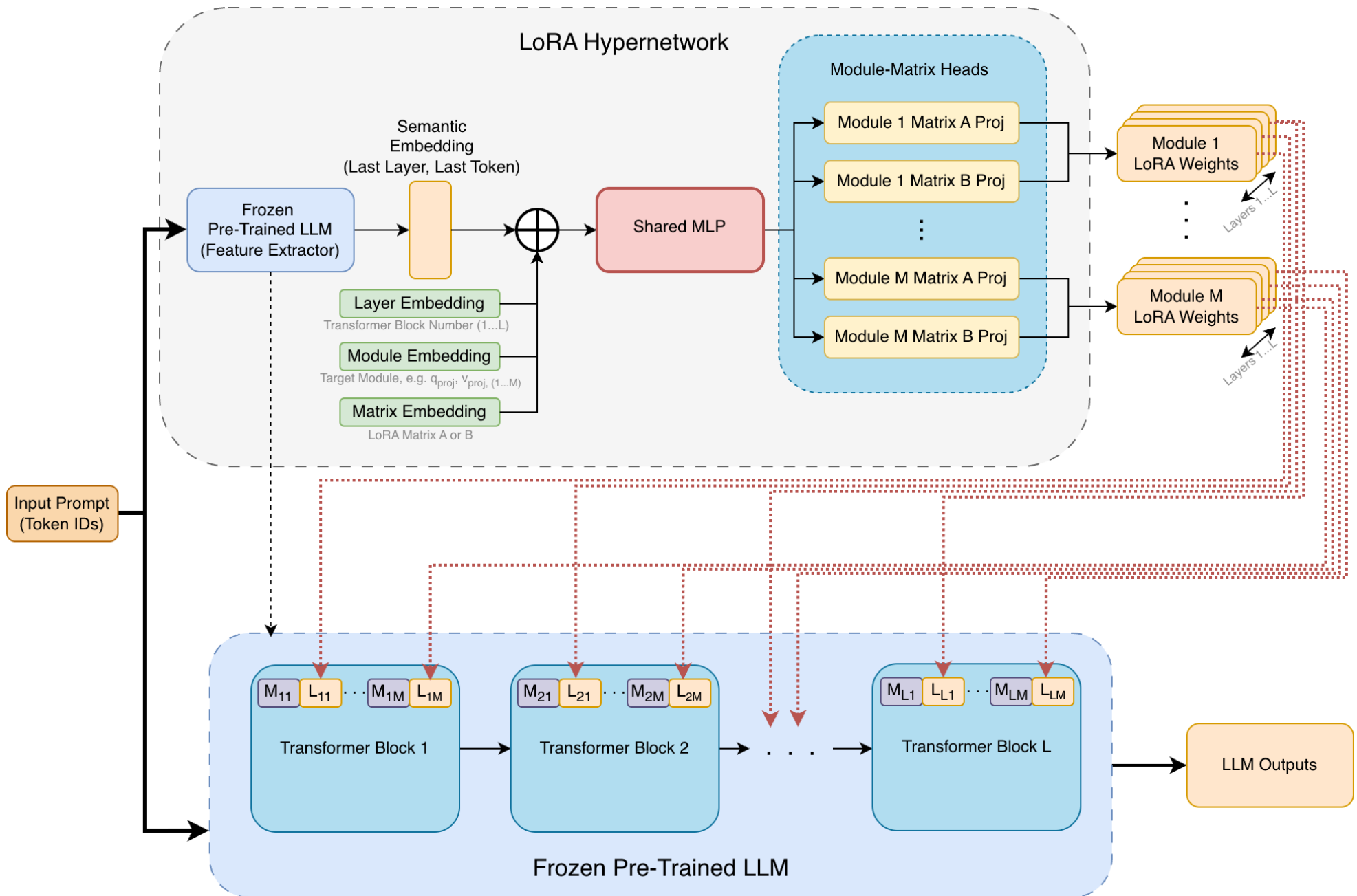
- **Carefully** create a natural language description for each task
- Pass its embedding to a trained hypernetwork which **predicts LoRA weights!**
- Impressive results, *including zero-shot performance*
- Training end to end (SFT) or through reconstruction or a dataset of LoRA weights

Is that the end of our dream?

