Application of self-paced learning for noisy meta-learning

Árpád Aszalós

Responsible Professor: Matthijs Spaan

Supervisor : Joery de Vries

Contact : Aszalos@student.tudelft.nl | arpi.aszalos@gmail.com

1. Introduction

- Meta-learning, aims to create models that adapt quickly to new tasks by learning transferable representations across them, improving data–efficiency and generalization.
- Noisy data is a challenge in machine learning, addressed in traditional learning but lacking attention in meta-learning
- Self-Paced Learning (SPL) is a Curriculum Learning (CL) strategy that progresses from easier to harder examples based on model performance, and showed effectiveness in noisy data, and restricted learning resource scenarios.[1][5]
- There is a gap in research on noisy meta-learning, with most studies focusing on managing noise during testing.[3]
- Limited research exists on combining meta-learning with CL/SPL, leaving unexplored areas in their application to noisy meta-learning scenarios.

2. Research question

How can Self-Paced learning aid meta-learning in presence of noisy training data?

- How does SPL affect the meta-training trajectory in noisy training environments?
- How does incorporating SPL influence the generalization performance for within training task and out ouf of training task distributions under clean and noisy test conditions, considering prior noisy training environments?

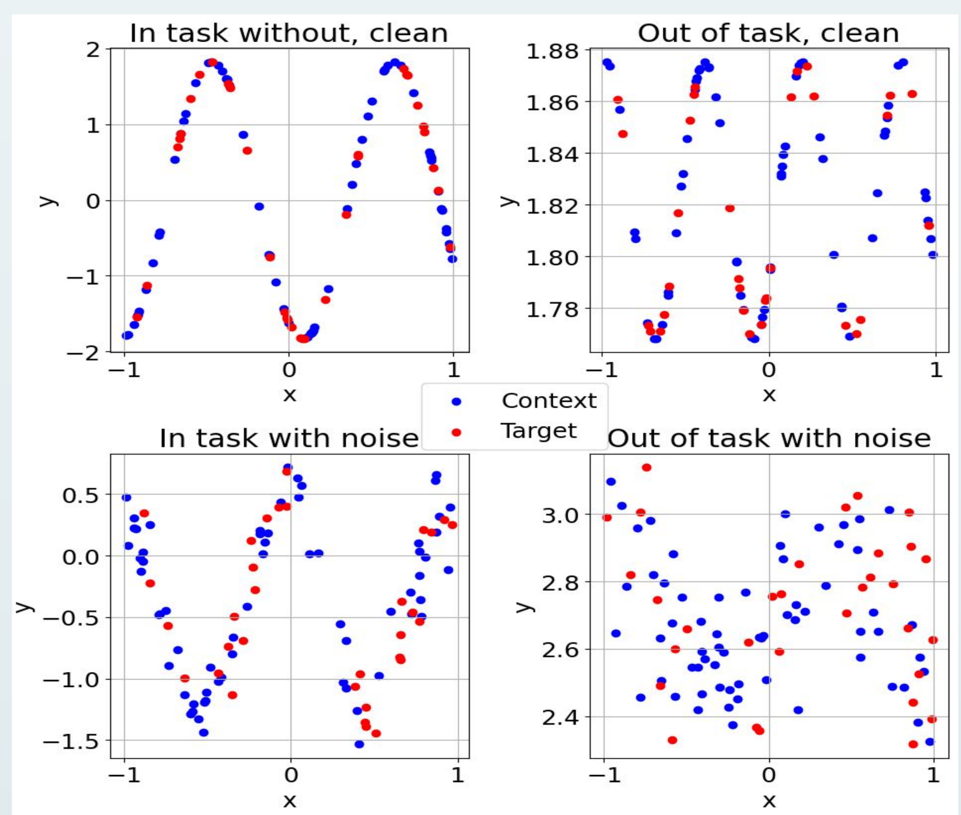

Figure 3. Showing example training (In task) and test (Out of task) regression task with s=0.2 Noise added

3. Background

Supervised Learning formulation of Meta-learning

Traditional supervised learning algorithm

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}, \quad A_t : \mathcal{D} \to f(x), \quad f(x_T) = y_T$$

A meta-dataset is a collection of tasks

$$\mathcal{D}_{\text{task}} = (\mathcal{D}, x_T, y_T), \quad M_t = \{\mathcal{D}_{\text{task}_i}\}_{i=1}^{N_{\text{task}}}$$

Meta-learning algorithm formalised in supervised learning context with meta-dataset (dataset of datasets).

$$A_m : M_t \to (D \to f(x; D)), \quad f(x_T; D) = y_T$$

Neural Process Model (NP)[4]

Combines neural networks' computational efficiency and data fitting capability with Gaussian Processes' ability to produce uncertainty estimates around target predictions.

