Scalable Data Ablation Approximations for Language Models through Modular Training and Merging

Clara Na[1,2], Ian Magnusson[1,3], Ananya Harsh Jha[1,3], Tom Sherborne[4],
Emma Strubell[1,2], Jesse Dodge[1], Pradeep Dasigi[1]

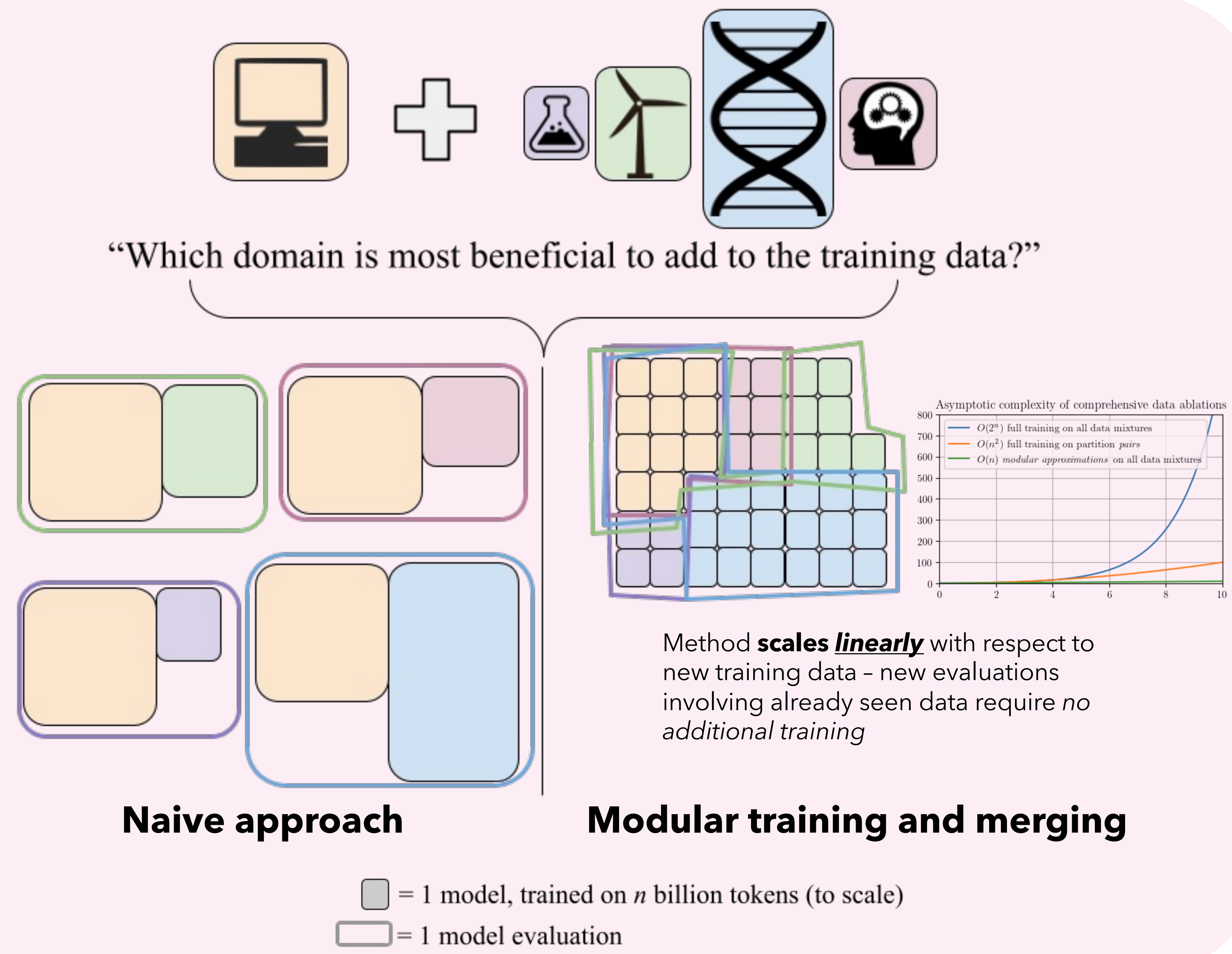
[1]Allen Institute for AI, [2]Carnegie Mellon University, [3]University of Washington, [4]Cohere

How can we understand and improve our language models' data mixtures?

