Model Ratatouille:

Recycling Diverse Models for Out-of-Distribution Generalization

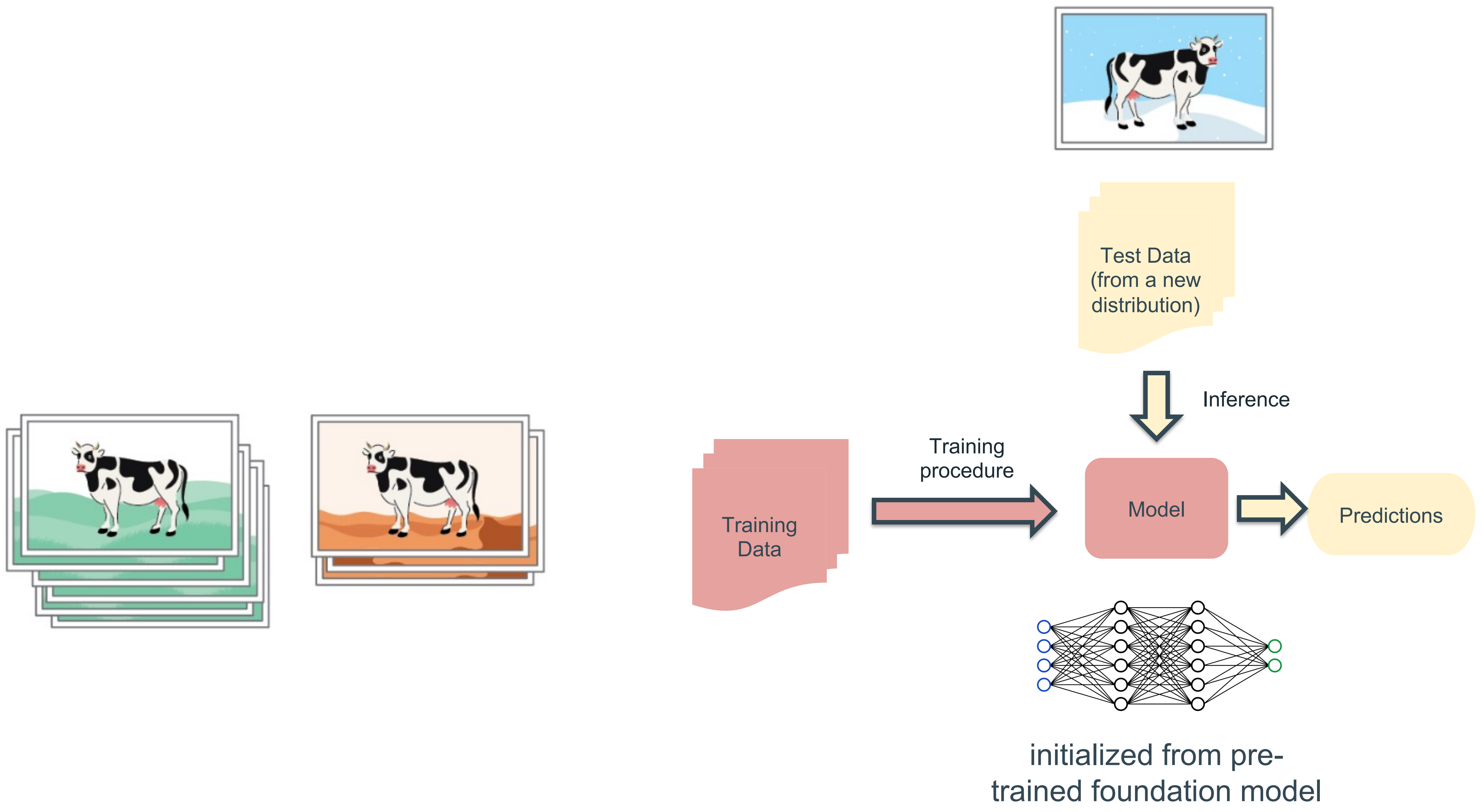
Alexandre Ramé, Kartik Ahuja, Jianyu Zhang, Matthieu Cord, Léon Bottou, David Lopez-Paz

