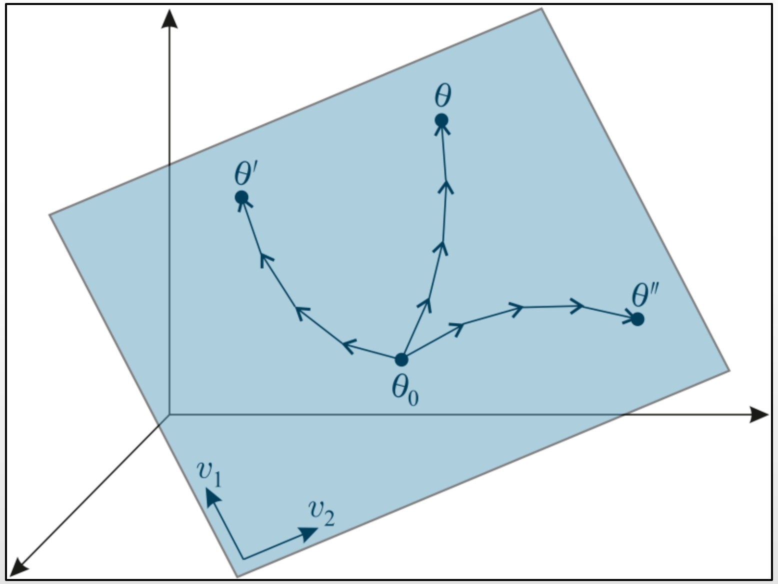
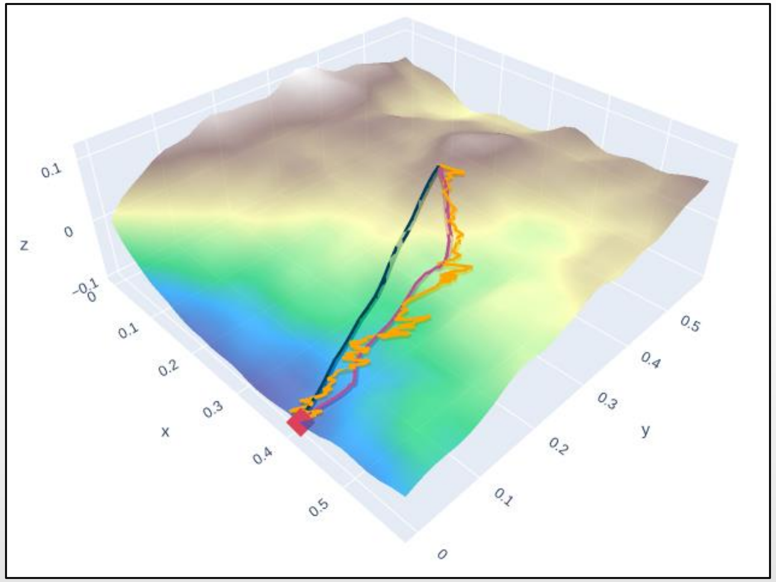
Institute for
Machine Learning

Few-Shot Learning by Dimensionality Reduction in Gradient Space



Maximilian Beck

PhD Seminar Talk



Paper by Martin Gauch, Maximilian Beck, Thomas Adler, Dmytro Kotsur, Stefan Fiel, Hamid Eghbal-zadeh, Johannes Brandstetter, Johannes Kofler, Markus Holzleitner, Werner Zellinger, Daniel Klotz, Sepp Hochreiter, Sebastian Lehner

Agenda

Institute for Machine Learning

- Introduction
- Problem Setup: Few-shot learning
- Method: Subspace Gradient Descent
- Experiments
- Summary

Motivation

Institute for Machine Learning

- Deep learning needs a lot of data to succeed
- If a large amount of data is available:
  Deep learning often outperforms other methods or even humans
- For many real world applications there is often not enough data available
  - Examples: Industrial Applications, Autonomous Driving, Environment Modeling
- Gives rise to the research areas such as Few-shot- and Meta-learning

Supervised- vs. Meta-Learning

(Hospedales et al., 2020)