Not quite.



We want to predict LoRA weights for every prompt, no task-specific embeddings or descriptions. **Truly zero-shot!**



Architecture

- Reuse language model as a prompt encoder
- Layer, Module and Matrix are identified through learned embeddings
- Each Module-Matrix pair has its own output head, shared across layers
- Predicted LoRA weights are injected into LM before forward pass
- **Key Challenge:** How do we train this efficiently? Each batch element requires unique LoRA weights.
- **Idea:** Batch-dependent LoRA weight prediction! $B \times D \times r$ and $B \times r \times D$ matrices
- End-to-end training through minimizing language modelling loss

Datasets

- Limited compute budget
- Avoid tasks that require LLM judges
- How do we achieve maximum coverage?
 - Choose orthogonal datasets

Science: ARC-Easy/Challenge

Math: GSM8K, SVAMP

Logic: SNLI, winogrande, HellaSwag

Comprehension: BoolQA, OpenBookQA, RACE

Common sense: commonsense_qa

Models

Pythia 70M

Gemma3 270M IT

Qwen3 0.6B

Compute

A100 x 1

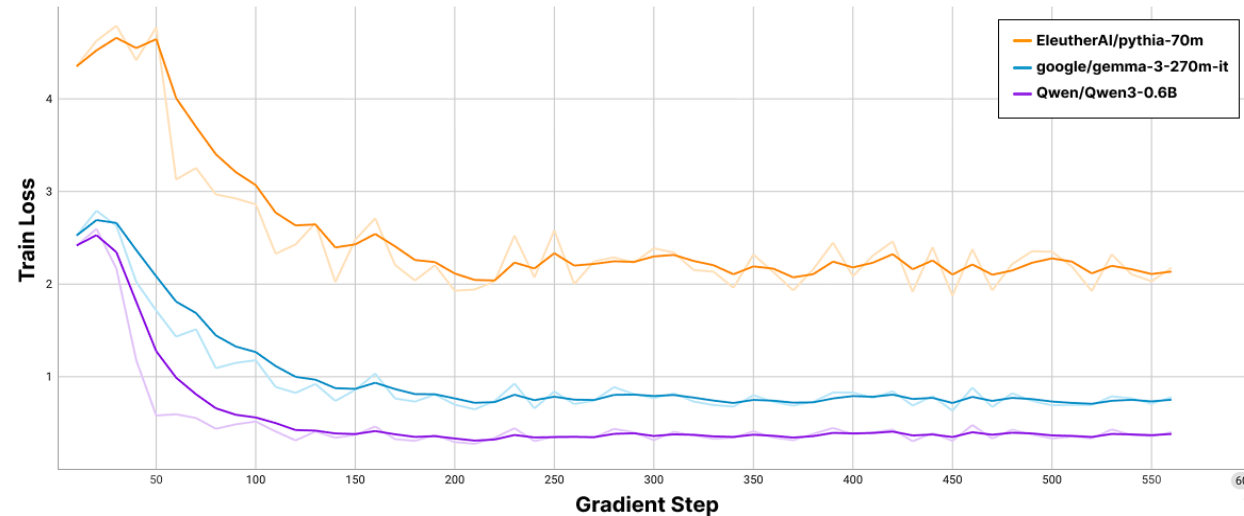
3090 x 1

3070 x 1

M4 Pro x 1

Training

Language Model Size



	Pythia	Gemma	Qwen
Hidden dim	1024	1024	1024
LoRA Rank	2	2	2
LoRA Alpha	8	8	8

	Pythia	Gemma	Qwen
Batch size	16	8	4
Training time	3:06	21:30	1:07:25

Training and Performance Insights

- Larger models have smoother training, converge faster and to a lower loss
- TaskWeaver performs strongest for smaller models. Maybe due to unoptimized hypernet backbone that who's expressiveness may be too small for larger models.
- Compared to LoRA (Mix), TaskWeaver performs better on Math datasets, indicating better tolerance to data skew.
- T-SNE Analysis shows a generally well structured space of predicted LoRA adapters, but these are separate from those achieved through PEFT Training