ICML 2023

SCIENCES SORBONNE UNIVERSITÉ

Meta

Goal: generalization under distribution shift

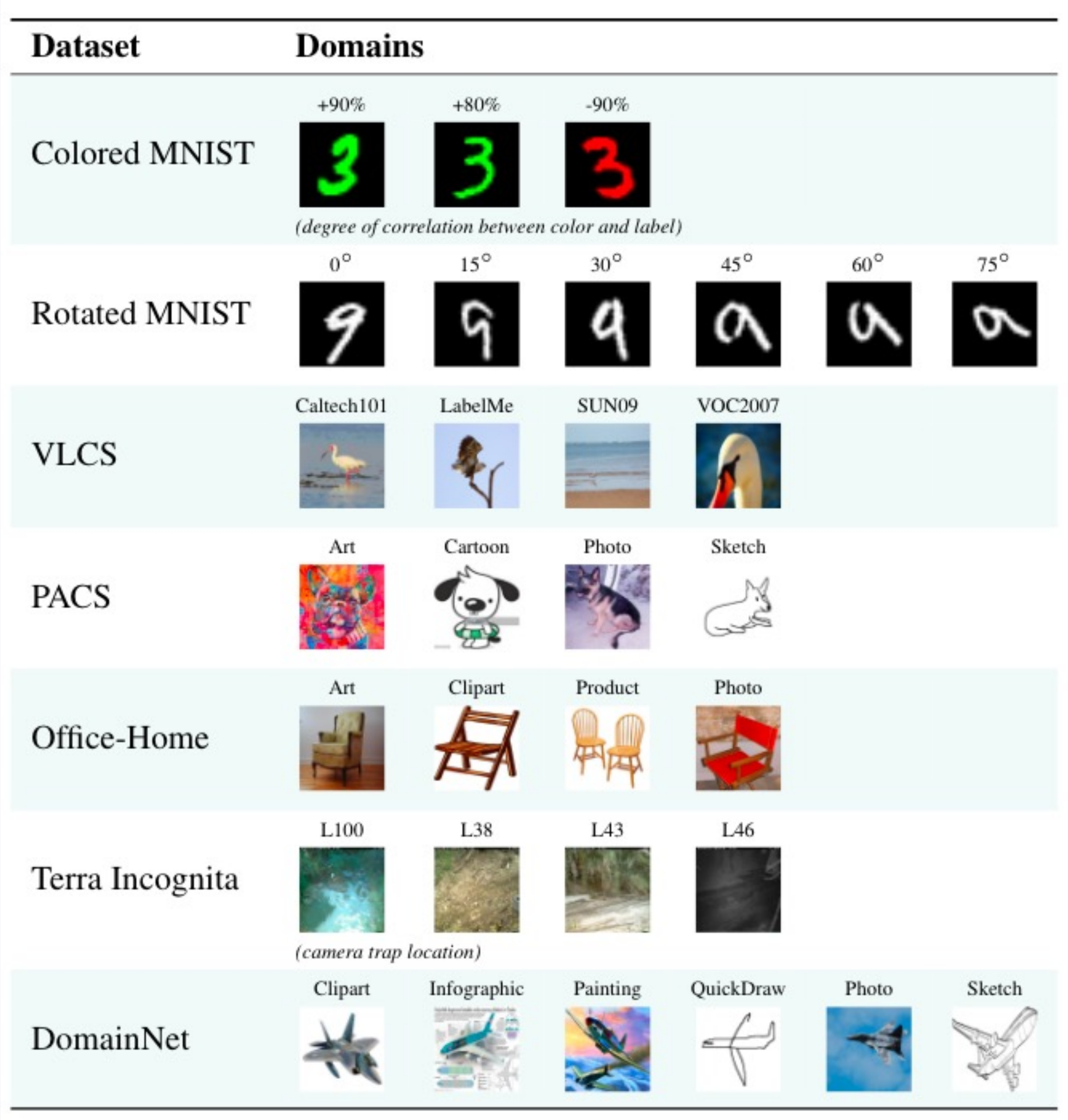


Class	Probability
Cow	0.8
Camel	0.19
Bear	0.01

How to best fine-tune foundation models ?

The github.com/facebookresearch/domainbed benchmark compares the different OOD approaches.

Key insight: ERM remained the best approach until …



Available algorithms

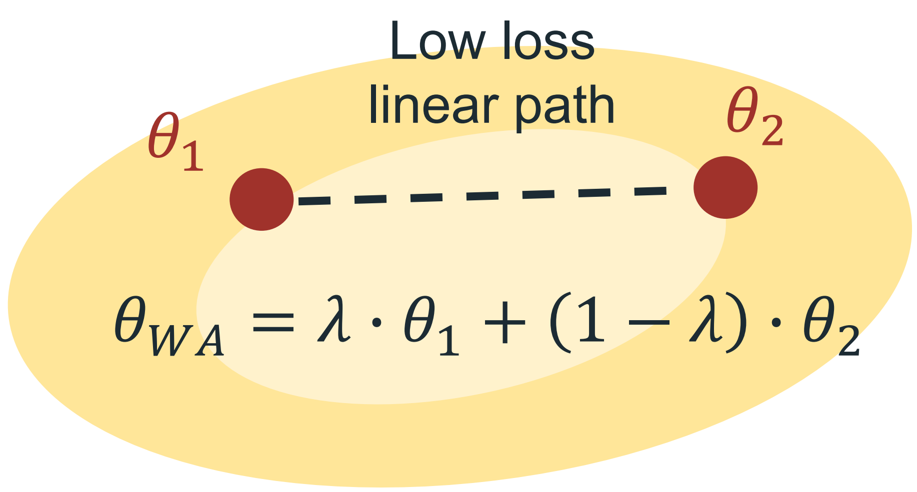
The currently available algorithms are:

