Cramming Protein Language Model Training in 24 GPU Hours

Nathan C. Frey, Taylor Joren, Aya Abdelsalam Ismail, Allen Goodman,
Stephen Ra, Richard Bonneau, Kyunghyun Cho, Vladimir Gligorijević (*Prescient Design*)

Prescient Design
A Genentech Accelerator

Motivation

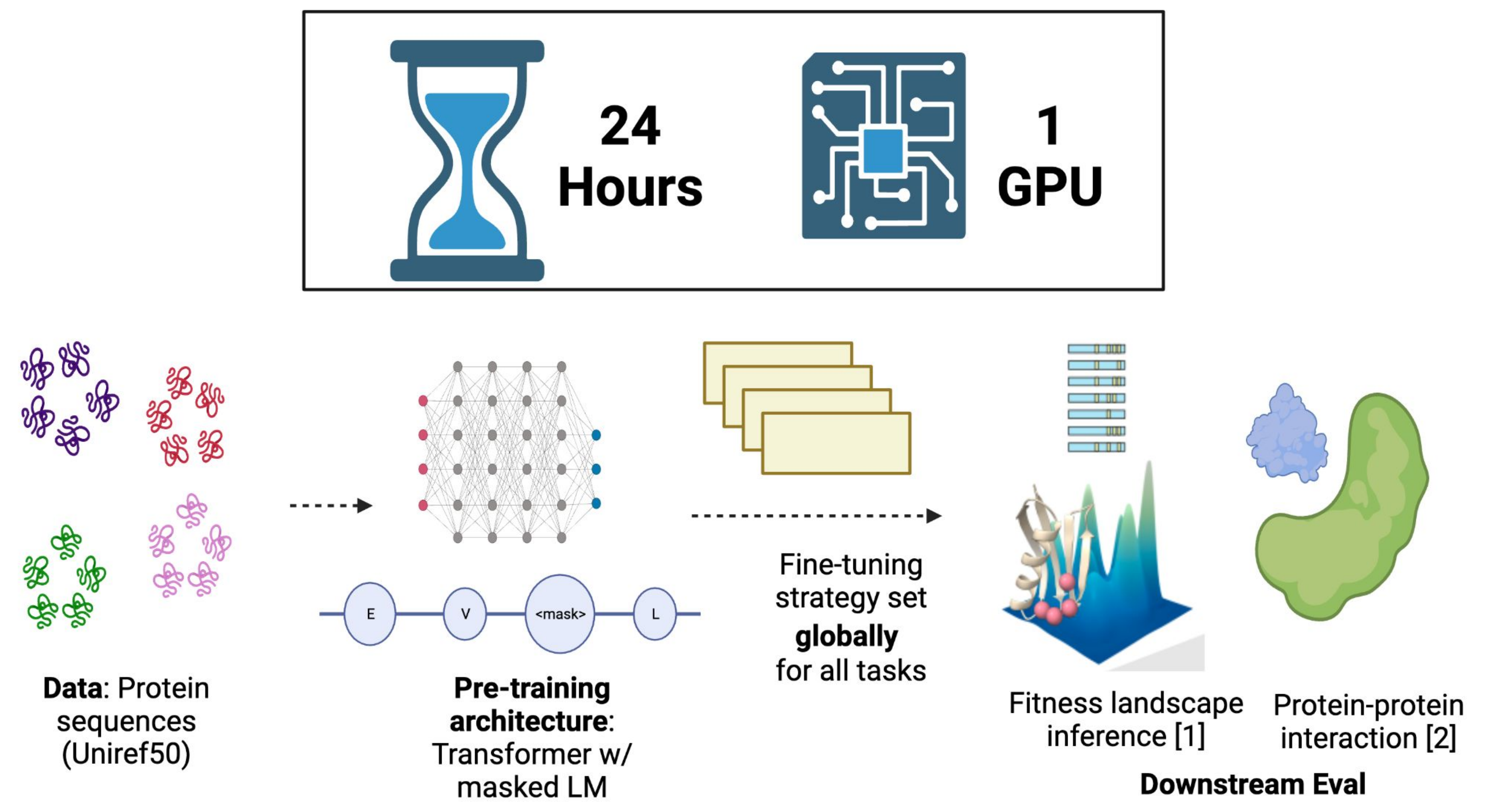
- Protein Language Models (pLMs) are traditionally trained following recipes from natural language processing for **hundreds of thousands of GPU hours**, making scientific investigations of pre-training and fine-tuning impractical for most BioML practitioners.
- **Rapid pre-training and fine-tuning** of pLMs is needed to enable fundamental progress in language modeling for proteins.

Summary

- We define a **"cramming" challenge** for Protein Language Models (pLMs): to train competitive pLMs in **24 hours on a single GPU**.
- We re-examine many aspects of pLM training and achieve a **15,000x speedup** in pre-training a pLM that is competitive with ESM2 on downstream protein landscape inference tasks.

