Metalic

Meta-Learning In-Context with Protein Language Models

Jacob Beck, Shikha Surana, Manus McAuliffe, Oliver Bent, Thomas D. Barrett,
Juan Jose Garau Luis, Paul Duckworth

InstaDeep™

ICLR

Summary

We introduce **Metalic** – leveraging protein foundation models,
in-context learning (ICL) over related tasks, and fine-tuning, for protein
fitness prediction. Metalic achieves **state-of-the-art** results on
ProteinGym benchmarks with less computation and fewer parameters.

Background

Goal: We have a protein; we want to optimize a property, called *fitness*
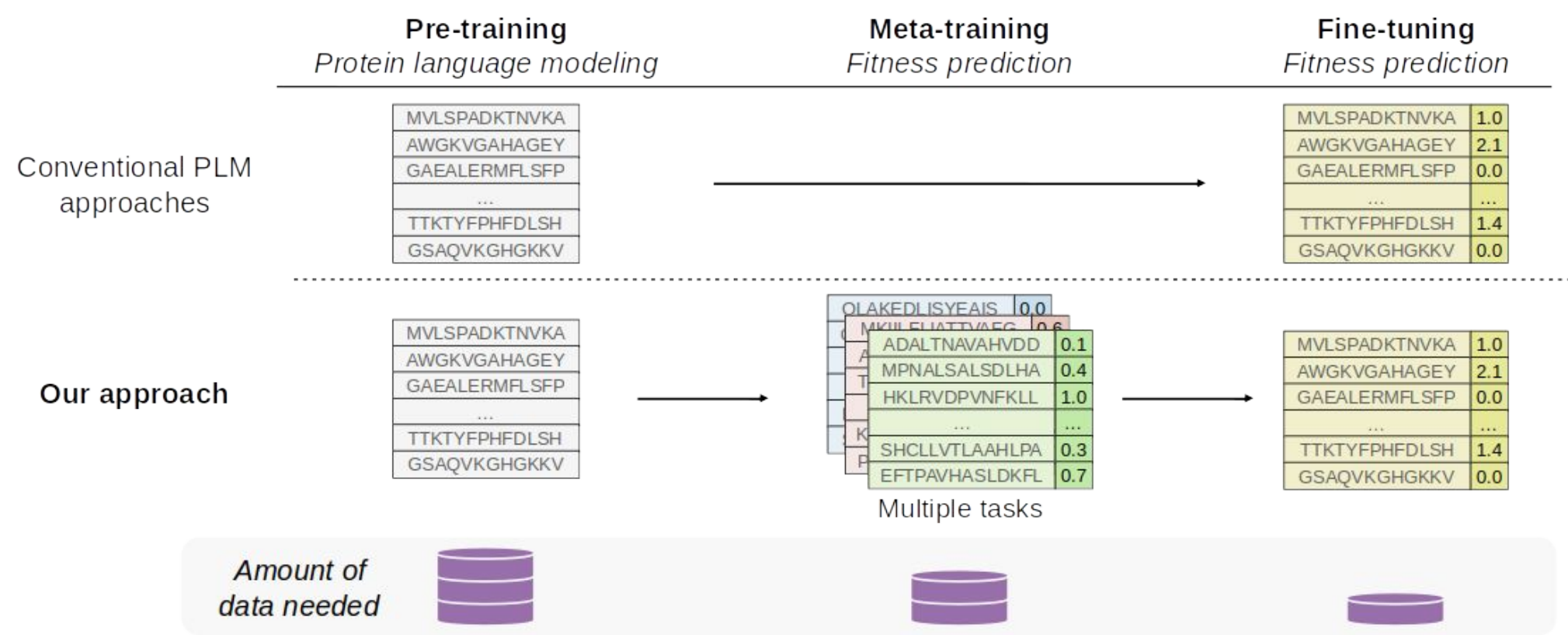**Problem:** We have limited data
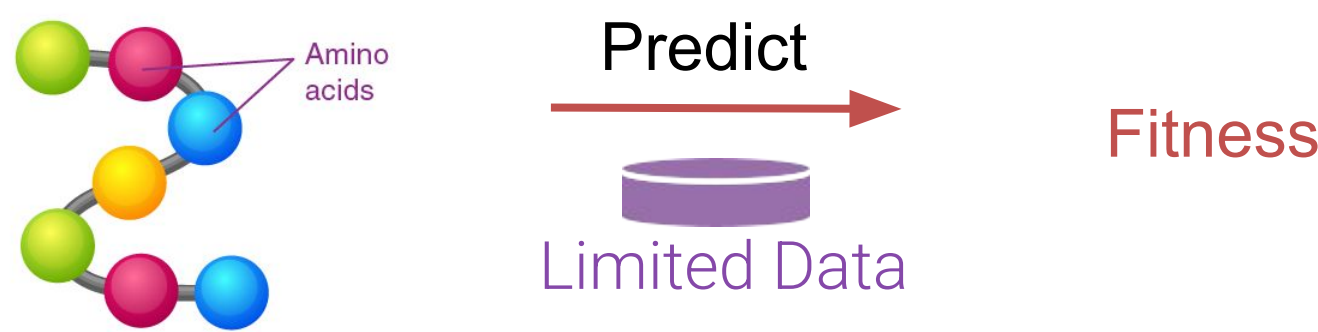
Existing Solution: Fine-tune a Protein Language Model (PLM)
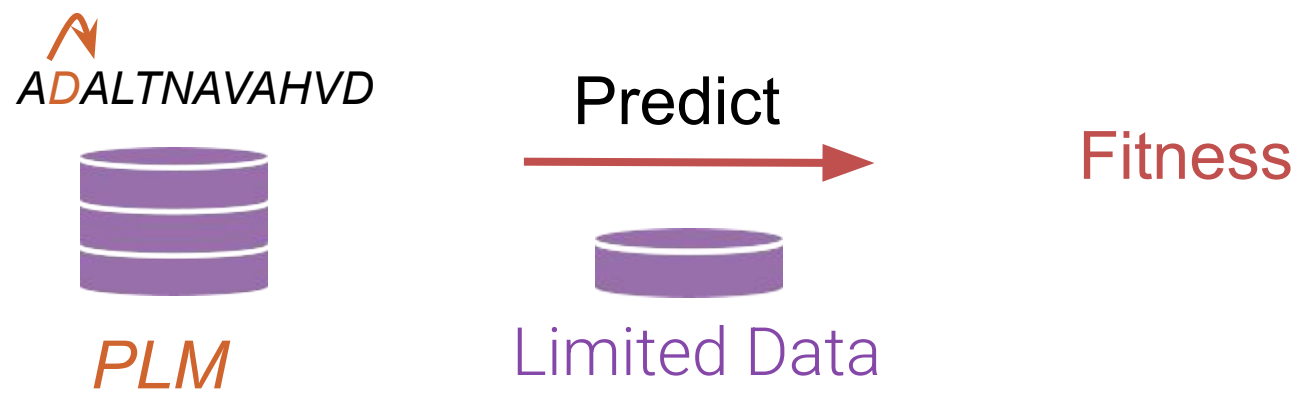
Each individual task has limited data, however, there are many related
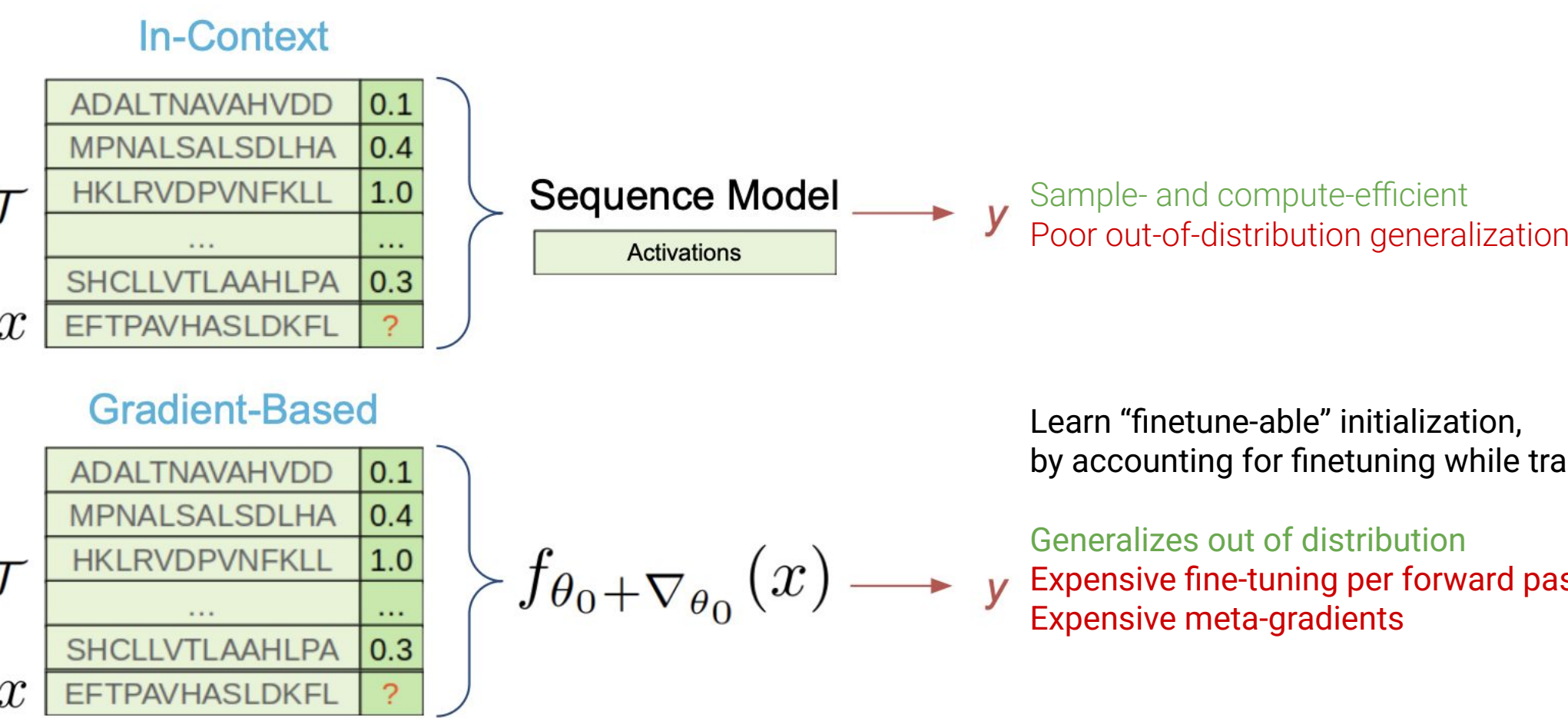proteins and properties