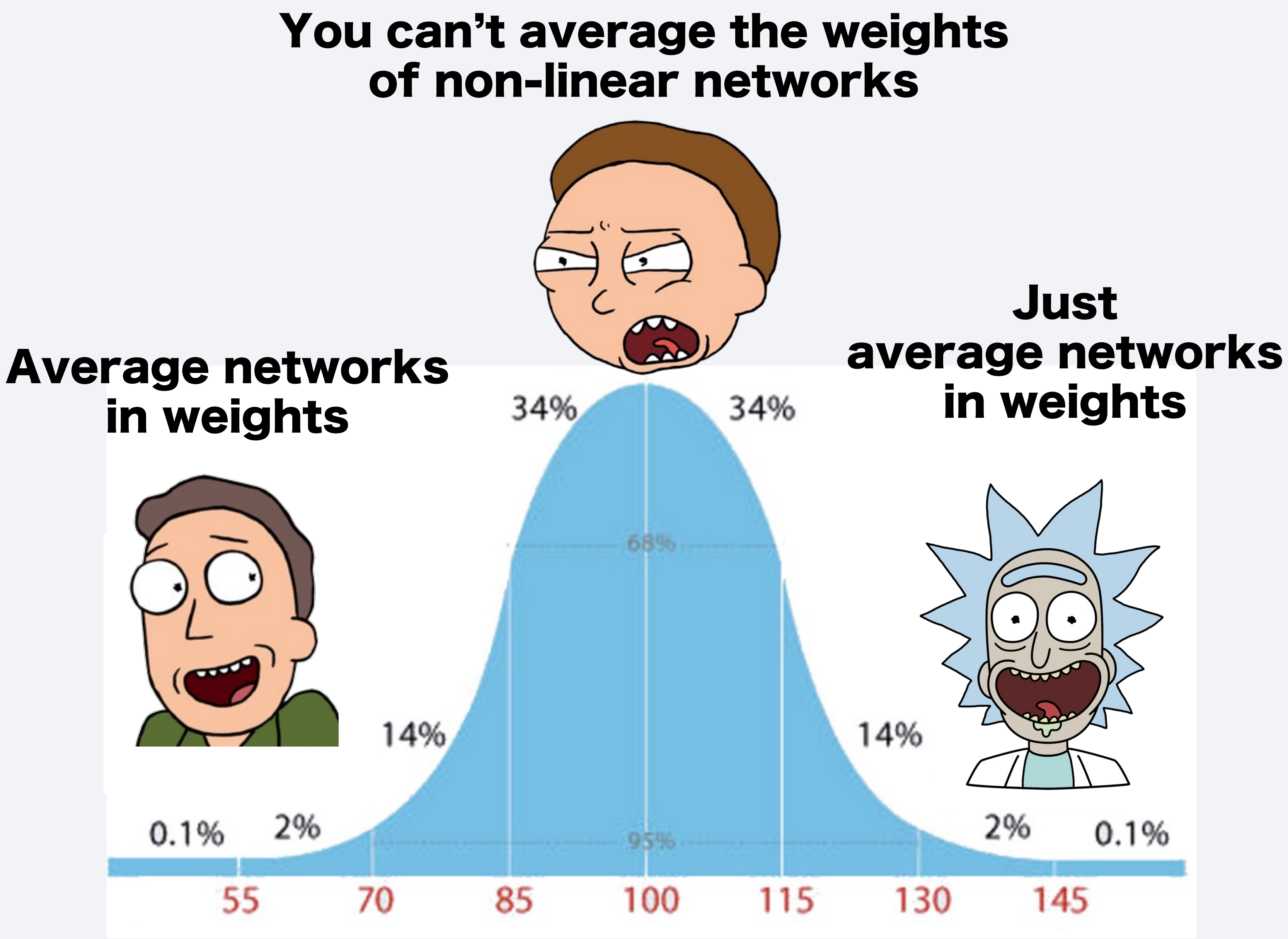
- Empirical Risk Minimization (ERM, Vapnik, 1998)
- Invariant Risk Minimization (IRM, Arjovsky et al., 2019)
- Group Distributionally Robust Optimization (GroupDRO, Sagawa et al., 2020)
- Interdomain Mixup (Mixup, Yan et al., 2020)
- Marginal Transfer Learning (MTL, Blanchard et al., 2011-2020)
- Meta Learning Domain Generalization (MLDG, Li et al., 2017)
- Maximum Mean Discrepancy (MMD, Li et al., 2018)
- Deep CORAL (CORAL, Sun and Saenko, 2016)
- Domain Adversarial Neural Network (DANN, Ganin et al., 2015)
- Conditional Domain Adversarial Neural Network (CDANN, Li et al., 2018)
- Style Agnostic Networks (SagNet, Nam et al., 2020)
- Adaptive Risk Minimization (ARM, Zhang et al., 2020), contributed by @zhangmarvin
- Variance Risk Extrapolation (VREx, Krueger et al., 2020), contributed by @zdhNarsil
- Representation Self-Challenging (RSC, Huang et al., 2020), contributed by @SirRob1997
- Spectral Decoupling (SD, Pezeshki et al., 2020)
- Learning Explanations that are Hard to Vary (AND-Mask, Parascandolo et al., 2020)
- Out-of-Distribution Generalization with Maximal Invariant Predictor (IGA, Koyama et al., 2020)
- Gradient Matching for Domain Generalization (Fish, Shi et al., 2021)
- Self-supervised Contrastive Regularization (SelfReg, Kim et al., 2021)
- Smoothed-AND mask (SAND-mask, Shahtalebi et al., 2021)
- Invariant Gradient Variances for Out-of-distribution Generalization (Fishr, Rame et al., 2021)
- Learning Representations that Support Robust Transfer of Predictors (TRM, Xu et al., 2021)
- Invariance Principle Meets Information Bottleneck for Out-of-Distribution Generalization (IB-ERM , Ahuja et al., 2021)
- Invariance Principle Meets Information Bottleneck for Out-of-Distribution Generalization (IB-IRM, Ahuja et al., 2021)
- Optimal Representations for Covariate Shift (CAD & CondCAD, Ruan et al., 2022), contributed by @ryoungj
- Quantifying and Improving Transferability in Domain Generalization (Transfer, Zhang et al., 2021), contributed by @Gordon-Guojun-Zhang
- Invariant Causal Mechanisms through Distribution Matching (CausIRL with CORAL or MMD, Chevalley et al., 2022), contributed by @MathieuChevalley

