A Trajectory-Based Bayesian Approach to Multi-Objective Hyperparameter Optimization with Epoch-Aware Trade-Offs

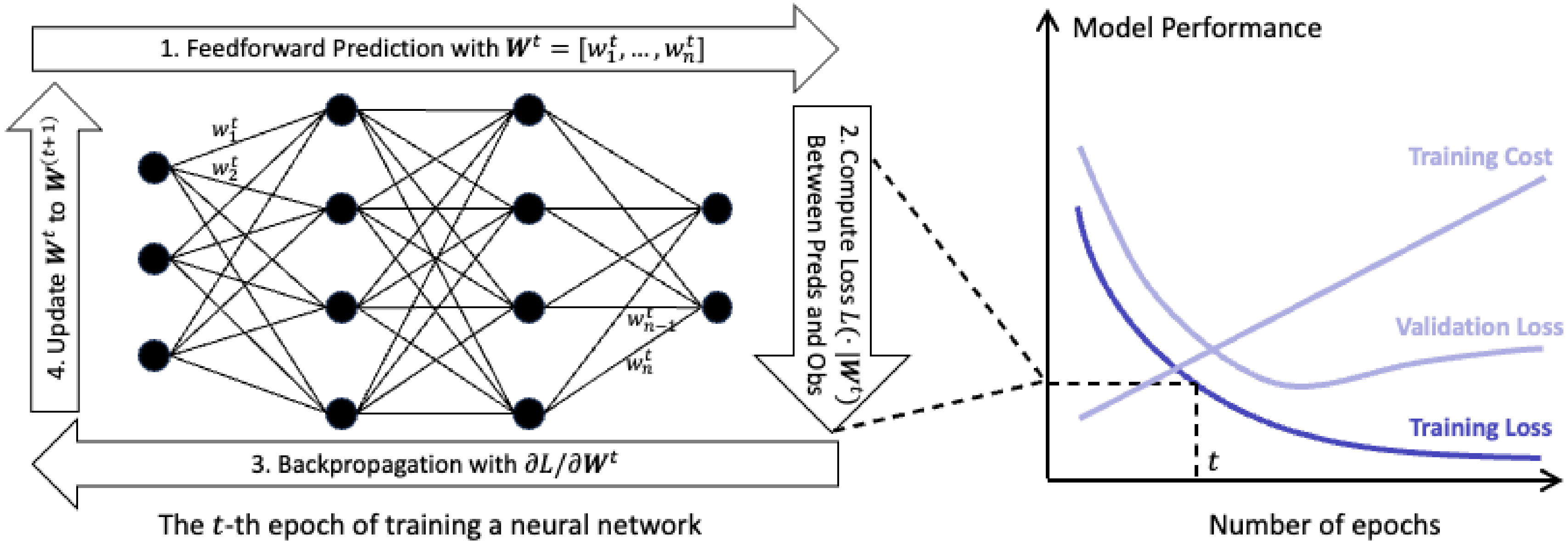
Wenyu Wang[1], Zheyi Fan[2,3], Szu Hui Ng[1,*]

[1]Industrial Systems Engineering and Management, National University of Singapore, Singapore
[2]Academy of Mathematics and System Science, Chinese Academy of Sciences, China
[3]School of Mathematical Sciences, University of Chinese Academy of Sciences, China

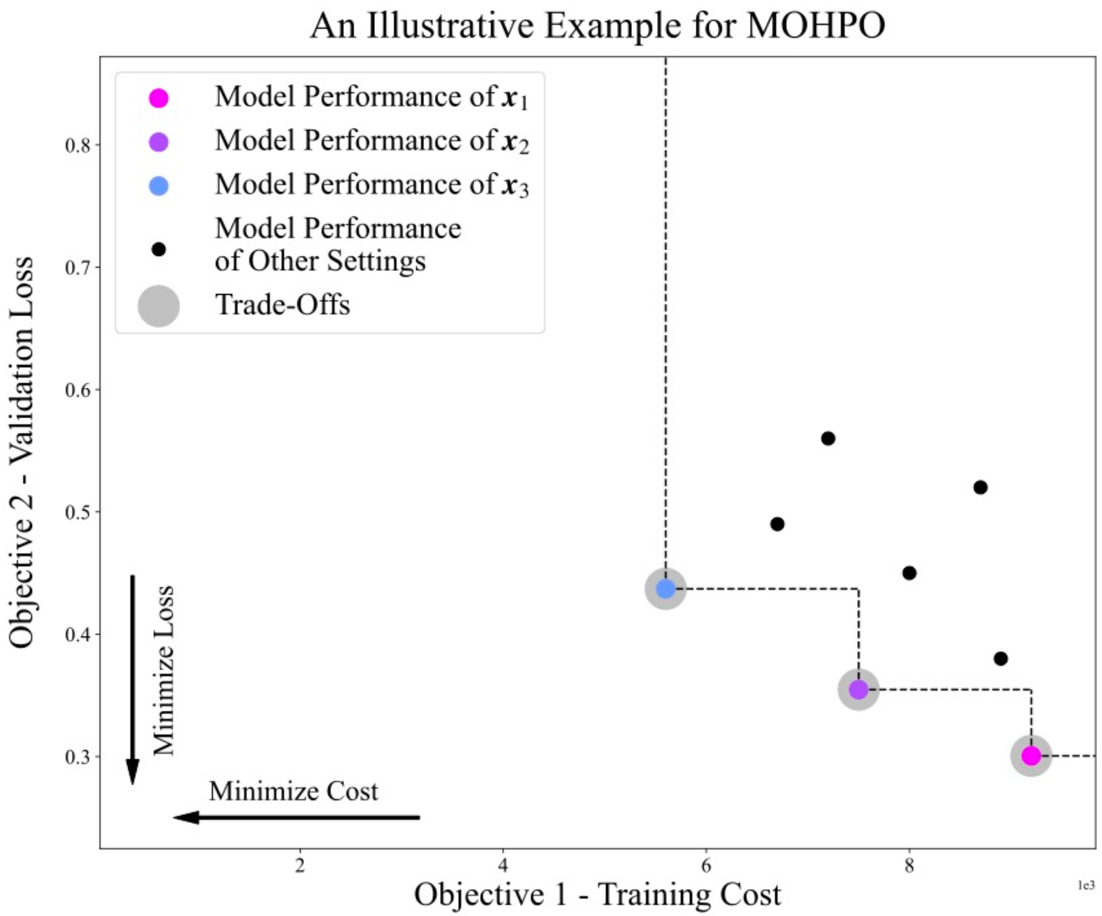
Outline

Hyperparameter Optimization

- Addressing Hyperparameter Optimization (HPO) problem has long been challenging as it involves resource-intensive model training that prevents optimizers from exhaustively exploring the hyperparameter space.
- More recently, the surge in the demand for HPO is not only in pursuit of prediction accuracy but also for ensuring the efficiency and robustness of models, which leads to Multi-Objective HPO (MOHPO).