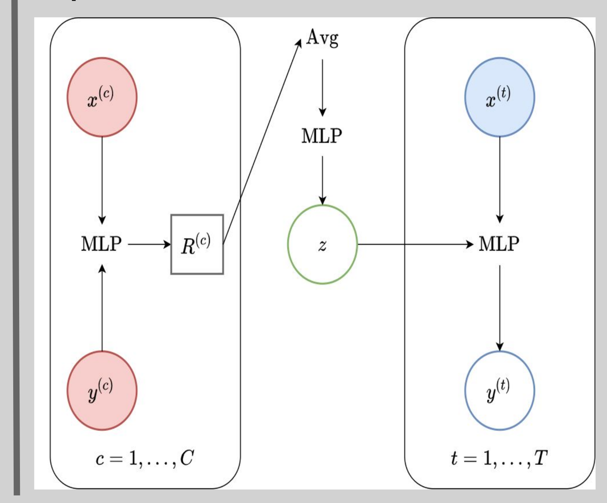

Figure 1. Showcasing the computational graph of NP used to meta-learn over regression tasks. [6]

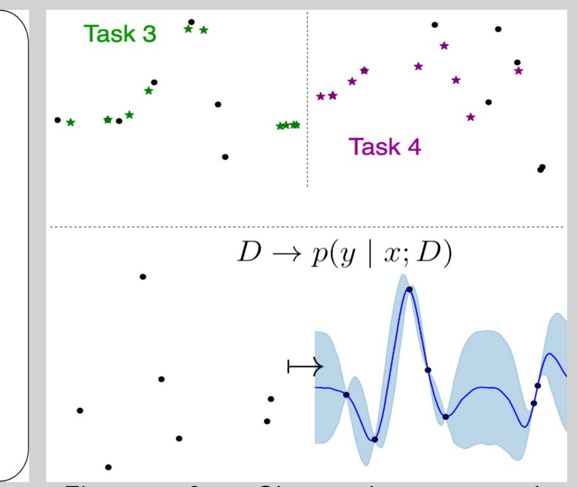

Figure 2. Showcasing example regression tasks and the output produced by NP to solve a regression task. Original image from Dubois [6].

Self-Paced Learning (SPL)

SPL is formulated as weighted loss included in the learning objective, with parameter **λ** that determines the threshold for lower loss samples to be trained on.

$$\min_{\mathbf{w},\theta} \sum_{i=1}^{N} w_i \ell(f(\mathbf{x}_i;\theta), y_i) - \lambda \sum_{i=1}^{N} w_i$$

The SPL weights are optimised by:

$$w_i^* = \begin{cases} 1, & \text{if } \ell_i < \lambda \\ 0, & \text{otherwise} \end{cases}$$

Then, with fixed weights we perform gradient descent step on the parameters.

4. Experimental setup

Dataset

- 128.000 sinusoidal regression tasks (Fourier Series) (Figure 3.) $\mathcal{C} = \{(x_i, y_i)\}_{i=1}^{64}, \quad \mathcal{T} = \{(x_j, y_j)\}_{j=1}^{32}$
- 3 Clean / Noisy training data split setups
  - 0% Noisy / 100% Clean
  - 30% Noisy / 70% Clean
  - 60% Noisy / 40% Clean

Added Noise

$$\hat{y}_i = y_i + \epsilon_i \cdot s, \quad \epsilon_i \sim N(0,1), \quad s = 0.2$$

Evaluation metrics

$$\text{ECE} = -\frac{1}{N}\sum_{i=1}^{N} \log P(y_i|x_i) \quad \text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

Evaluations

- Intra training evaluations
  - In-task , out of task distribution , Clean / Noisy (s=0.2) on both
- Post-training evaluations
  - In task
    - s ∈ {0.0,0.2,0.3,0.4,0.5,0.6}
  - Out of task
    - higher - amplitude, periodicity, number of sine components, with s ∈{0.0,0.2,0.4}

SPL

- CurML[2] implementation adapted to JAX
- start rate of 10% ; grows to full training set size over 5 epochs

6. Limitations

- Sole focus on sinusoidal regression task restrict real-world applicability, due to the data complexity differences that might exist.
- Narrow scope of clean/noisy training splits, noise levels, and noise types, left more complex noisy environments unexplored.
- Lack of ablations in varied curriculum hyperparameter settings, to establish potential sensitivity of SPL to certain noisy environments.
- Lack of comparative analysis with other CL approaches and general techniques against noise robustness.