[Gulrajani2021] In Search of Lost Domain Generalization. ICLR.

Weight averaging

Consider θ_1 and θ_2 two weights for a given architecture. If the **linearly mode connectivity holds** in the test-loss landscape, then you can average them.

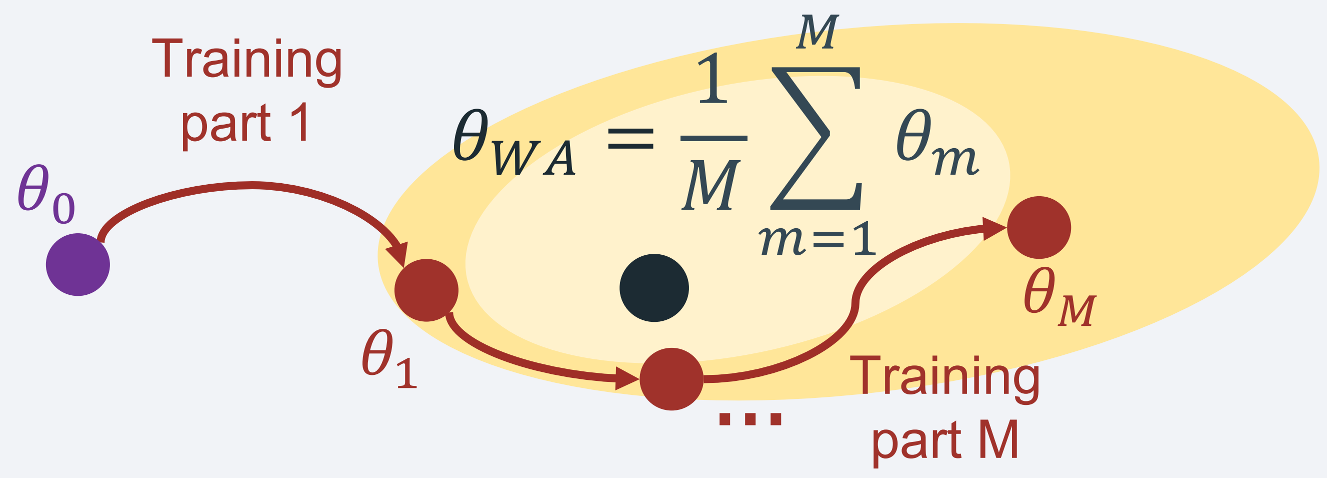




[Frankle2020] Linear mode connectivity and the lottery ticket hypothesis. ICML.