Solution: Learn how to learn from limited data – i.e. *Meta-learning*

SL $\quad \mathbb{E}_{x,y \in \mathcal{D}}[\mathcal{J}(f_\theta(x), y)]$

Meta-SL $\quad \mathbb{E}_{\mathcal{D}_\mathcal{T} \in \mathcal{D}}[\mathbb{E}_{x,y \in \mathcal{D}_\mathcal{T}}[\mathcal{J}(f_\theta(x, \mathcal{D}_\mathcal{T}), y)]]$

Task — Inner-Loop

Two Types of Inner-Loop
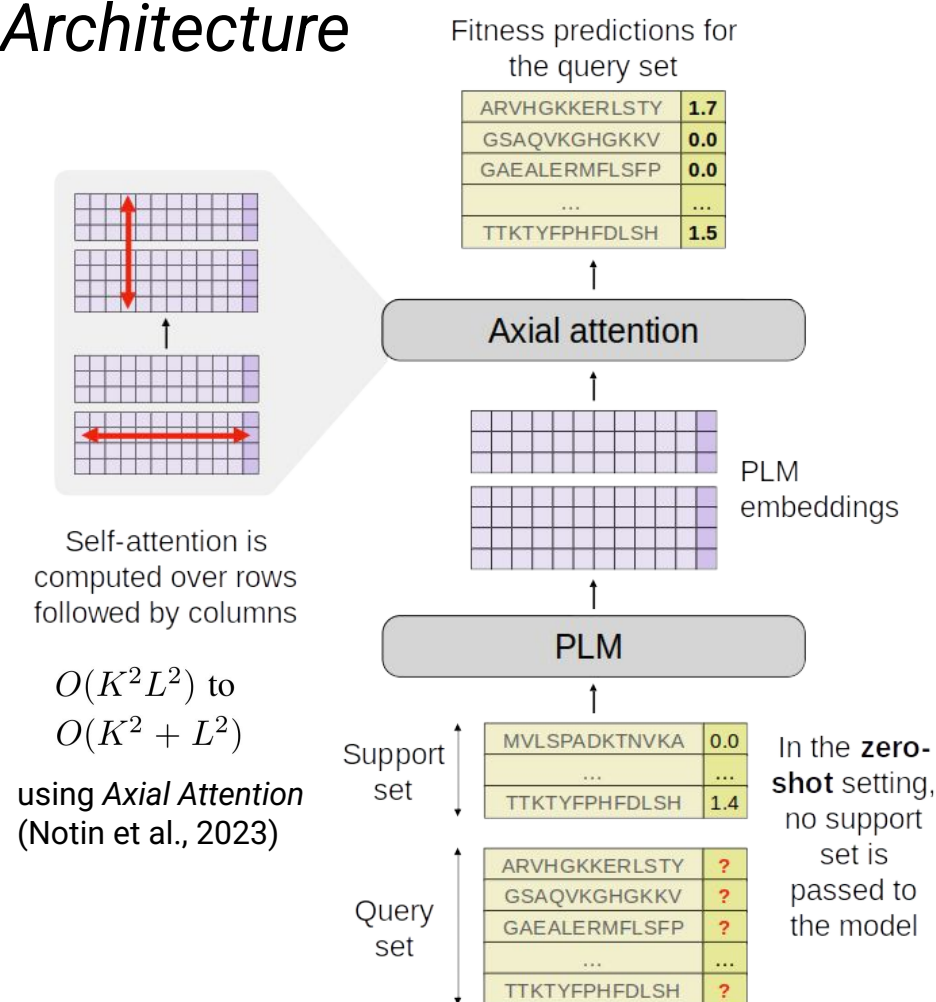
In-Context

Sample- and compute-efficient
Poor out-of-distribution generalization

Gradient-Based

Learn "finetune-able" initialization,
by accounting for finetuning while training

Generalizes out of distribution