Institute for Machine Learning

Model: $\hat{y} = f_\theta(x),\ \theta \in \mathbb{R}^d$

Across-task-/Meta-knowledge: ω

Data distribution: $\mathcal{D}_{train},\ \mathcal{D}_{test} \sim p(x, y)$

$\mathcal{D}_* = \{(x_1, y_1), ..., (x_N, y_N)\}$

Training:

$$\theta^* = \arg\min_\theta \mathcal{L}(\mathcal{D}_{train}; \theta, \omega)$$

Testing:

$$\mathcal{L}(\mathcal{D}_{test}; \theta^*, \omega)$$

Task distributions: $p_{train}(\mathcal{T}),\ p_{test}(\mathcal{T})$

$\mathcal{T}_i \sim p_*(\mathcal{T})$ $\mathcal{T}_i = \{\mathcal{D}_*, \mathcal{L}\}$ $\mathcal{D}_* = \{\mathcal{D}_*^{support}, \mathcal{D}_*^{query}\}$

Meta-training:

$$\omega^* = \arg\min_\omega \mathop{\mathbb{E}}_{\mathcal{T} \sim p_{train}(\mathcal{T})} \mathcal{L}(\mathcal{D}^*_{train}, \omega)$$

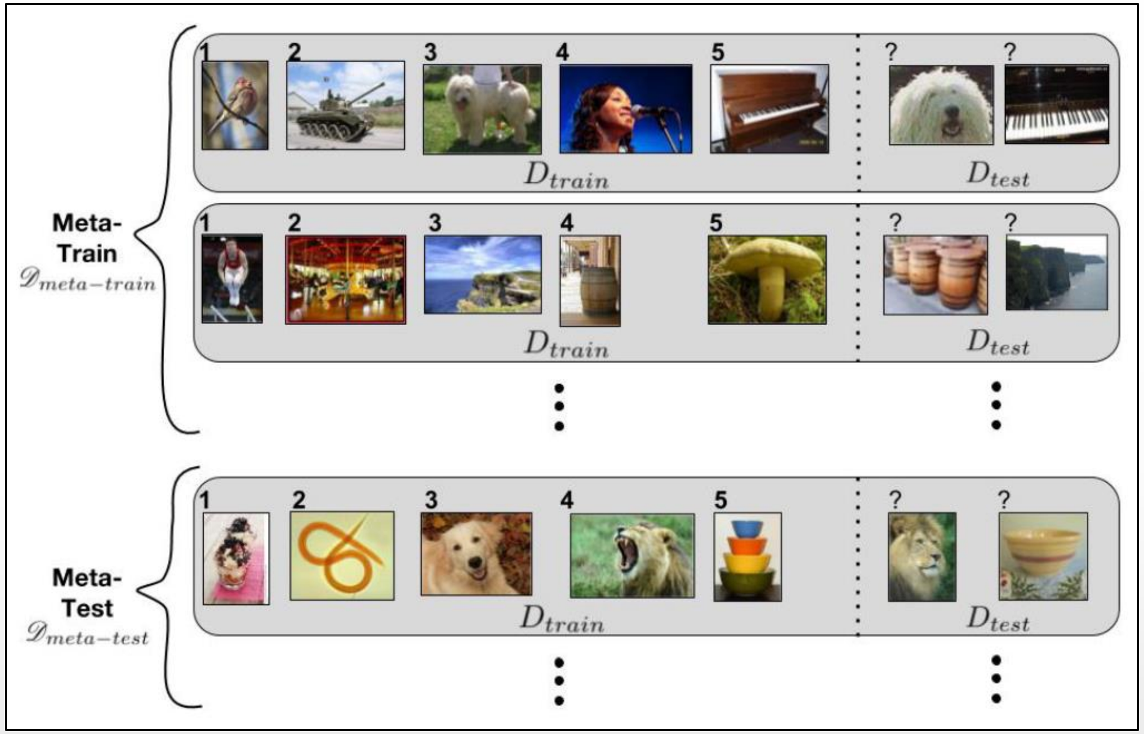
Meta-testing:

$$\theta^{*(i)} = \arg\min_\theta \mathcal{L}(\mathcal{D}^{support}_{test,(i)}; \theta, \omega^*)$$

$$\mathcal{L}(\mathcal{D}^{query}_{test,(i)}; \theta^{*(i)}, \omega^*)$$

Few-shot Learning

- Typical example: Few-shot image classification
- N-way - K-shot scenario:
  support set consists of N classes
  with K images each
- In our paper we consider predictions of dynamical systems behavior
- Support and query set are short sequences of system behavior, different tasks are different systems
  (more on this later)