Weight averaging along a training trajectory

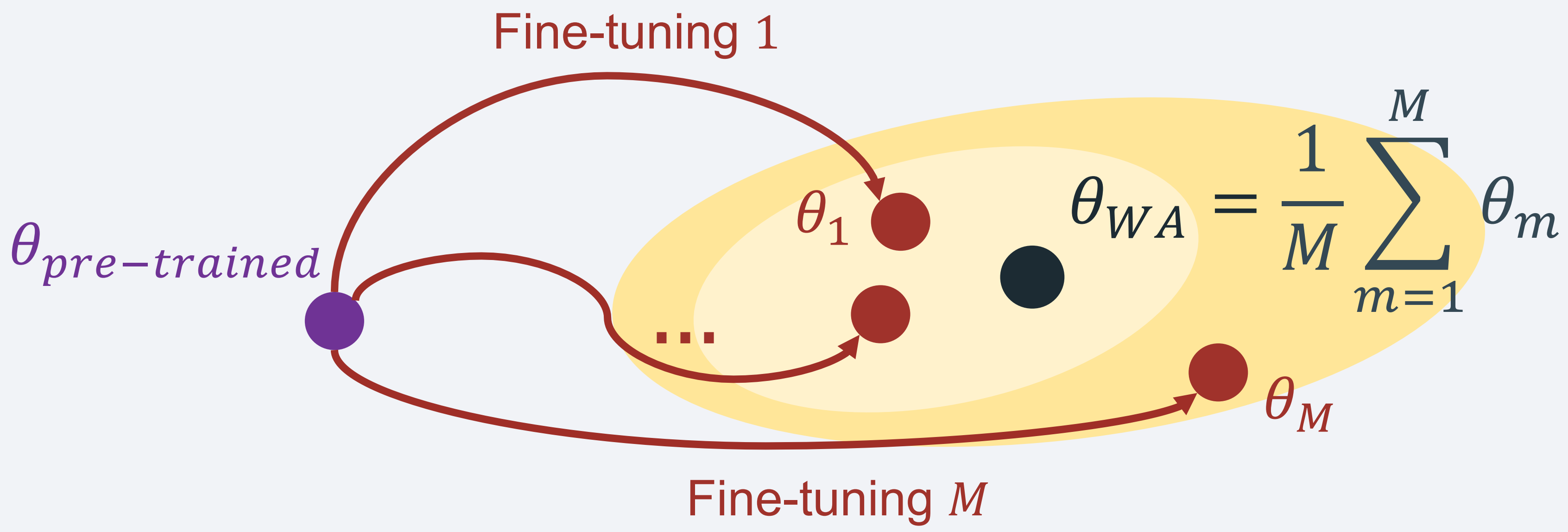
Moving average [Izmailov2018]: checkpoints collected along a training trajectory remain linearly connected.



[Izmailov2018] Averaging Weights Leads to Wider Optima and Better Generalization. UAI.
[Cha2021] SWAD: Domain Generalization by Seeking Flat Minima. NeurIPS.

Weight averaging from multiple trajectories

Model soups [Wortsman2022]: when fine-tuned from a shared pre-trained model with different hyperparams, weights remain linearly connected.



[Neyshabur2020] What is being transferred in transfer learning? NeurIPS.
[Wortsman2022] Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. ICML.
[Rame2022] DiWA: diverse weight averaging for out-of-distribution generalization. NeurIPS.

Weight averaging approximates ensembling.

Thus the 3 key criteria to trade-off:

1. Averageability

The weights should remain linearly connected.

2. Individual accuracies

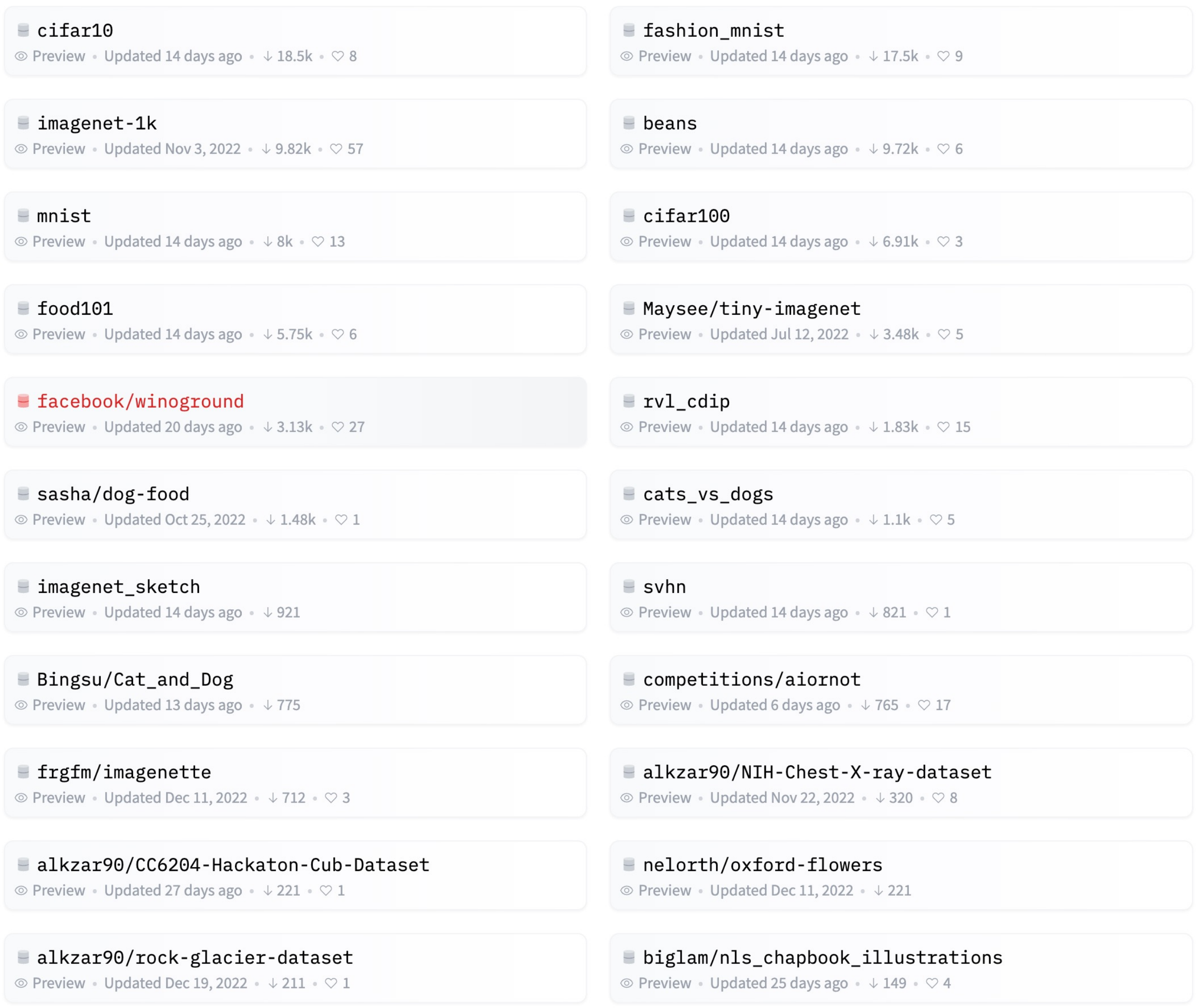
The weights should be individually accurate.