Expensive fine-tuning per forward pass
Expensive meta-gradients

Metalic performs *in-context* meta-learning, followed by fine-tuning – the best of both worlds

Method: *Metalic*

1. Architecture

Fitness predictions for the query set

Self-attention is computed over rows followed by columns

$O(K^2L^2)$ to
$O(K^2 + L^2)$
using *Axial Attention*
(Notin et al., 2023)

In the **zero-shot** setting, no support set is passed to the model

2. Meta-Training Phase
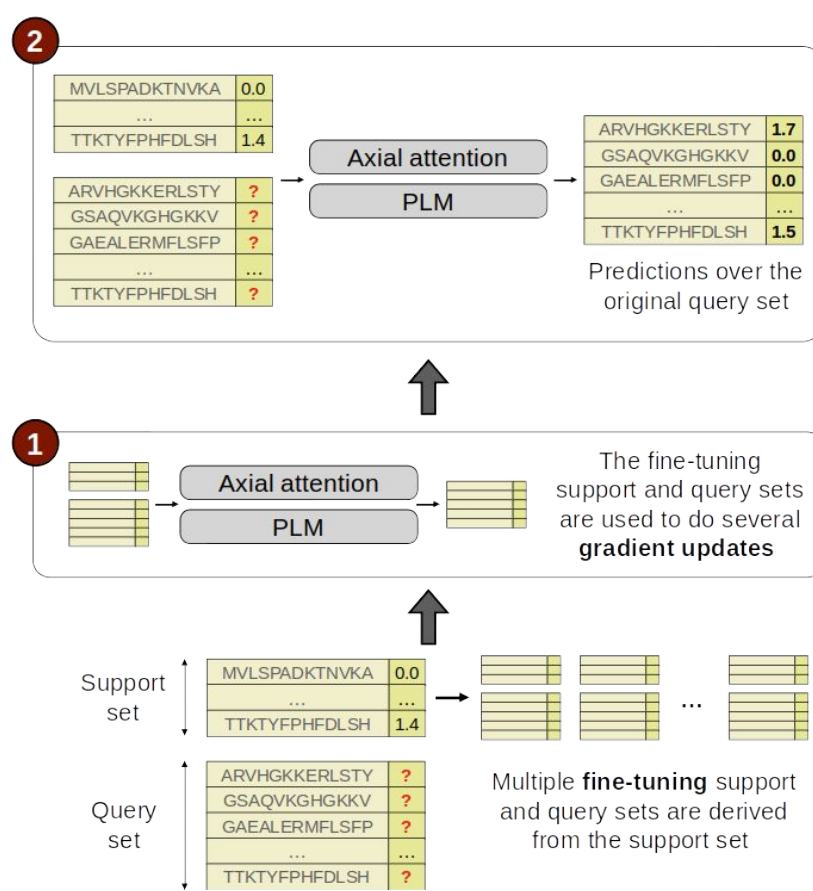
Preference Loss[1]
$$p\left(y_i^{(Q)} > y_j^{(Q)}\right) = \sigma\left(v_i^{(Q)} - v_j^{(Q)}\right)$$