Naive, infeasible: train and evaluate models on every possible data recipe

Efficient, scalable: Data partitioning, Modular training, Evaluating merged models

Key finding: Evaluations of merged models correlate strongly with evaluations of models trained on combined datasets
→ we can *reuse* training computation across evaluations
→ we can simulate *comprehensive* and *fine-grained* data ablations



Hypothesis:

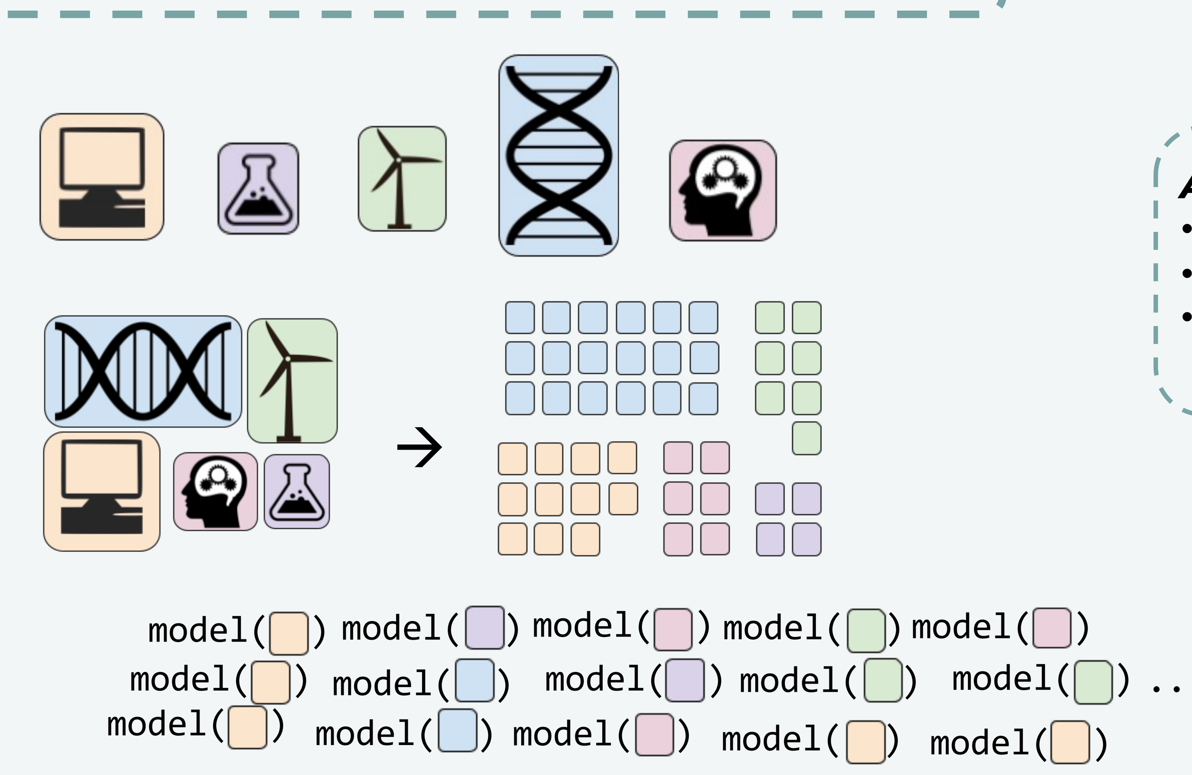
For training data `A, B, C,` eval data `x`:

```
eval(merge{model(A), model(B), model(C)} | x ) ∝ eval(model{A + B + C} | x )
```

Proposed Method

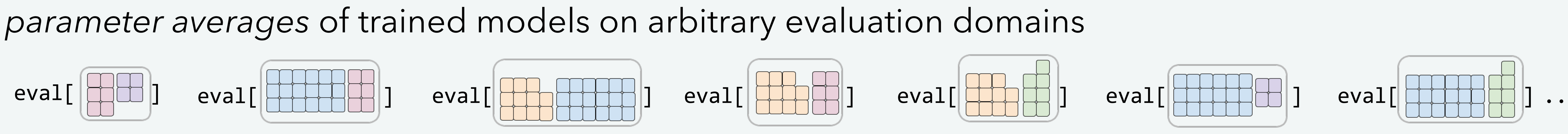
Start with a training corpus with data domains of interest.

1. Further partition / recombine into similarly sized "base units"
2. Train one model on each base unit of data
   - Use same seed model initialization and amount of training for each model!