MOHPO with Iterative Learning Procedure

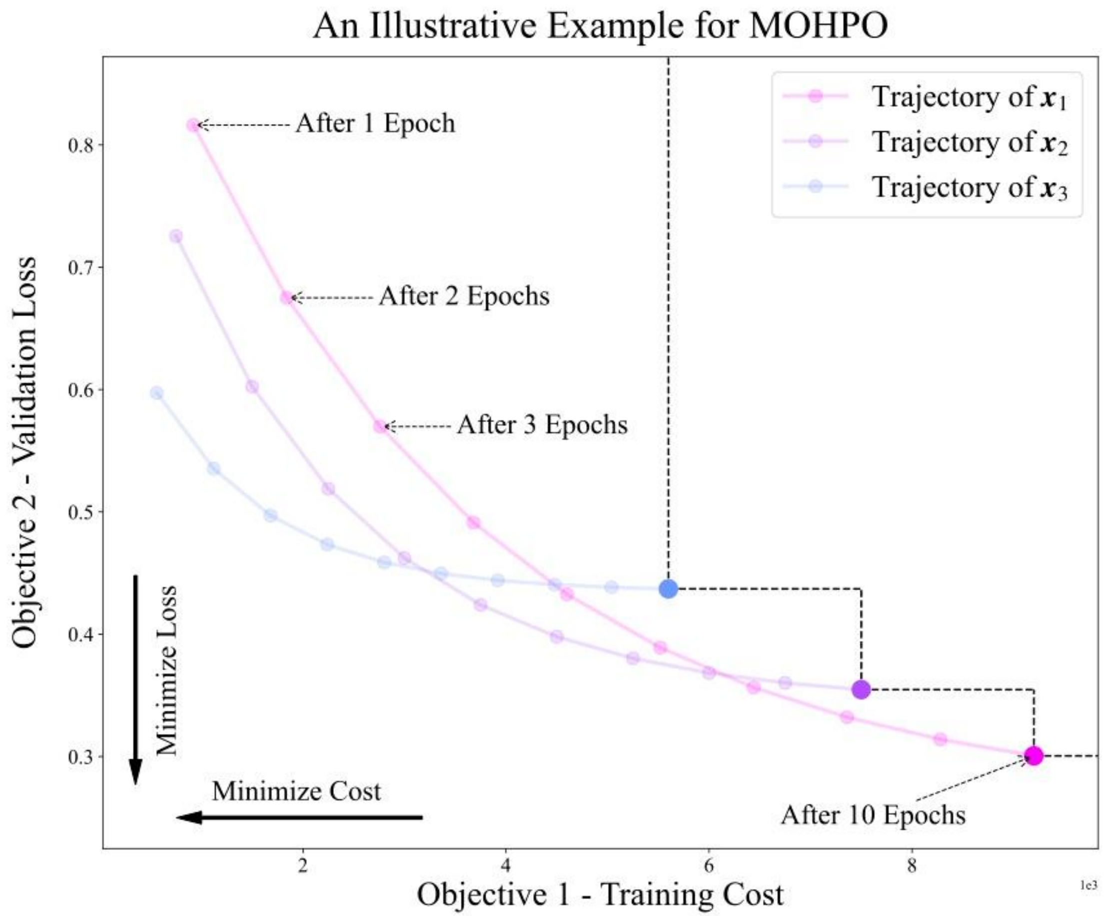


- Minimizing an MOHPO is equivalent to finding trade-offs between performances at the end of training.

$$\min_{\boldsymbol{x} \in \mathbb{X}} \boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), \ \ f_2(\boldsymbol{x})]$$

  - A solution dominates another if it is no worse in all objectives and strictly better in at least one.
  - The Pareto-optimal front consists of all non-dominated solutions

MOHPO with Iterative Learning Procedure



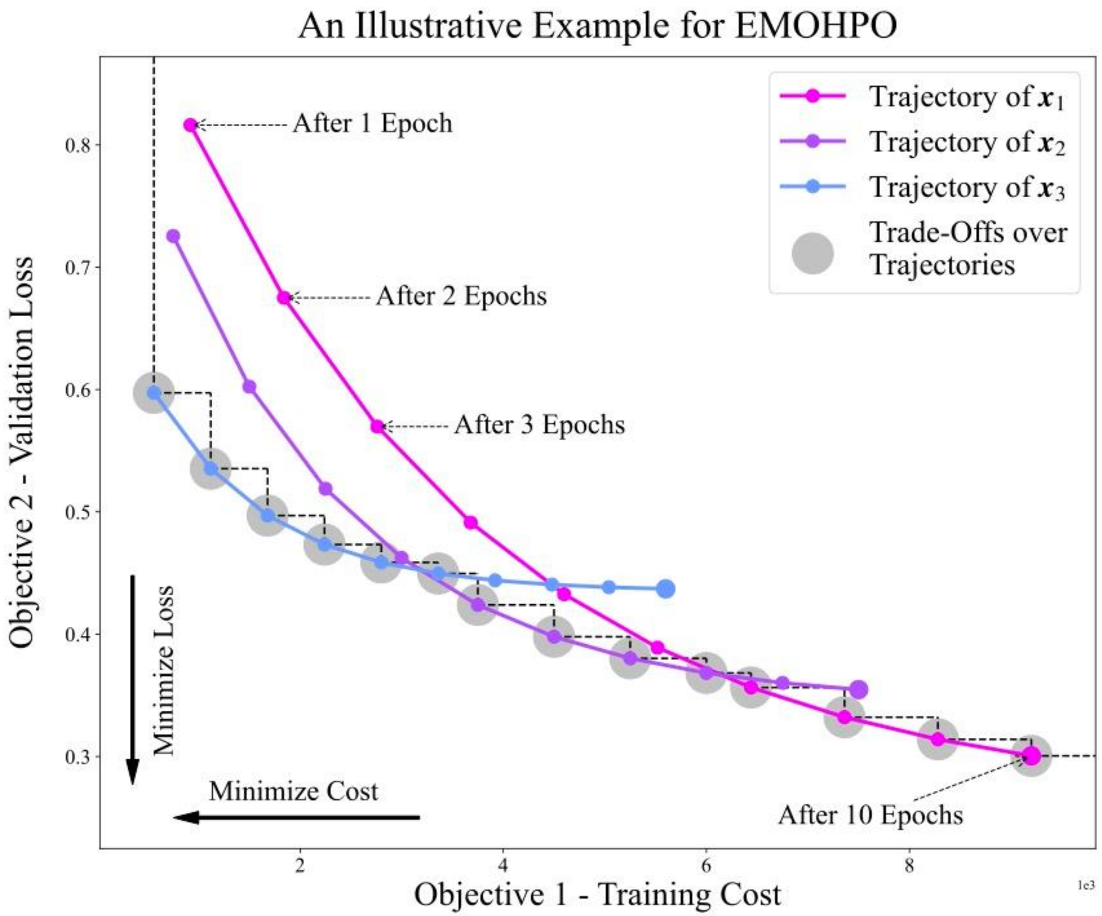
- Minimizing an MOHPO is equivalent to finding trade-offs between performances at the end of training.