5. Results and Conclusion

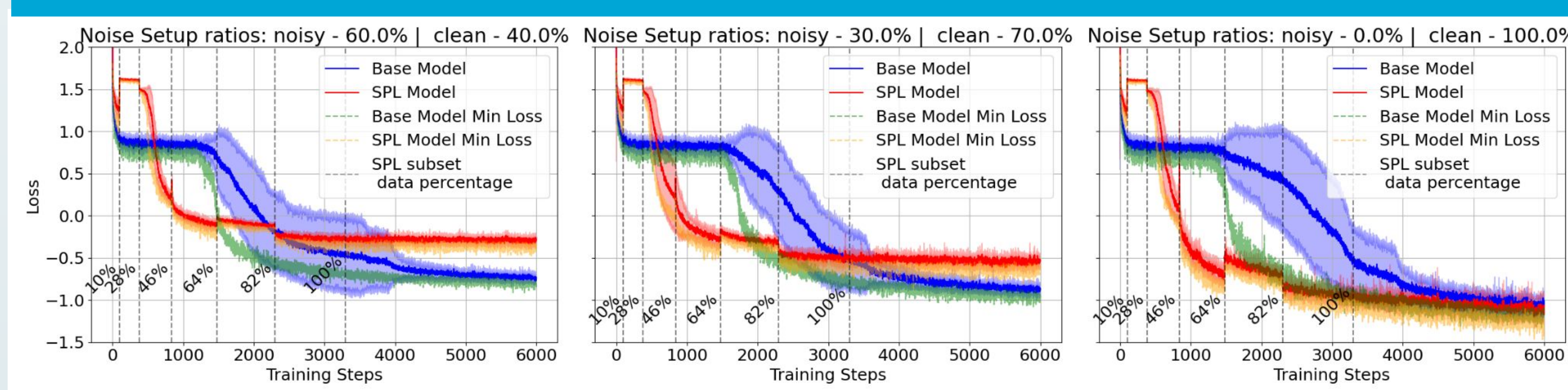

Figure 4. ***Training losses*** *averaged over 10 seeds per training step, showing the standard deviation as a fill. The dashed line show the curriculum subset progression over the training with start_rate=0.1 , growth_epochs=5.*

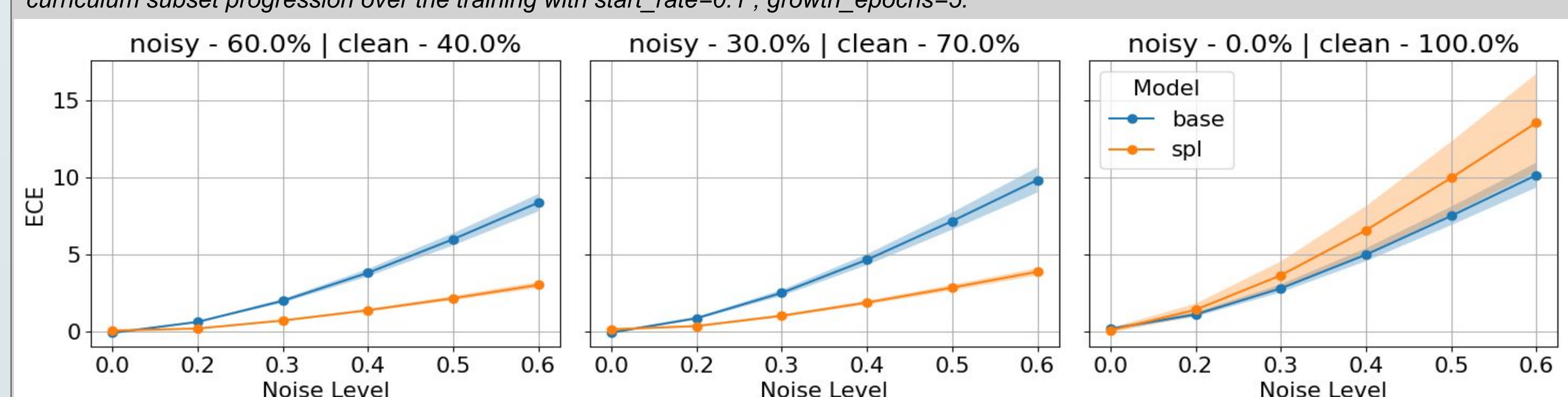

Figure 5. ***Post training in task distribution ECE performances*** *in different noisy training setups, and with increasing level of added noise, averaged over 10 runs, fill showing 95% CI.*