(Ravi et al., 2017)

Subspace Gradient Descent (SubGD) (I)

Institute for Machine Learning

Motivation:

- Gradient descent happens in a small subspace. (Li et al., 2018; Gur-Ari et al., 2018)
- Restricting learning to certain low-dimensional subspaces does not deteriorate performance, and can even improve performance in case of "lottery subspaces". (Larsen et al., 2022)
  - "Lottery subspaces": Subspace consists of the top r principal components of an entire training trajectory for a single task

Hypothesis:

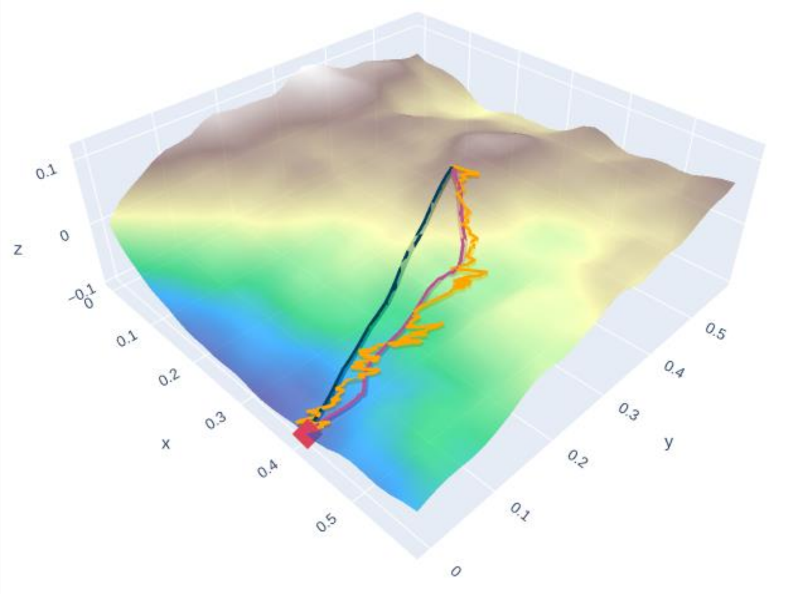
- A subspace shared across different Few-shot learning tasks might lead to better sample efficiency and generalization on new tasks

Subspace Gradient Descent (SubGD) (II)

Institute for Machine Learning

Idea:

- Restrict gradient descent to a r dimensional subspace that is learned during meta-training.
- Modify update rule:
  - SGD: $\theta \leftarrow \theta - \eta g$
  - SubGD: $\theta \leftarrow \theta - \eta C g, \quad C = PSP^\top$
- Preconditioning matrix C can be decomposed:
  - Projection matrix: $P \in \mathbb{R}^{d \times r}$
  - Scaling Matrix: $S \in \mathbb{R}^{r \times r}$



Stochastic Gradient:

$$g = \nabla_\theta \frac{1}{|\mathcal{B}|} \sum_{x,y \in \mathcal{B}} \mathcal{L}\left(f_\theta(x), y\right)$$

Neural Network Parameters: $\theta \in \mathbb{R}^d$

Learning rate: η

Subspace Gradient Descent (SubGD) (III)

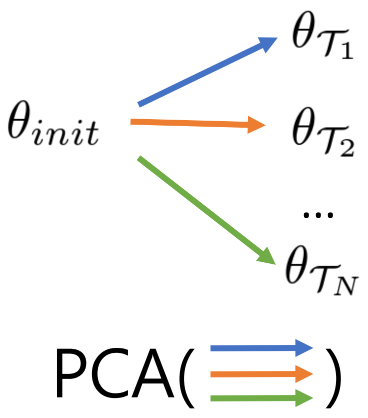
How do we construct the subspace, i.e. the matrices P and S?