$$\min_{\boldsymbol{x} \in \mathbb{X}} \boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), \;\; f_2(\boldsymbol{x})]$$

  - A solution dominates another if it is no worse in all objectives and strictly better in at least one.
  - The Pareto-optimal front consists of all non-dominated solutions

- Training ML model is an iterative learning procedure, allowing epoch-wise tracking on model performances.
  - Does a trade-off emerge when the number of training epochs is fewer than the maximum allowed?

MOHPO with Iterative Learning Procedure



- Minimizing an MOHPO is equivalent to finding trade-offs between performances at the end of training.

$$\min_{\boldsymbol{x} \in \mathbb{X}} \boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), \;\; f_2(\boldsymbol{x})]$$

  - A solution dominates another if it is no worse in all objectives and strictly better in at least one.
  - The Pareto-optimal front consists of all non-dominated solutions

- Training ML model is an iterative learning procedure, allowing epoch-wise tracking on model performances.
  - Does a trade-off emerge when the number of training epochs is fewer than the maximum allowed?
  - E.g., (1) Partially-trained model; (2) Overfitting.

$$\min_{(\boldsymbol{x},t) \in \mathbb{X}\times\mathbb{T}} \boldsymbol{f}(\boldsymbol{x},t) = [f_1(\boldsymbol{x},t), \;\; f_2(\boldsymbol{x},t)]$$

Outline

Enhanced Multi-Objective Hyperparameter Optimization Problem

Consider the sequential minimization of an EMOHPO in the following form:

$$\min_{(\boldsymbol{x},t)\,\in\,\mathbb{X}\times\mathbb{T}} \boldsymbol{f}(\boldsymbol{x},t) = [f_1(\boldsymbol{x},t),\ldots,f_k(\boldsymbol{x},t)]\,, \tag{2}$$

- $\boldsymbol{x}$ denotes a d-dimensional hyperparameter setting with $\boldsymbol{x} \in \mathbb{X} \subset \mathbb{R}^d$.
- t denotes the number of training epochs with $t \in \mathbb{T} = \{1,\ldots,t_{max}\}$.
- $\boldsymbol{f}$: $\mathbb{X} \times \mathbb{T} \mapsto \mathbb{R}^k$ comprises k objectives, each of which represents a specific performance measure of the ML model after training with setting $\boldsymbol{x}$ for t epochs.

Enhanced Multi-Objective Hyperparameter Optimization Problem

Consider the sequential minimization of an EMOHPO in the following form:

$$\min_{(\boldsymbol{x},t)\in\mathbb{X}\times\mathbb{T}} \boldsymbol{f}(\boldsymbol{x},t) = [f_1(\boldsymbol{x},t),\dots,f_k(\boldsymbol{x},t)], \qquad (2)$$

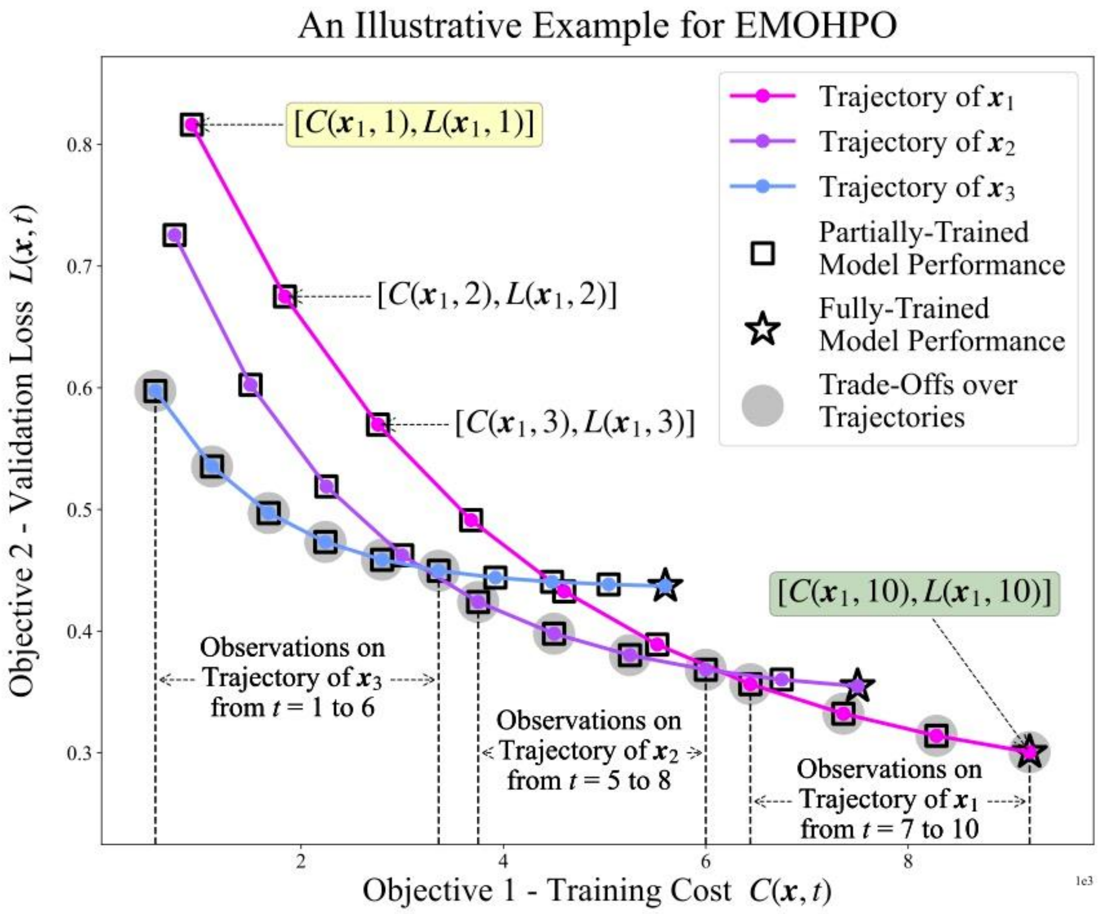
- $\boldsymbol{x}$ denotes a d-dimensional hyperparameter setting with $\boldsymbol{x} \in \mathbb{X} \subset \mathbb{R}^d$.
- t denotes the number of training epochs with $t \in \mathbb{T} = \{1,\dots,t_{max}\}$.
- $\boldsymbol{f}\colon\ \mathbb{X} \times \mathbb{T} \mapsto \mathbb{R}^k$ comprises k objectives, each of which represents a specific performance measure of the ML model after training with setting $\boldsymbol{x}$ for t epochs.
- Assume that when querying at any feasible pair $\boldsymbol{z} = (\boldsymbol{x},t) \in \mathbb{X} \times \mathbb{T}$,
  - Noisy Observation: Each observed model performance is noisy, i.e., for any $i = 1,\dots,k$,