Results

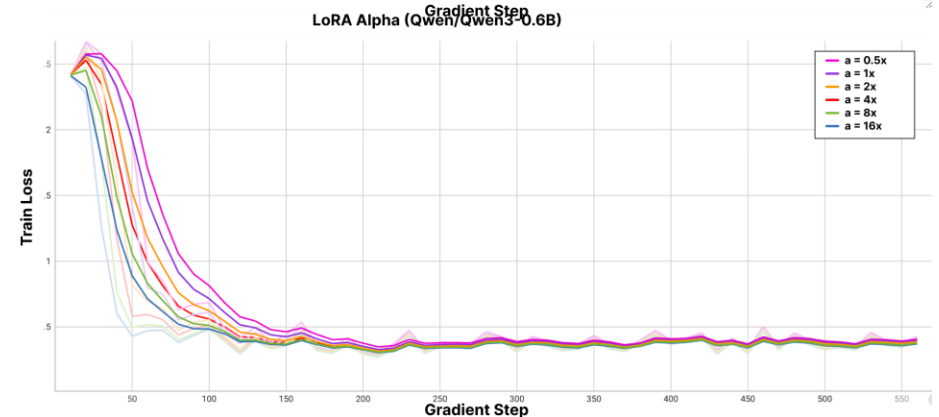
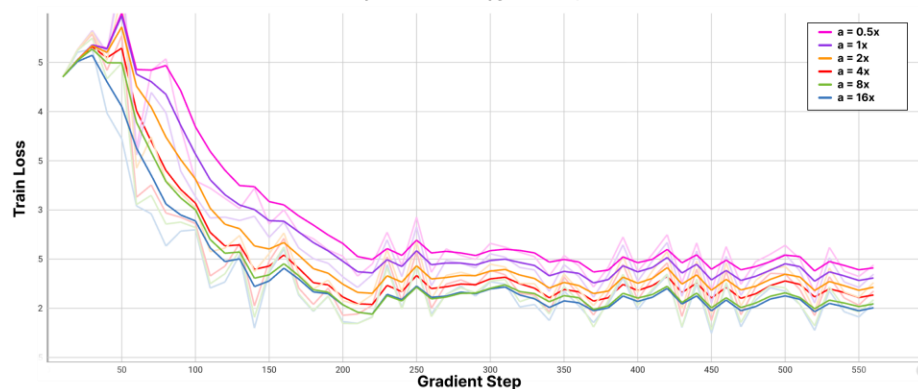
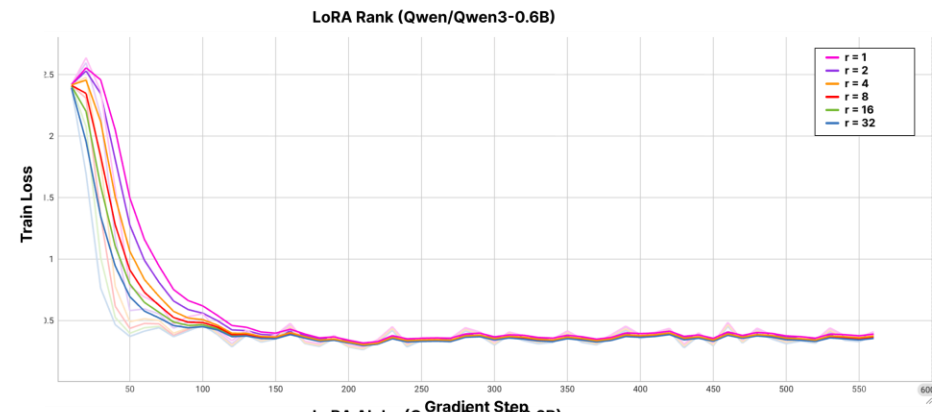
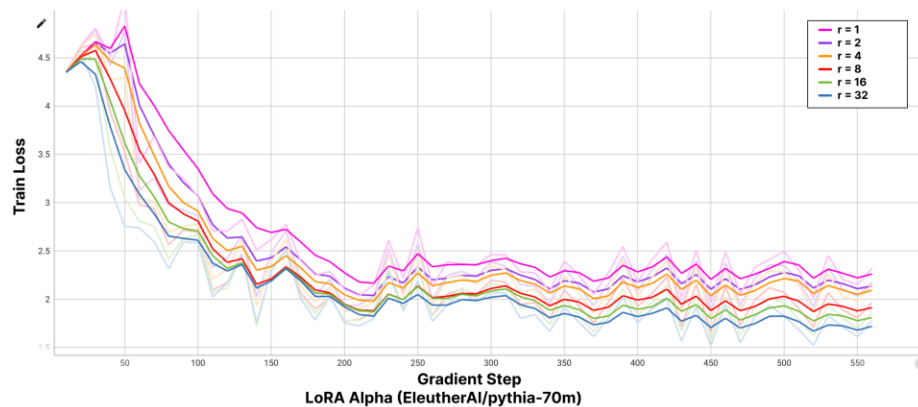
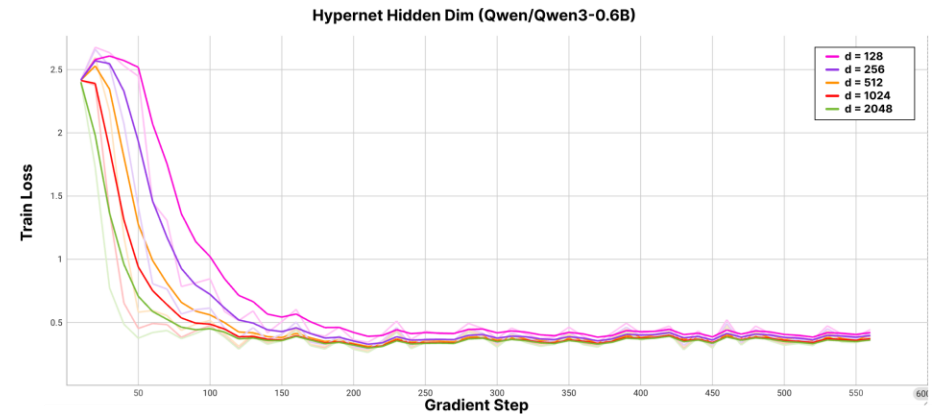
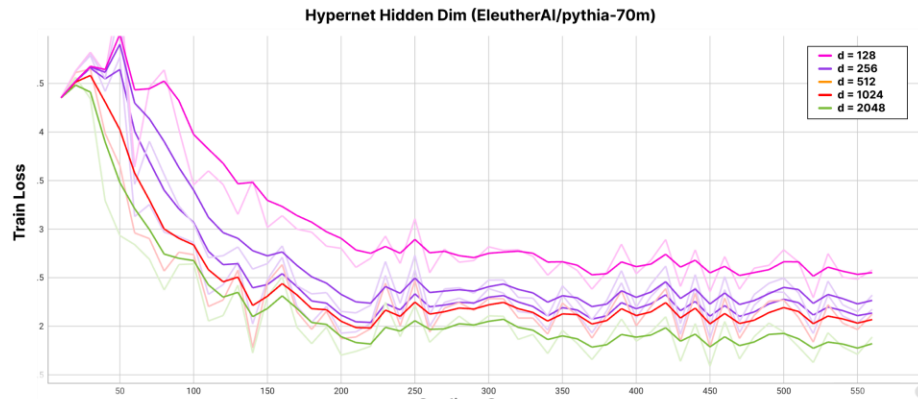
Model	Mode	ARC Challenge	ARC Easy	BoolQ	GMS8K	HELLASWAG	OpenBookQA
EleutherAI/pythia-70m	Base	9.13%	7.07%	14.2%	1.29%	4.94%	8.2%
	LoRA (Individual)	22.8%	23.7%	53.5%	1.14%	25.4%	22.4%
	LoRA (Mixed)	24.1%	22.3%	53.9%	1.97%	24.6%	26.6%
	TaskWeaver (Ours)	24.9%	23.9%	54.5%	1.67%	24.8%	28.6%
google/gemma-3-270m-it	Base	18.9%	23.0%	43.5%	4.55%	24.7%	24.6%
	LoRA (Individual)	25.6%	25.0%	56.4%	3.64%	24.2%	26.4%
	LoRA (Mixed)	25.3%	21.3%	52.0%	3.41%	23.5%	25.2%
	TaskWeaver (Ours)	21.1%	23.7%	45.7%	5.84%	24.5%	27.6%
Qwen/Qwen3-0.6B	Base	21.7%	31.7%	63.6%	50.0%	25.9%	34.2%
	LoRA (Individual)	50.7%	71.8%	78.8%	35.9%	50.7%	62.2%
	LoRA (Mixed)	54.4%	68.8%	68.6%	34.5%	33.0%	53.4%
	TaskWeaver (Ours)	35.2%	48.5%	64.1%	52.5%	29.0%	38.8%