3. Diversity

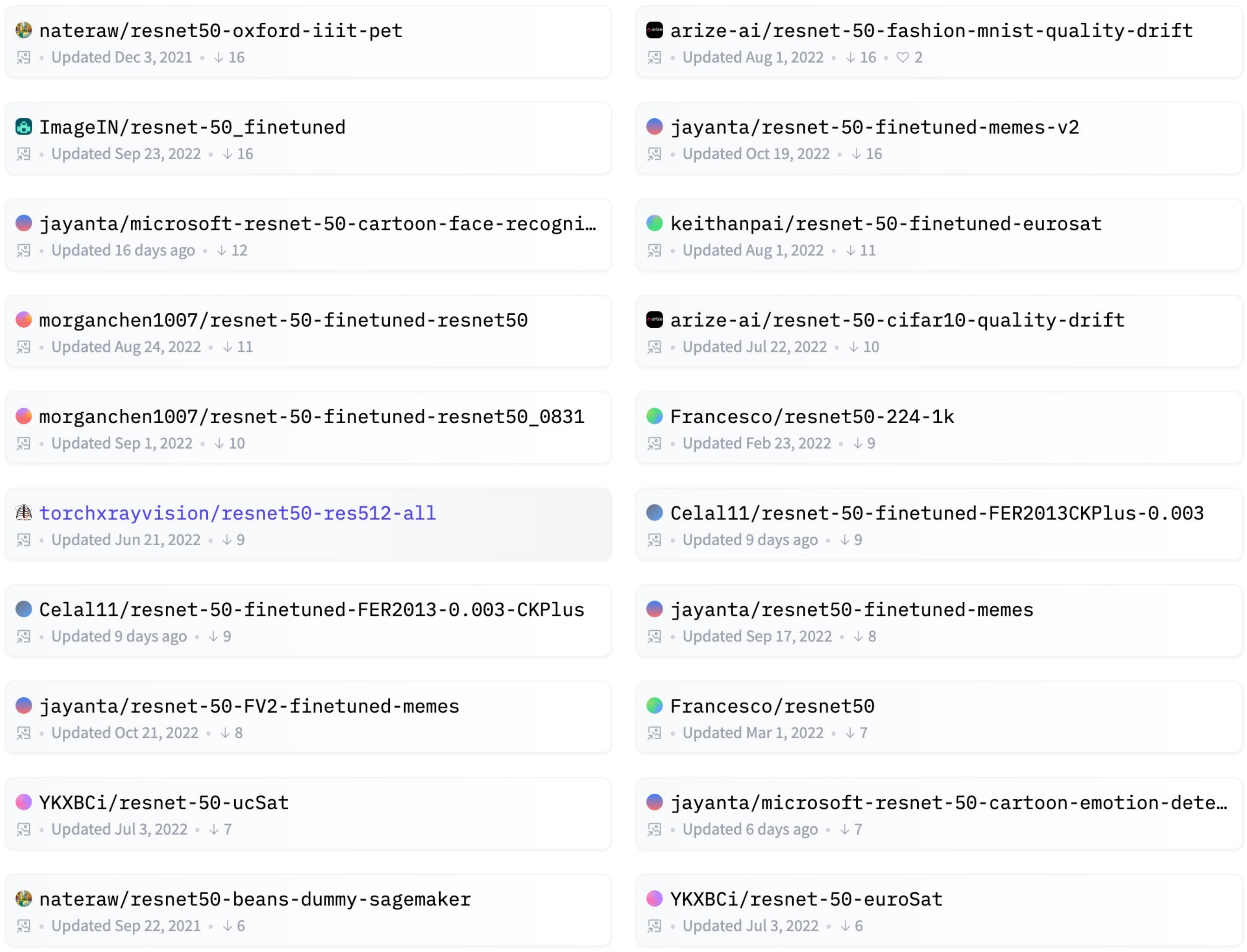
The weights should be sufficiently diverse to reduce variance.