$$y_i(\boldsymbol{x},t) = f_i(\boldsymbol{x},t) + \varepsilon_i \quad \text{and} \quad \varepsilon_i \sim \mathcal{N}(0,\sigma_i^2).$$

  - Iterative Learning: A sequence of multi-objective model performances are observed, i.e., $\{\boldsymbol{y}(\boldsymbol{x},1),\dots,\boldsymbol{y}(\boldsymbol{x},t)\}$ with

$$\boldsymbol{y}(\boldsymbol{x},\tau) = [y_1(\boldsymbol{x},\tau),\dots,y_k(\boldsymbol{x},\tau)], \qquad \forall\tau = 1,\dots,t.$$

Challenges in Solving EMOHPO



In the objective space of EMOHPO,

1. How to make prediction on trajectory?

   The model should be able to capture the characteristics of the trajectory as the epoch changes.

Gaussian Process for Trajectory Prediction

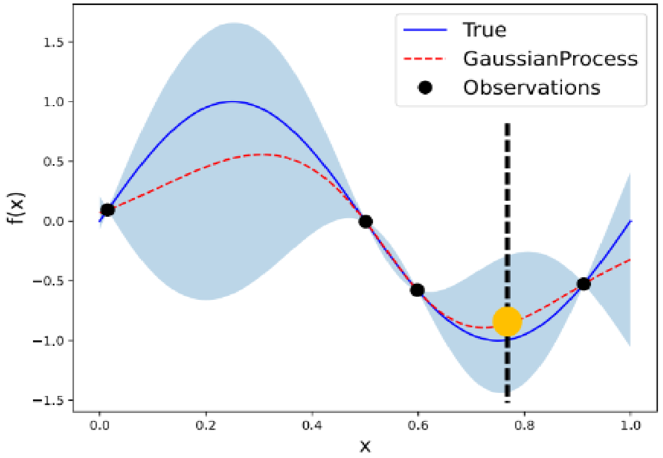
- The Prior:
  - Consider a function $f(\boldsymbol{z})$ to be sampled from a Gaussian Process (GP) with kernel $K(\boldsymbol{z}, \boldsymbol{z}')$ and let $K(Z, Z) \in \mathbb{R}^{n \times n}$ with $[K(Z, Z)]_{i,j} = K(\boldsymbol{z}_i, \boldsymbol{z}_j)$.

$$f(Z) \sim \mathcal{N}\big(0, K(Z, Z)\big), \tag{3}$$

- The Posterior:
  - Conditioning on the observations $Y = \{y_i\}_{i=1}^{n}$ at Z, the posterior predictive distribution at any input $\boldsymbol{z} \in Z$ is given by,

$$f(\boldsymbol{z}) \mid Z, Y \sim \mathcal{N}\big(\mu(\boldsymbol{z}), \Sigma(\boldsymbol{z})\big), \tag{4}$$

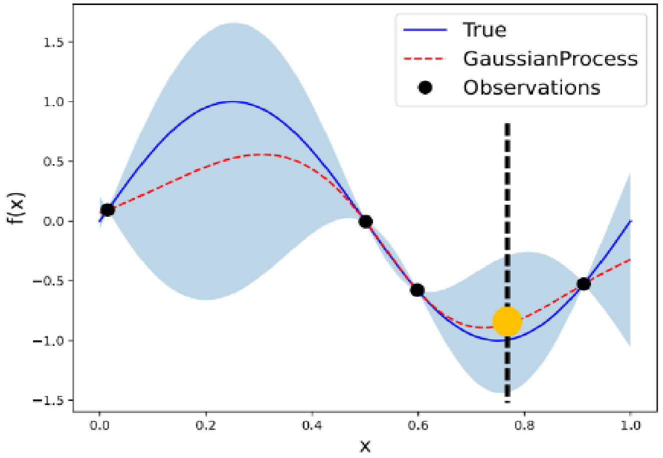
with $\mu(\boldsymbol{z}) = K(\boldsymbol{z}, Z)[K(Z, Z) + \sigma^2 I]^{-1} Y$ and $\Sigma(\boldsymbol{z}) = K(\boldsymbol{z}, \boldsymbol{z}) - K(\boldsymbol{z}, Z)[K(Z, Z) + \sigma^2 I]^{-1} K(Z, \boldsymbol{z})$.