Model	Mode	SNLI	Winogrande (M)	SVAMP*	CommonSenseQA *	RACE (Middle)*
EleutherAI/pythia-70m	Base	0.519%	6.08%	2.33%	3.44%	8.64%
	LoRA (Individual)	34.9%	49.7%	-	-	-
	LoRA (Mixed)	31.0%	44.3%	3.0%	18.3%	18.7%
	TaskWeaver (Ours)	29.5%	48.4%	1.0%	19.7%	25.1%
google/gemma-3-270m-it	Base	34.3%	50.6%	19.3%	18.4%	20.8%
	LoRA (Individual)	42.9%	53.8%	-	-	-
	LoRA (Mixed)	33.3%	50.9%	4.67%	20.6%	22.8%
	TaskWeaver (Ours)	34.7%	47.8%	16.3%	21.5%	23.0%
Qwen/Qwen3-0.6B	Base	42.4%	50.4%	73.0%	38.2%	34.4%
	LoRA (Individual)	84.2%	48.7%	-	-	-
	LoRA (Mixed)	76.9%	48.5%	52.7%	50.9%	64.8%
	TaskWeaver (Ours)	49.2%	52.1%	74.7%	45.9%	44.2%

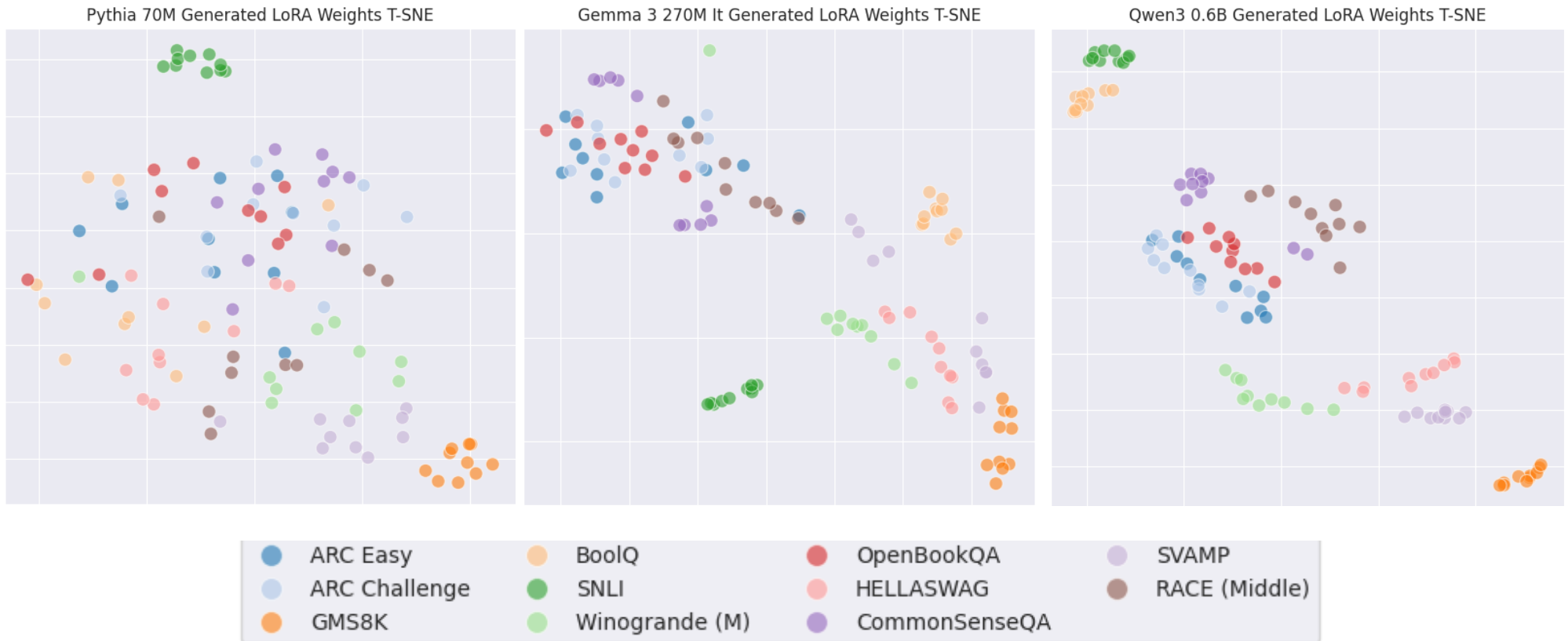
* Zero-shot evaluation

Scaling Analysis

- Larger models are more invariant to hyperparameter change
- Larger models achieve lower training loss with the same amount of data
- Train loss decreases with increase in hypernet dimension, LoRA rank and LoRA alpha



Analysis of Generated LoRA Weights



Analysis of Generated LoRA Weights

- Weights belong to well-structured space and can be semantically clustered
- Larger models yield far cleaner clusters. Cluster groups include Math (GSM8K, SVAMP), Multiple Choice Words (SNLI, BoolQ) and Multiple Choice Indices (rest).
- While multiple choice, Winogrande and HELLASWAG are closer to the Math cluster. Turns out, their multiple choice indices are numbers.

Instruction: Choose the most reasonable answer for the question from the given options. Respond only with A, B, C or D

Question: Planets outside of our solar system have been detected. What suggested the presence of a planet outside of our solar system? **A.** radio wave emissions **B.** a wobble in the rotation of the star **C.** regular occurring eclipses of its moons **D.** the discovery of a star as large as our own. **Answer:**