- Subspace consists of the top r principal components of the fine-tuning trajectories on meta-train tasks.
- In practice: We compute P and S by an Eigendecomposition of the (uncentered) covariance matrix consisting of the weight differences between initialized and finetuned models. (will be explained in more detail on the next slide)
  - P are the eigenvectors corresponding to the top r eigenvalues of the covariance matrix.
  - S is a diagonal matrix with the top r eigenvalues of the covariance matrix on its diagonal.
- Important subspace directions get higher weight, due to scaling matrix S

Subspace Gradient Descent (SubGD) (IV)

Training procedure:

Meta-training			Meta-testing
Pretraining	**Finetuning**	**Gridsearch**	Evaluate performance of optimal meta-parameters
Supervised, MAML / Reptile, Random	(see diagram)	learning rate and number of update steps	$\omega^* = \{\theta_{init}, P, S, \eta, N_{steps}\}$ on test tasks.
θ_{init}	$P,\ S$	$\eta,\ N_{steps}$	



Summary of Ablations on Sinusoid

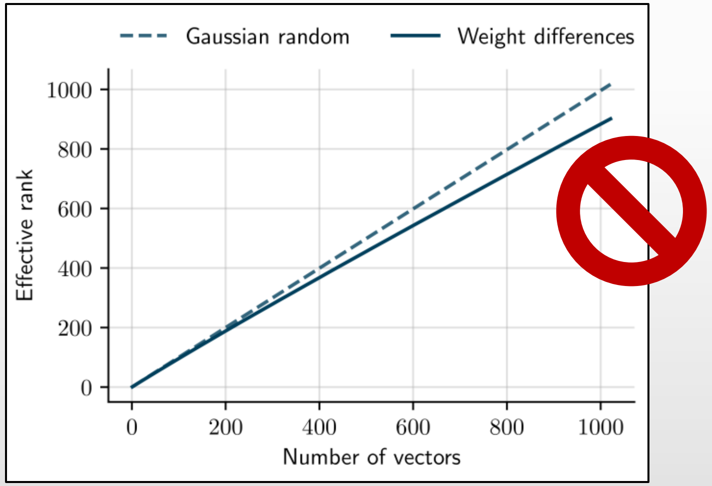
- SubGD can be applied to different pre-trained initializations
  - Examples: Random initialization, Supervised pre-training, Meta-learned initializations
  - Meta-learned initializations perform better than random or supervised pre-trained initializations
  - SubGD can benefit from this

- SubGD chooses the effective subspace dimensionality by weighting with eigenvalues
  - No tuning of the subspace dimension necessary

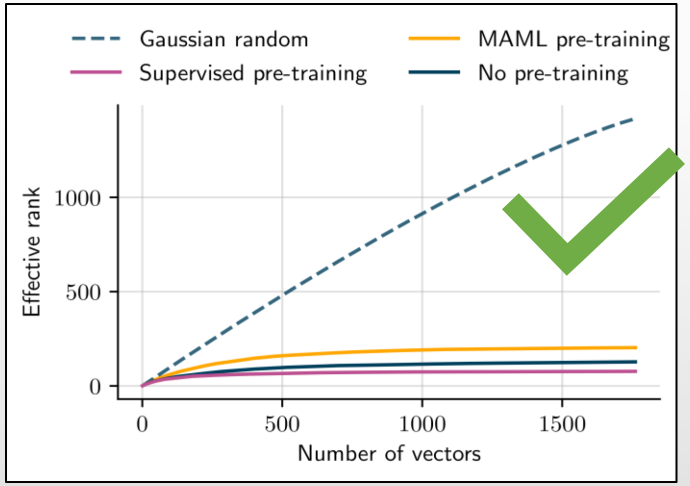
- SubGD's subspace based on PCA of the update directions outperforms simpler subspace variants
  - Examples: Random directions, Diagonal preconditioning

Limitations

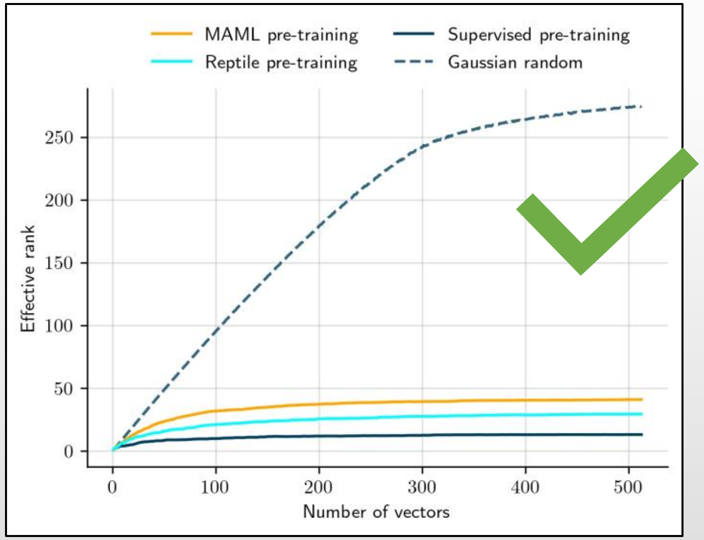
- We expect SubGD to work, when...
  - the test tasks are not too different from the training tasks (they share some common structure)
  - a shared subspace on the training tasks can be found with gradient descent