Gaussian Process for Trajectory Prediction

- **The Prior:**
  - Consider a function $f(\boldsymbol{z})$ to be sampled from a Gaussian Process (GP) with kernel $K(\boldsymbol{z}, \boldsymbol{z}')$ and let $K(Z, Z) \in \mathbb{R}^{n \times n}$ with $[K(Z,Z)]_{i,j} = K(\boldsymbol{z}_i, \boldsymbol{z}_j)$.

$$f(Z) \sim \mathcal{N}\big(0, K(Z,Z)\big), \tag{3}$$



- **The Posterior:**
  - Conditioning on the observations $Y = \{y_i\}_{i=1}^{n}$ at Z, the posterior predictive distribution at any input $\boldsymbol{z} \in Z$ is given by,

$$f(\boldsymbol{z}) \mid Z, Y \sim \mathcal{N}\big(\mu(\boldsymbol{z}), \Sigma(\boldsymbol{z})\big), \tag{4}$$

with $\mu(\boldsymbol{z}) = K(\boldsymbol{z}, Z)[K(Z,Z) + \sigma^2 I]^{-1} Y$ and $\Sigma(\boldsymbol{z}) = K(\boldsymbol{z}, \boldsymbol{z}) - K(\boldsymbol{z}, Z)[K(Z,Z) + \sigma^2 I]^{-1} K(Z, \boldsymbol{z})$.

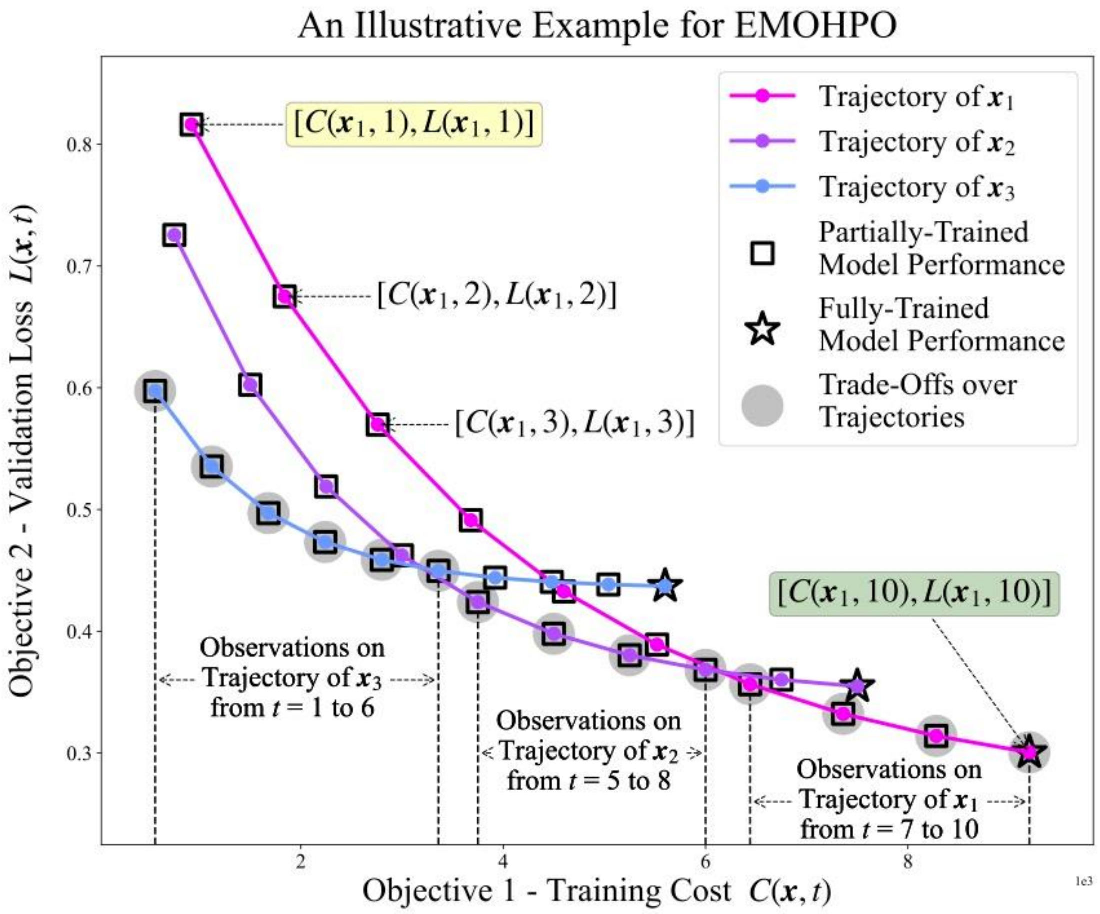
- **Product Kernel:**
  - As a pair $\boldsymbol{z} = (\boldsymbol{x}, t) \in \mathbb{X} \times \mathbb{T}$, a kernel can be decomposed into two parts to capture the iterative learning characteristics

$$K(\boldsymbol{z}, \boldsymbol{z}') = \boxed{K_1(\boldsymbol{x}, \boldsymbol{x}')} \times \boxed{K_2(t, t')}$$

Kernel over hyperparameter setting
e.g., Matérn kernel

Kernel over epoch
e.g., Exponential decay or linear kernel

Challenges in Solving EMOHPO



In the objective space of EMOHPO,

1. How to make prediction on trajectory?

   The model should be able to capture the characteristics of the trajectory as the epoch changes.