Era of open-source datasets and weights

huggingface.co/datasets



huggingface.co/models/resnet-50

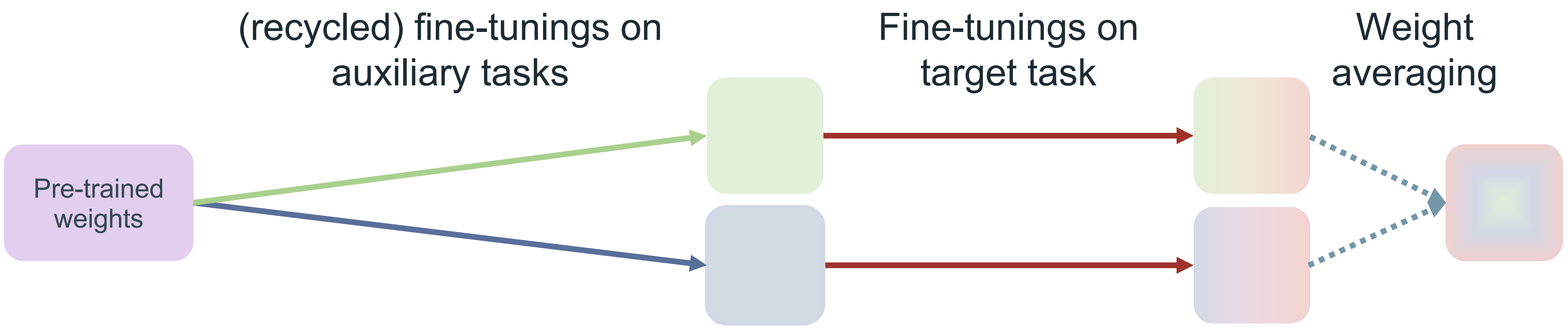


Key idea: recycle these weights as initializations for the target task.

Model Ratatouille: Recycling Diverse Models for Out-of-Distribution Generalization

A simple strategy:

1. From a shared pre-trained network.
2. Recycle multiple fine-tunings on auxiliary tasks.
3. From these weights, launch multiple fine-tunings on the target task.
4. Average all the fine-tuned weights.



Does Ratatouille meet the 3 key criteria for successful weight averaging ?

1. Averageability

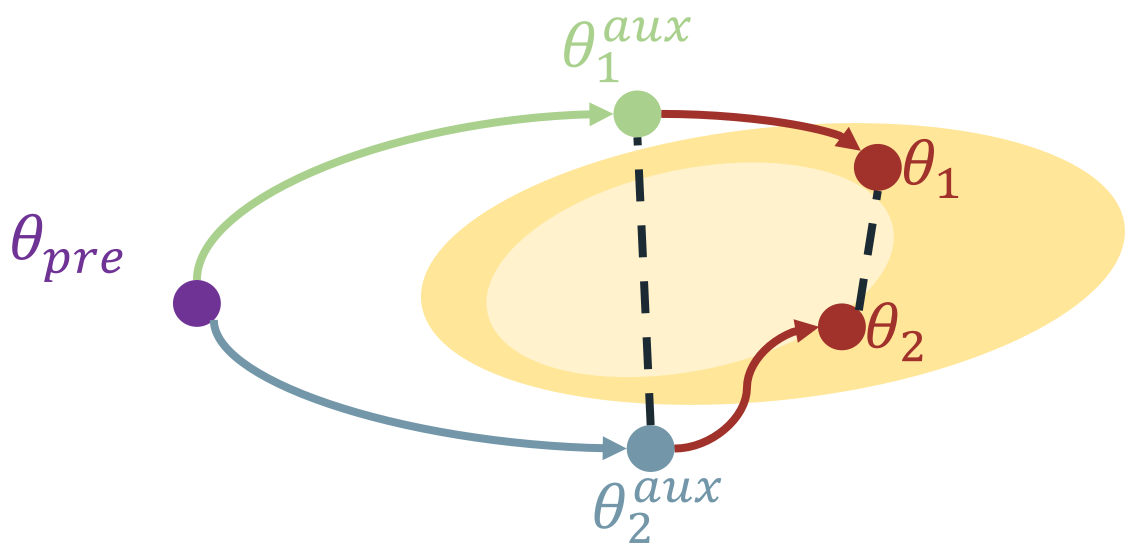
Yes, the weights remain linearly connected when auxiliary tasks are sufficiently similar.