Mini-Imagenet



Sinusoid

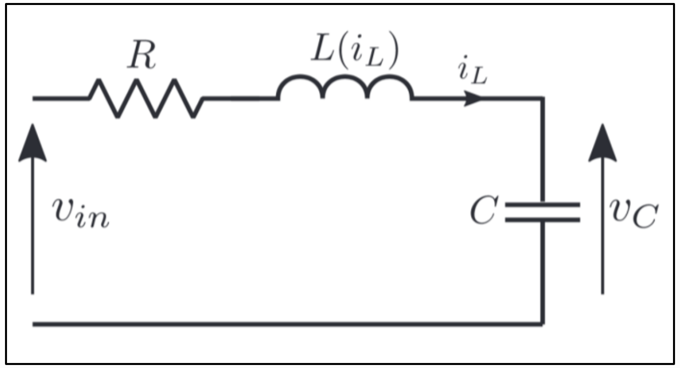


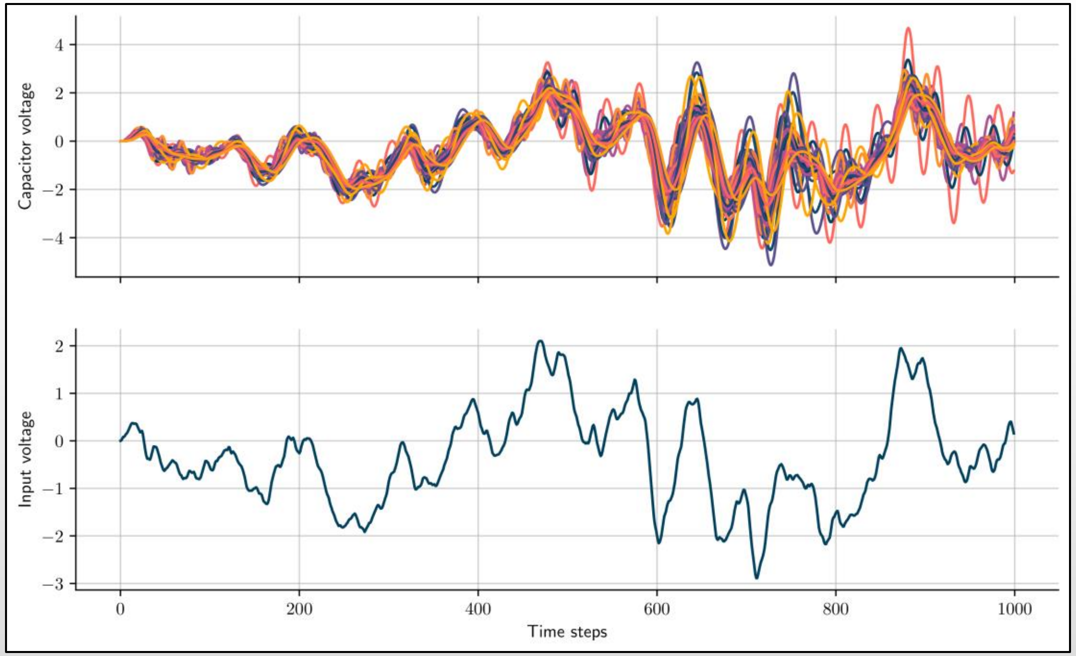
Non-linear RLC

Non-linear RLC – Experiment Setup

Institute for Machine Learning

- Electrical circuit consisting of resistance R, inductance L(i), and capacitance C
- Generate train and test tasks by sampling random parameter values for R, L and C
- Generate ground-truth data by simulating the system behaviour of each parameter combination for random input signals
- Goal: Learn a model that predicts the output voltage given the input voltage
  - Problem of System Identification