2. How to sequentially determine next hyperparameter setting with trajectory prediction? (i.e., new x')

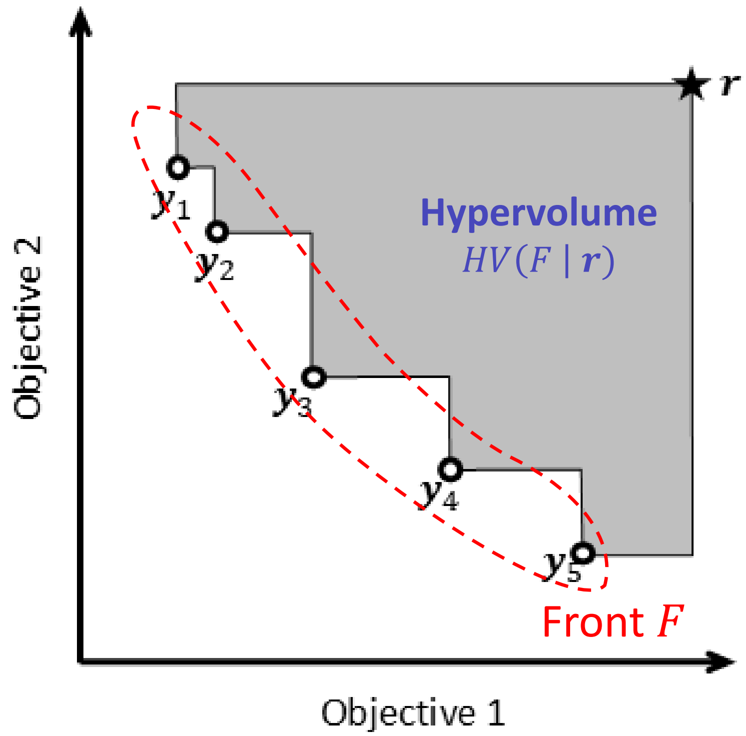
   Assessing the quality of a setting requires consideration of its entire trajectory instead of a single position.

Choose Setting $\boldsymbol{x}'$ - Trajectory-Based Acquisition Function

Definition 1: Hypervolume Improvement (HVI)

Given a front F and a fixed point $\boldsymbol{r}$, the HVI of an objective vector $\boldsymbol{y}'$ is the change in Hypervolume before and after including $\boldsymbol{y}'$ into the front F, i.e.,

$$HVI(\boldsymbol{y}' \mid F, \boldsymbol{r}) = HV(F \cup \{\boldsymbol{y}'\} \mid \boldsymbol{r}) - HV(F \mid \boldsymbol{r}).$$

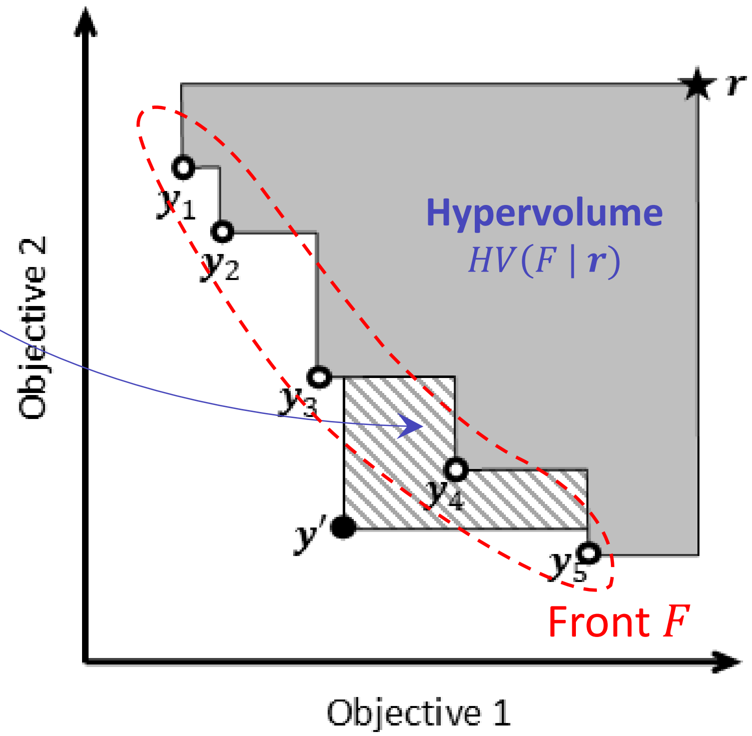


Choose Setting $\boldsymbol{x}'$ - Trajectory-Based Acquisition Function

Definition 1: Hypervolume Improvement (HVI)

Given a front F and a fixed point $\boldsymbol{r}$, the HVI of an objective vector $\boldsymbol{y}'$ is the change in Hypervolume before and after including $\boldsymbol{y}'$ into the front F, i.e.,

$$HVI(\boldsymbol{y}' \mid F, \boldsymbol{r}) = HV(F \cup \{\boldsymbol{y}'\} \mid \boldsymbol{r}) - HV(F \mid \boldsymbol{r}).$$



Choose Setting $\boldsymbol{x}'$ - Trajectory-Based Acquisition Function

Definition 1: Hypervolume Improvement (HVI)

Given a front F and a fixed point $\boldsymbol{r}$, the HVI of an objective vector $\boldsymbol{y}'$ is the change in Hypervolume before and after including $\boldsymbol{y}'$ into the front F, i.e.,

$$HVI(\boldsymbol{y}' \mid F, \boldsymbol{r}) = HV(F \cup \{\boldsymbol{y}'\} \mid \boldsymbol{r}) - HV(F \mid \boldsymbol{r}).$$

Definition 2: Trajectory

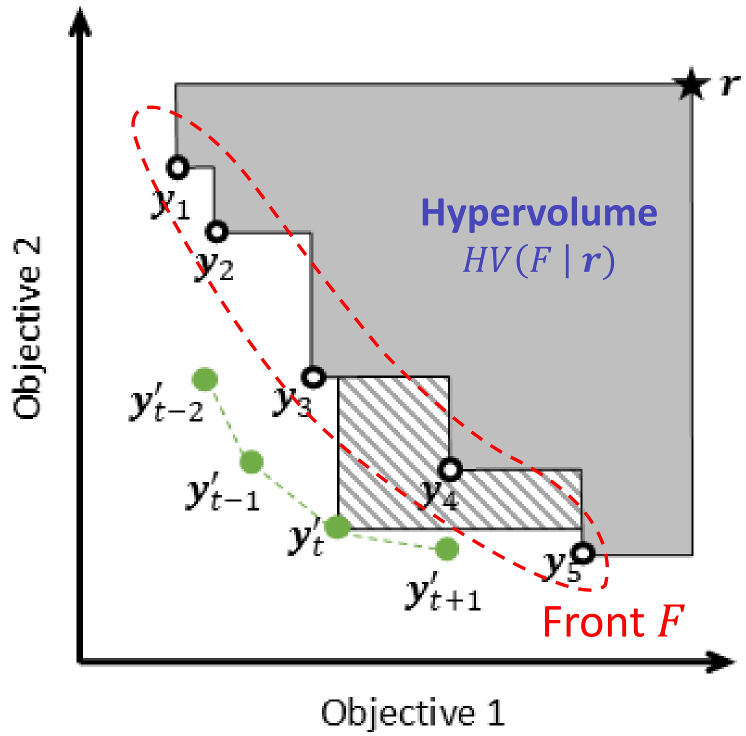
The trajectory of a hyperparameter setting $\boldsymbol{x}$ is defined as the collection of all model performances observed during the entire training with $\boldsymbol{x}$, i.e.,

$$Trj(\boldsymbol{x}) \coloneqq \{\boldsymbol{f}(\boldsymbol{x}, t)\}_{t=1}^{t_{max}} = \{f_1(\boldsymbol{x}, t), \dots, f_k(\boldsymbol{x}, t)\}_{t=1}^{t_{max}}.$$