"Cramming" challenge for Protein Language Models

1. **Transformer-based pLM** is trained from scratch with a masked LM objective.
2. Training may **not exceed 24 hours on a single GPU**.
3. **No existing pre-trained models** are used at any point.
4. **Train/val/test splits are pre-specified from UniRef50**. The training data can be sampled in any way that does not involve a pre-trained model, hence speedups may be achieved by careful choices of how and when to sample training data.
5. **All preparation of raw FASTA inputs for training is included in the training budget**. (e.g., tokenization, filtering, sorting, etc.) The downloading of raw data in FASTA format is *exempt* from the overall compute budget.
6. **Downstream performance is evaluated on tasks from the FLIP [1] and PPI [2] benchmarks**. Fine-tuning strategy is flexible but must be set globally for all downstream tasks:
   - 🌎 Set globally: Prediction head architecture, hyperparameters, aggregation to pool token embeddings.
7. **Downstream fine tuning is not included** in the 24 GPU hour budget.


24
Hours
1
GPU
E
V
<mask>
L
Data: Protein sequences (Uniref50)
Pre-training architecture: Transformer w/ masked LM
Fine-tuning strategy set **globally** for all tasks
Fitness landscape inference [1]
Protein-protein interaction [2]
Downstream Eval

Language model cramming setup adapted from [3].

Strategies for accelerating pLM pre-training

- The goal of all architectural and training interventions is to **maximize per-token training efficiency**.
- *Balance learning rate & learning rate scheduler*: Maximize learning rate without causing training instabilities. Stabilize w/ gradient clipping.
- *Remove bias terms:* Starting from ESM2 [4] model architecture, we remove all query, key, and value biases in all attention blocks and all bias terms in intermediate linear layers.
- We *increase the masking rate* to 25%.
- Apply *gradient accumulation* to achieve an effective batch size of ~1M tokens.

All pre-training code, pre-trained models, datasets and splits will be open sourced in a future, archival version of the publication.

Experiments

Learning dynamics dominate pLM pre-training perplexity.

Learning rate	Number of warmup steps	Validation perplexity ↓
0.001	**1000**	**13.72**
0.0004	1000	13.92
0.01	10000	13.96
0.001	100	14.10
0.001	10000	14.31
0.001	40000	14.88
0.004	1000	17.42
0.004	100	20.49

Crammed pLMs achieve comparable performance (Spearman correlation) to fully trained pLMs on downstream tasks.

10% time limit, IID split

Model	GB1	AAV	Meltome	PPI
Crammed pLM-67M (Ours)	0.53	0.76	0.34	0.78
ESM2-8M	0.59	0.81	0.42	0.86
ESM2-150M	0.55	0.78	0.29	0.88
ESM2-3B	0.40	0.62	0.20	0.85

No time limit, IID splits

Model	GB1	AAV	Meltome	PPI
Crammed pLM-67M (Ours)	0.63	0.79	0.51	0.78
ESM2-8M	0.59	0.83	0.59	0.86
ESM2-150M	0.58	0.82	0.63	0.88
ESM2-3B	0.66	0.81	0.54	0.88

- We impose a 10% cramming time limit (2.4 hours) for fine-tuning, which enforces that the computational cost of fine-tuning is negligible compared to the pre-training budget → penalizes larger models.
- Both with and without the time limit, crammed pLMs perform similarly or better than fully trained models on IID splits ("mixed" split for Meltome).

Performance gains explained by pre-training in most cases.

OOD splits, No time limit

	Pre-trained			Baseline (no pre-training)		
Model	**GB1**	**AAV**	**Meltome**	**GB1**	**AAV**	**Meltome**
Crammed pLM-67M (Ours)	0.42	0.12	0.41	0.33	-0.03	0.48
ESM2-8M	0.16	0.29	0.29	-0.02	-0.10	-0.19
ESM2-150M	0.16	0.38	0.44	0.17	-0.13	-0.21
ESM2-3B	0.19	0.20	0.36	0.30	-0.10	-0.23

- In some cases, fine-tuning drives performance more than pre-training in crammed & non-crammed models. This questions whether standard fine-tuning practices like global downstream pooling are suboptimal for OOD generalization.
- 24 hrs of pre-training produce representations that generalize to OOD data.

Discussion

- We introduce a "cramming" challenge for pLMs and encourage others to improve on the work, democratizing pre-training research for BioML.
- Using modified transformer-based architectures and masked LM training recipes, we trained performant protein language models (pLMs) in 24 hours on a single GPU. This allows us to rapidly test novel pre-training and fine-tuning ideas and question fundamental assumptions related to treating biological sequence data in the same way as natural language.
- Our results indicate that pre-trained pLMs have advantages for structural (e.g., protein-protein interface prediction) tasks and there is great room for improvement on global downstream pooling and fine-tuning strategies.

Genentech
A Member of the Roche Group

References

[1] Dallago, et al. Flip: Benchmark tasks in fitness landscape inference for proteins. bioRxiv, pp. 2021–11, 2021.

[2] Suyu Mei and Kun Zhang. International Journal of Molecular Sciences, 20, October 2019. ISSN 1422-0067. doi: 10.3390/ijms20205075.

[3] Jonas Geiping and Tom Goldstein. Cramming: Training a language model on a single gpu in one day. In International Conference on Machine Learning, pp. 11117–11143. PMLR, 2023.

[4] Lin, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. Science, 379(6637):1123–1130, 2023.

Figures created with BioRender.com

Code, models, datasets and splits will be made available:

https://github.com/prescient-design

https://github.com/Genentech