2. Individual accuracies

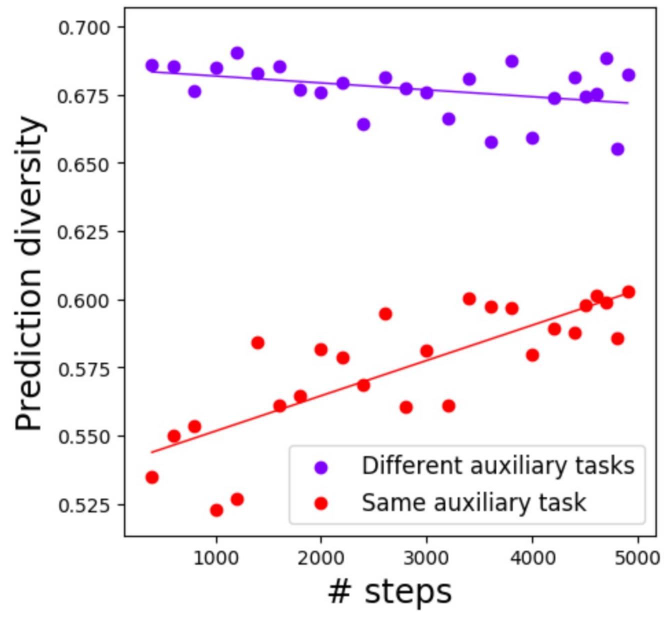
Yes, when the auxiliary tasks learn rich features, that help for the target task.

3. Diversity

Yes !! huge gain in diversity caused by different initialization and remains along fine-tuning on the target task.



[Phang2018] Sentence encoders on stilts: Supplementary training on intermediate labeleddata tasks.
[Choshen2022] Where to start? analyzing the potential value of intermediate models.



New SoTA on DomainBed

- Use the other datasets from DomainBed as the auxiliary datasets.
- No inference overhead.
- No training overhead if auxiliary weights are recycled.

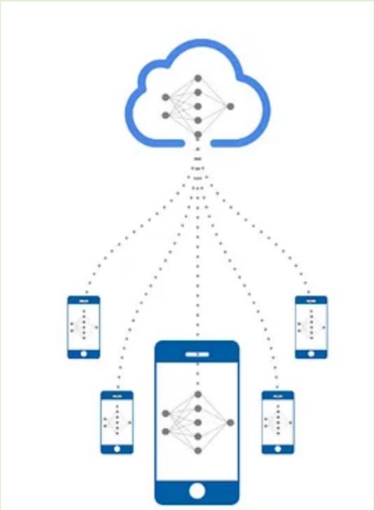
Algo	Strategy	VLCS	PACS	OH	Terra	DomainNet	Avg
ERM	Standard ft	78.1	85.9	69.4	50.4	44.3	65.6
Soups	WA of networks with same inits	78.4	88.7	72.1	51.4	47.4	67.6
Inter-training	Auxiliary task	77.7	89.0	69.9	46.7	44.5	65.6
Ratatouille	**WA of networks with intertrain**	**78.5**	**89.5**	**73.1**	**51.8**	**47.5**	**68.1**

Updatable machine learning [Raffel2023]

Data privacy concerns

- Only share weights,
- data are kept private

⟹ scalable federated strategy.



Collaboration and open-source

Foundation models come with risk of:

- Centralization.
- Lack of reproducibility.
- Two-speed research.

⟹ new collaborative solutions.

github.com/r-three/git-theta

Embarrassingly simple parallelization

Compute parallelism [Wortsman2022]!

- Simple engineering.
- Efficiency and training time.
- No waste: leverage all runs.
- Better compute scaling laws ?

	Average updates per second, normalized (↑)		
	fully synchronized (TRANSFORMER-LM)	partially synchronized (DEMIX)	BTM: embarrassingly parallel (branched ELMs)
125M	1.00	1.01	1.05
350M	1.00	1.11	1.23
750M	1.00	1.01	1.27
1.3B	1.00	0.97	1.33

[Raffel2023] A Call to Build Models Like We Build Open-Source Software. ACM.
[Wortsman2022] lo-fi: distributed fine-tuning without communication? JMLR.

Conclusion

- Linear mode connectivity across weights fine-tuned on different tasks
- New ratatouille strategy for out-of-distribution generalization
- Code is available: github.com/facebookresearch/ModelRatatouille

Thank you for your attention

SCIENCES SORBONNE UNIVERSITÉ

Meta