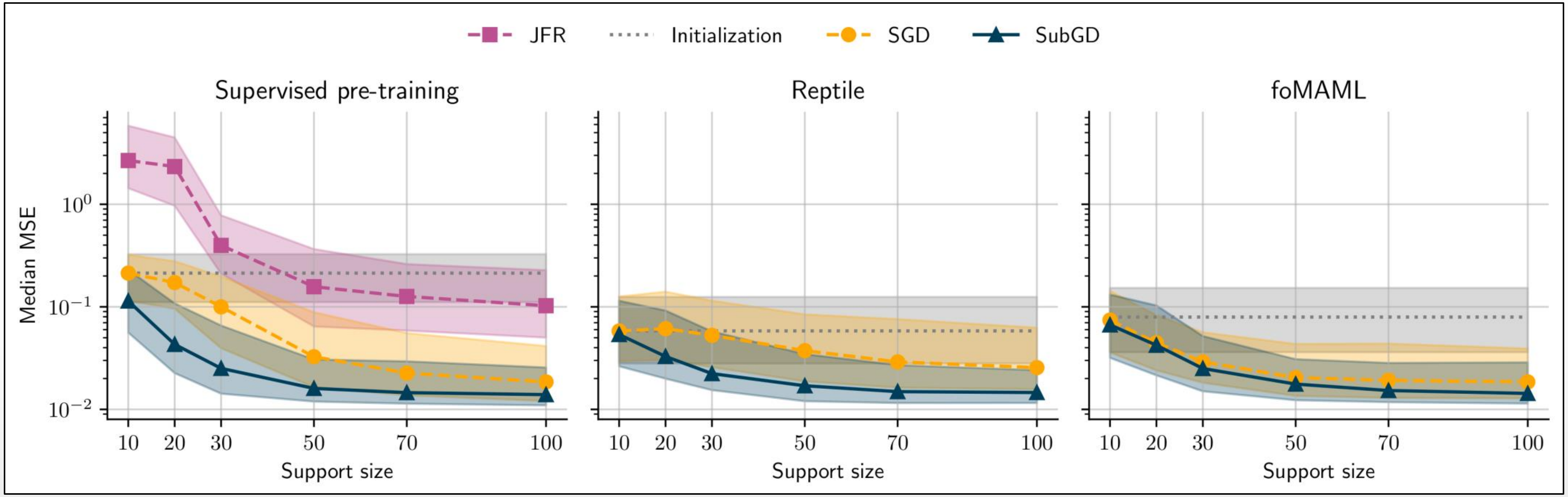


Output voltage of 50 different test systems

Non-linear RLC – Experiment Results

Institute for Machine Learning

Median MSE over 256 test tasks for different support sizes and pretraining strategies:



MetaSGD and Meta-Curvature also employ preconditioning

Method	10-shot	20-shot	30-shot	50-shot	70-shot	100-shot
MetaSGD	0.072	0.048	0.035	0.021	0.023	0.022
Meta-Curvature	0.062	**0.038**	0.029	0.020	0.018	0.017
Reptile+SubGD	**0.054**	**0.033**	**0.022**	**0.017**	**0.015**	**0.015**

Summary

- Comparison between Supervised- and Meta-learning setting
- Few-shot learning setting
- Subspace Gradient Descent
- Experiment results on the RLC dataset

References

Hospedales, Timothy, Antreas Antoniou, Paul Micaelli, and Amos Storkey. "Meta-Learning in Neural Networks: A Survey." *ArXiv:2004.05439 [Cs, Stat]*, November 7, 2020. http://arxiv.org/abs/2004.05439.

Larsen, Brett W., Stanislav Fort, Nic Becker, and Surya Ganguli. "How Many Degrees of Freedom Do We Need to Train Deep Networks: A Loss Landscape Perspective." *ArXiv:2107.05802 [Cs, Stat]*, February 3, 2022. http://arxiv.org/abs/2107.05802.

Li, Chunyuan, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. "Measuring the Intrinsic Dimension of Objective Landscapes." arXiv, April 24, 2018. http://arxiv.org/abs/1804.08838.

Gur-Ari, Guy, Daniel A. Roberts, and Ethan Dyer. "Gradient Descent Happens in a Tiny Subspace." arXiv, December 11, 2018. http://arxiv.org/abs/1812.04754.

Ravi, Sachin, and Hugo Larochelle. "OPTIMIZATION AS A MODEL FOR FEW-SHOT LEARNING," 2017, 11.