Binary Cross Entropy over Query Set
$$\mathcal{L}(\theta, \mathcal{D}_\mathcal{T}^{(Q)}, \mathcal{D}_\mathcal{T}^{(S)}) = -\sum_{i=1}^{N^{(Q)}} \sum_{\substack{j=1 \\ j \neq i}}^{N^{(Q)}} \mathbb{I}\left(y_i^{(Q)} > y_j^{(Q)}\right) \log \sigma\left(v_i^{(Q)} - v_j^{(Q)}\right)$$

Evaluate Across Tasks
$$\mathcal{J}(\theta, \mathcal{D}) = -\mathbb{E}_{\mathcal{D}_\mathcal{T} \in \mathcal{D}} \mathbb{E}_{(\mathcal{D}_\mathcal{T}^{(S)}, \mathcal{D}_\mathcal{T}^{(Q)}) \in \mathcal{D}_\mathcal{T}} \mathcal{L}(\theta, \mathcal{D}_\mathcal{T}^{(Q)}, \mathcal{D}_\mathcal{T}^{(S)})$$

3. Fine-Tuning Phase

Problem

Combining in-context learning & fine-tuning (FT) creates a unique problem
- Query set labels are not available for FT
- If we update on support set, we could memorize

Solution

1) Divide support set into small support & query sets (ignoring original query set)

2) Then, recombine entire support and query set for in-context predictions

The fine-tuning support and query sets are used to do several **gradient updates**

Multiple **fine-tuning** support and query sets are derived from the support set

Results

ProteinGym (Notin et al., 2024)

We consider protein fitness tasks of length < 750 (to fit a single A100-80Gb)
- 121 single mutation DMS
- 68 multiple mutation DMS

Hold out 8 single and 5 multi-mutant DMS for evaluation.

1. Zero-Shot (n=0)

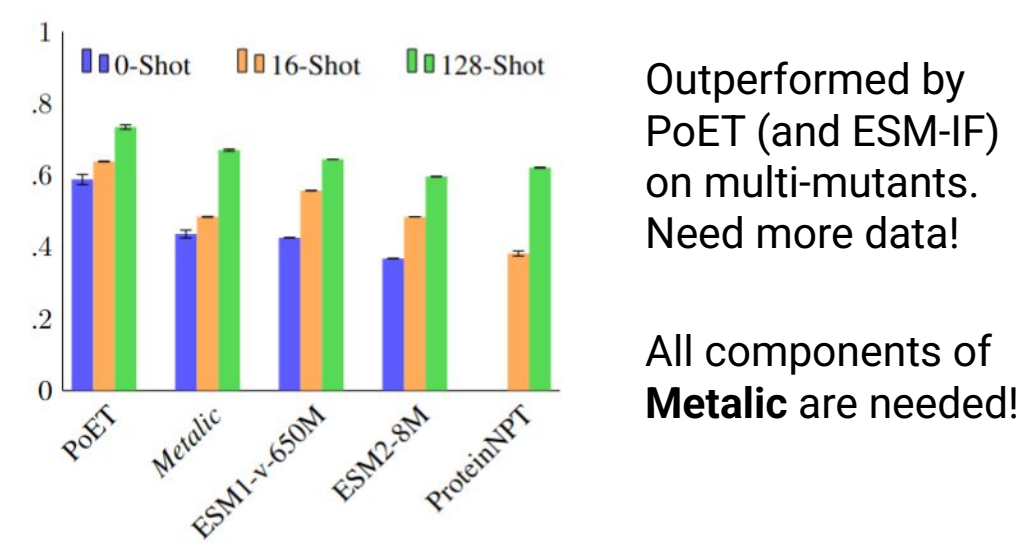| Model Name	Spearman Correlation
Metalic (max)	**.484**
Metalic (mean)	**.482** ± .002
VESPA	.464
TranceptEVE-Medium	.457
ESM1-v-650M	.437
Tranception-Medium	.427
Progen2-Medium	.419
ESM2-650M	.399
MSA Transformer	.398
ESM-IF1	.365
ESM2-8M	.121

SOTA, despite using ESM2-8M as PLM.
Baseline predictions from (Notin 2024). Baselines assume
PLM likelihood correlate with fitness whereas
Metalic meta-learns to use in-context information

2. Few-Shot (n=16, 128)

Model Name	$n = 16$	$n = 128$
Metalic	**.484** ± .001	.552 ± .009
Metalic-AuxIF	**.500** ± .002	.556 ± .005
ESM1-v-650M	.452 ± .000	.553 ± .000
ESM2-8M	.226 ± .000	.406 ± .000
PoET	.475 ± .026	**.588** ± .006
ProteinNPT	.192 ± .003	.443 ± .003

Strong, despite ESM2-8M as PLM and NPT architecture
ESF-IF1 as input (n=0) improves performance
Baselines are re-trained. Code from (Hawking-Hooker 2024).
**Metalic** only outperformed in 128 by PoET.