Pythia: The sun is the surface of the moon E.the sun is the sun and the stars are the sun. A.The sun is the sun, the sun is the sun. The sun B. a wobble in the rotation of the star C. regular occurring eclipses of its moons D. the discovery of a star as large as our own ✖

TaskWeaver: B ⚠ (incorrect but follows instruction)

Question: Johnny is picking up the toys on the floor of his room. He'd dumped a lego boxed set with 500 pieces on the floor, and another one that had 3 times more pieces than the 500 piece one, and another one that had $\frac{1}{4}$ the number of pieces. How many blocks does Johnny pick up if he picks up all the legos?

Qwen: Johnny has three toy boxes:\n- First box: 500 pieces\n- Second box: 3 times more than the first, so $3 \times 500 = 1500$ \$ pieces\n- Third box: $\frac{1}{4}$ \$ of the second, so $\frac{1}{4} \times 1500 = 375$ \$ pieces\n\nAdding them all up: $500 + 1500 + 375 = 2375$ ✖

TaskWeaver: Johnny has three boxes of toys:\n1. A box with 500 pieces.\n2. A box with 3 times more pieces than the first box: $3 \times 500 = 1500$ \$ pieces.\n3. A box with $\frac{1}{4}$ \$ the number of pieces than the first box: $\frac{1}{4} \times 500 = 125$ \$ pieces.\n\nAdding them up: $500 + 1500 + 125 = 2125$ ✔

Limitations

- Datasets heavily biased to language/logic over Math since math requires more compute for thinking through steps
- Very basic hypernetwork backbone
- Rigid evaluation. Current evaluation for open-ended generation is exact-match/rule-based, leading to potential false positives/negatives

Going forward...

- More experiments with more datasets, bigger models, more modules
- More sophisticated hypernetwork backbone (transformer based?)
- Explore context compilation in LoRA weights for context intensive task such as RAG or few-shot prompting.