Definition 3: Trajectory-Based Expected HVI (TEHVI)

TEHVI estimates the gain of an out-of-sample setting $\boldsymbol{x}$ by taking the expectation of HVI over the predictive distribution of its trajectory,

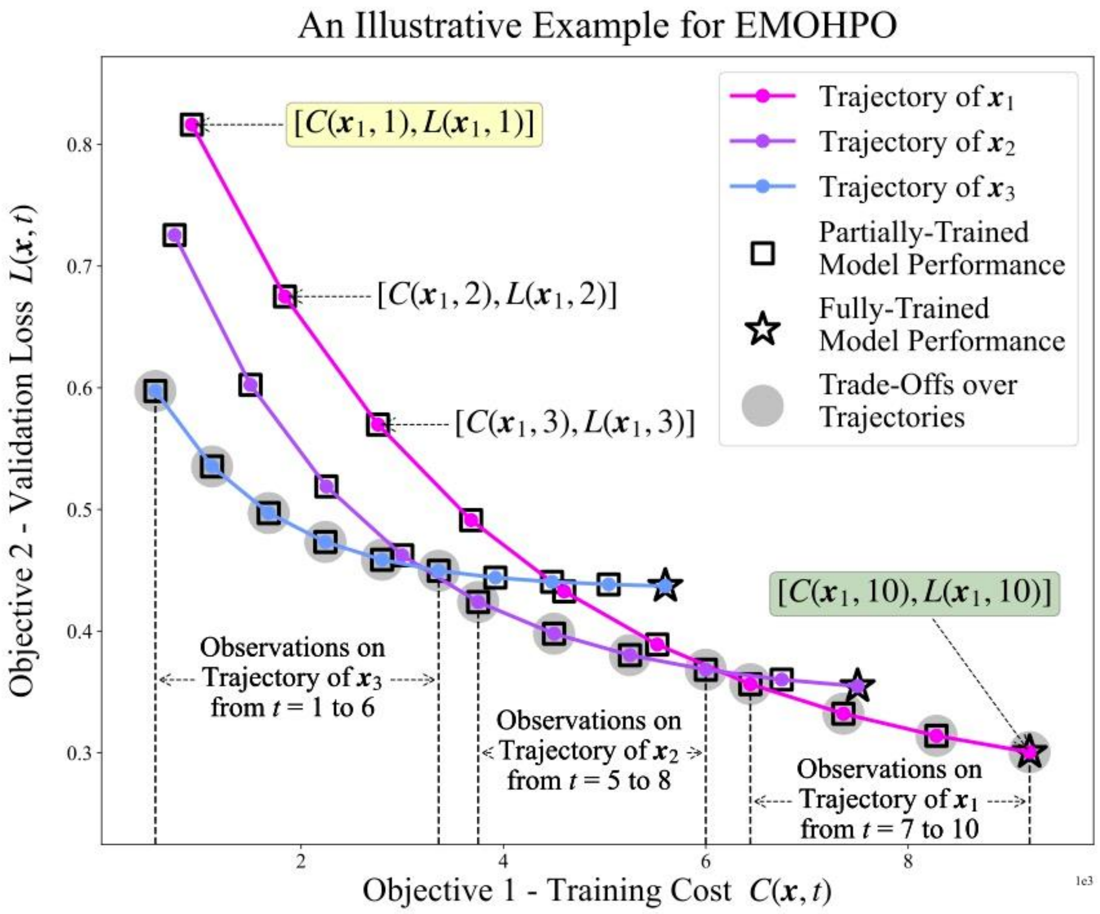
$$TEHVI(\boldsymbol{x} \mid F, \boldsymbol{r}) \coloneqq \mathbb{E}[HVI(Trj(\boldsymbol{x}) \mid F, \boldsymbol{r})] = \mathbb{E}\left[HVI\left(\{\boldsymbol{f}(\boldsymbol{x}, t)\}_{t=1}^{t_{max}} \mid F, \boldsymbol{r}\right)\right].$$



$$\approx \operatorname*{argmax}_{\boldsymbol{x} \in \mathbb{X}} \frac{1}{M} \sum_{m=1}^{M} HVI\left(\widehat{Trj}_m(\boldsymbol{x}) \mid F, \boldsymbol{r}\right)$$

by Monte Carlo integration

Challenges in Solving EMOHPO



In the objective space of EMOHPO,

1. How to make prediction on trajectory?

   The model should be able to capture the characteristics of the trajectory as the epoch changes.