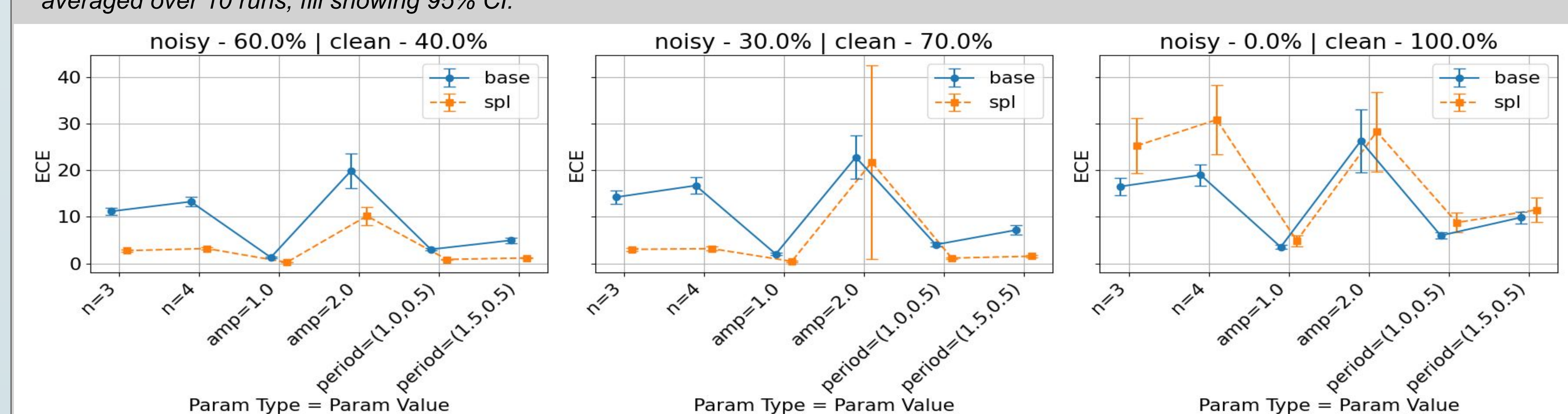

Figure 6. ***Post training Out of task distribution with 0.2 Noise, ECE performances*** *averaged over the 10 runs with 95%CI. The graph shows the improved generalizatoin capability of the model due to SPL in the noisy training splits.*

Conclusions

- SPL affected the meta-training by improving loss convergence and stabilising the variance of per training step losses. (Figure 4.)
- SPL showed no performance improvement in fully clean data, and improvement remained marginal between different noisy splits (Figure 5. ,6.).
- Incorporation of SPL improved noise robustness for in task distribution performances with increasing noise levels. (Figure 5.)
- Incorporation of SPL improved overall out of task generalization performance on both clean and noisy data. (Figure 6.)
- Improvements in performance are mostly seen in ECE, meaning that the model's uncertainty has been decreased due to SPL
- Overall SPL, as a CL strategy, showed applicability for noisy meta-learning along the same lines as shown in traditional learning settings:
  - Improved convergence[1]
  - Improve generalization[1]

References

X. Wang, Y. Chen, and W. Zhu, "A Survey on Curriculum Learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–1, 2021, doi: 10.1109/TPAMI.2021.3069908.] [1]

Y. Zhou *et al.*, "CurML: A Curriculum Machine Learning Library," in *Proceedings of the 30th ACM International Conference on Multimedia*, Lisboa Portugal: ACM, Oct. 2022, pp. 7359–7363. doi: 10.1145/3503161.3548549. [2]

J. Galjaard, "Meta-Learning with Label Noise: A Step Towards Label Few-Shot Meta-Learning with Label Noise," Master's thesis, Delft University of Technology, 2023. [Online]. Available:http://resolver.tudelft.nl/uuid:65aa4a0c-d2d6-44f1-bda8-6dd093488f40 [3]

M. Garnelo *et al.*, "Neural Processes." arXiv, Jul. 04, 2018. doi: 10.48550/arXiv.1807.01622. [4]

T. Gong, Q. Zhao, D. Meng, and Z. Xu, "Why curriculum learning & self-paced learning work in big/noisy data: A theoretical perspective,"[5] *BDIA*, vol. 1, no. 1, pp. 111–127, Aug. 2015, doi: 10.3934/bdia.2016.1.111.

Y. Dubois, J. Gordon, and A. Y. Foong, "Neural Process Family." Sep. 2020. [Online]. Available: http://yanndubs.github.io/Neural-Process-Family/ [6]