Forgione, Marco, and Dario Piga. "Continuous-Time System Identification with Neural Networks: Model Structures and Fitting Criteria." *European Journal of Control* 59 (May 2021): 69–81. https://doi.org/10.1016/j.ejcon.2021.01.008.

Forgione, Marco, Aneri Muni, Dario Piga, and Marco Gallieri. "On the Adaptation of Recurrent Neural Networks for System Identification." arXiv, January 21, 2022. https://doi.org/10.48550/arXiv.2201.08660.

Non-linear RLC - Details

Institute for Machine Learning

- System equation of the RLC curcuit:

$$\begin{pmatrix} \dot{v}_C(t) \\ \dot{i}_L(t) \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{C} \\ -\frac{1}{L(i_L)} & -\frac{R}{L(i_L)} \end{pmatrix} \begin{pmatrix} v_C(t) \\ i_L(t) \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{L(i_L)} \end{pmatrix} v_{in}(t)$$

- Non-linear inductance:

$$L(i_L) = L_0 \left[0.9 \left(\frac{1}{\pi} \arctan(-5|i_L| - 5) + 0.5 \right) + 0.1 \right]$$

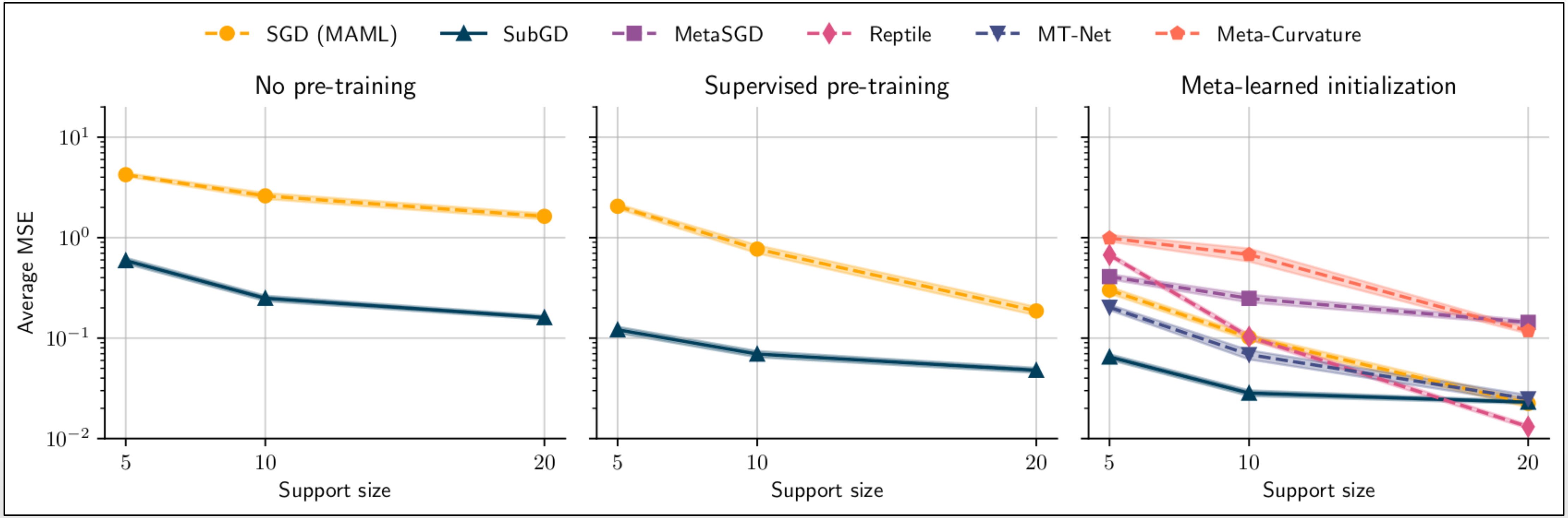
- Approximate dynamical system:

$$\dot{\hat{x}} = f_\theta(\hat{x}, u)$$

$$\hat{x}(t; \theta, x_0) = \mathrm{ODEINT}\left(t, f_\theta(\,\cdot\,,\,\cdot\,), u(\,\cdot\,), x_0\right)$$

Sinusoid Results (I)

Comparison of different pre-training strategies for SubGD



Sinusoid Results (II)

Institute for Machine Learning

Performance of SubGD for different subspaces, i.e. different construction mechanisms of the projection matrix