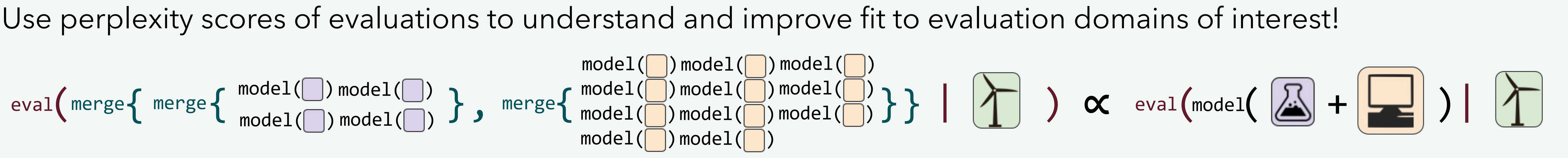


Ask me about:
- How much training?
- "Merge" how?
- What's the catch?

3. Evaluate *parameter averages* of trained models on arbitrary evaluation domains



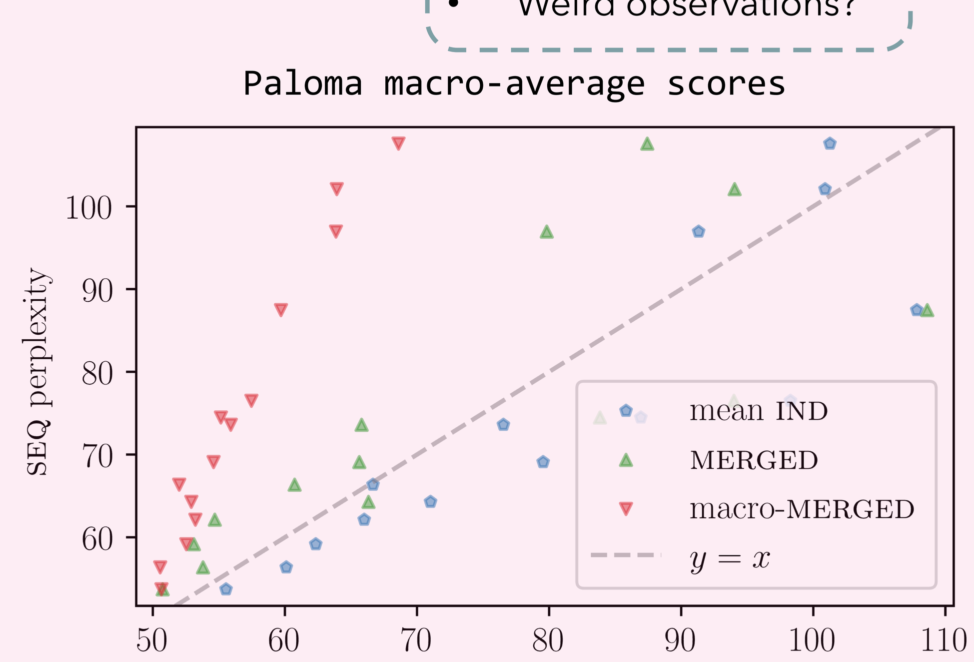
4. Use perplexity scores of evaluations to understand and improve fit to evaluation domains of interest!



Example : Two S2ORC fields of study

Ask me about:
- Other experiments?
- Weird observations?

	Average IND scores		MERGED scores		
	MERGED	SEQ	SEQ	macro-	micro-
$\mathcal{P}_1$	0.844	0.826	0.763	0.929	0.937
$\mathcal{P}_2$	0.909	0.930	0.914	0.959	0.946
$\mathcal{P}_1 + \mathcal{P}_2$	0.856	0.844	0.755	0.944	0.938
M2D2 S2	0.869	0.908	0.608	0.894	**0.918**
M2D2 Wiki	0.905	0.886	0.822	**0.966**	0.858
Wiki-103	0.927	0.885	0.783	**0.983**	0.867
PTB	0.880	0.835	0.694	**0.929**	0.773
4chan	0.874	0.883	0.866	**0.905**	0.785
c4-en	0.905	0.863	0.798	**0.985**	0.849
mc4-en	0.844	0.836	0.770	**0.969**	0.829
RedPajama	0.790	0.902	0.838	**0.930**	0.852
Manosphere	0.917	0.919	0.895	**0.976**	0.867
Avg (macro)	0.910	0.882	0.790	**0.984**	0.848



Experimental setting

- 130m and 1.1b decoder-only models
- Continued pre-training on S2ORC, M2D2 Wiki
- Perplexity evaluation on held-out sets + OOD sets from Paloma
- In one data ablation study, fix number of partitions

Future work

- Fine-grained downstream task adaptation?
- 7b models?
- Mergeability when training data is more diverse – multi-lingual? + code?