3. Multi-Mutant

Outperformed by
PoET (and ESM-IF)
on multi-mutants.
Need more data!

All components of
Metalic are needed!

4. Ablations

Model Name	$n = 0$	$n = 128$
Metalic	**.482** ± .002	**.552** ± .009
Metalic-NoFT	**.482** ± .002	.488 ± .012
Metalic-NoPref	.465 ± .011	.520 ± .006
Metalic-NoAug	.464 ± .011	**.558** ± .009
Metalic-NoICL	.441 ± .002	.529 ± .002
Metalic-NoMetaTrain	-.046 ± .018	.346 ± .003

5. Gradient-based Comparison

Model Name	Total Gradient Computations	$n = 128$
Metalic (150k)	150,000	**.562** ± .004
Metalic-Reptile	150,000	**.562** ± .004
Metalic (50k)	50,000	.552 ± .009
Reptile-3-100	150,000	.539 ± .001
Reptile-3-3	150,000	.499 ± .008

Reptile is a gradient-based method

Metalic (50k) outperforms Reptile

Metalic with Reptile (**Metalic**-Reptile) is marginally
better, but not when controlling for compute (150k)

Conclusion: Accounting for fine-tuning during meta-training is needlessly complicated

Analysis

(a) Untrained Attention   **(b)** Half-Trained Attention   **(c)** Trained Attention
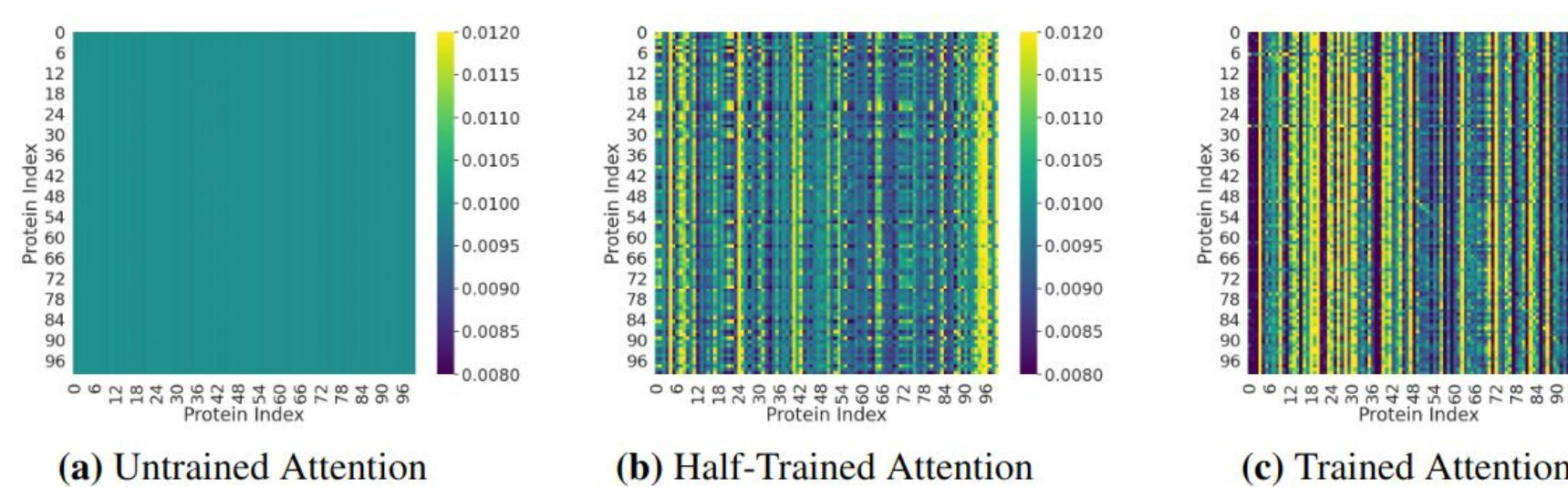
Query set **attention maps** in zero-shot setting

All rows show attention to other proteins; implying the need for **ICL**

Demonstrates the emergence of **unsupervised ICL**