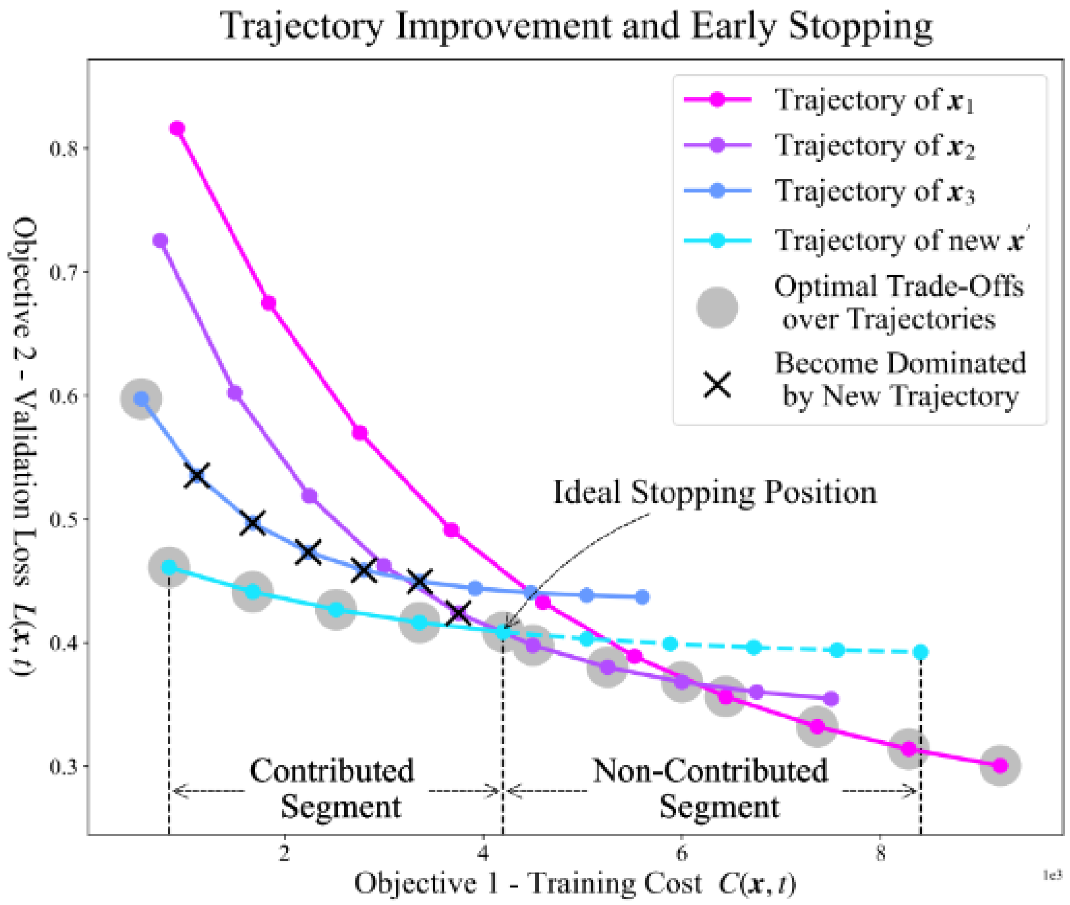
2. How to sequentially determine next hyperparameter setting with trajectory prediction? (i.e., new x')

   Assessing the quality of a setting requires consideration of its entire trajectory instead of a single position.

3. How to execute early stopping without compromising optimization results? (i.e., new t')

   A training procedure should only be stopped after as many trade-offs as possible have been observed along trajectory.

Choose Epoch t' - Trajectory-Based Early Stopping

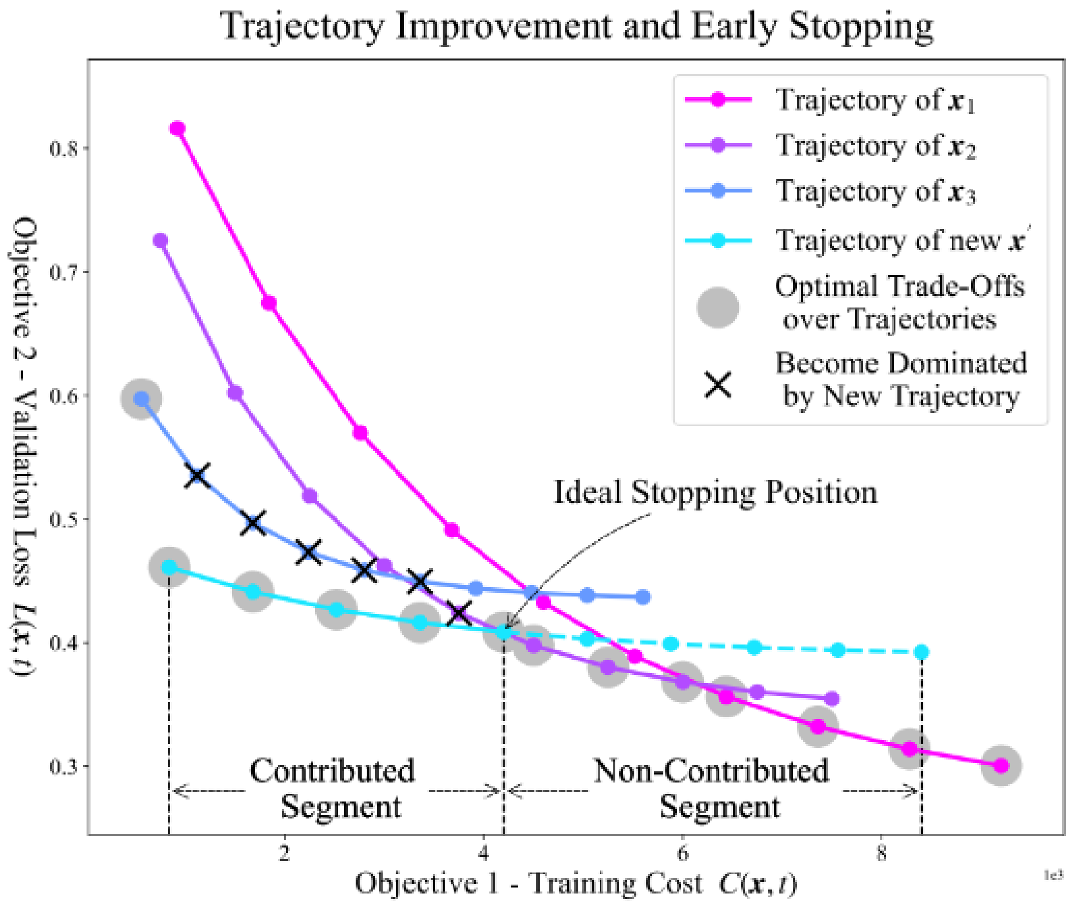


- Conservative stopping epoch

$$t^* = \sup\{t \in \mathbb{T} \mid \boldsymbol{\mu}(\boldsymbol{x}', t) - \beta\boldsymbol{\Sigma}(\boldsymbol{x}', t) \prec \boldsymbol{y}, \exists \boldsymbol{y} \in F\}$$

  - $\boldsymbol{\mu}(\boldsymbol{x}', t) - \beta\boldsymbol{\Sigma}(\boldsymbol{x}', t)$ denotes the lower bound of the performance at t, with β controls the confidence level.
  - Intuitively, t^* is the number of epochs after which future training results are unlikely to improve front F.

Choose Epoch t' - Trajectory-Based Early Stopping



- **Conservative stopping epoch**

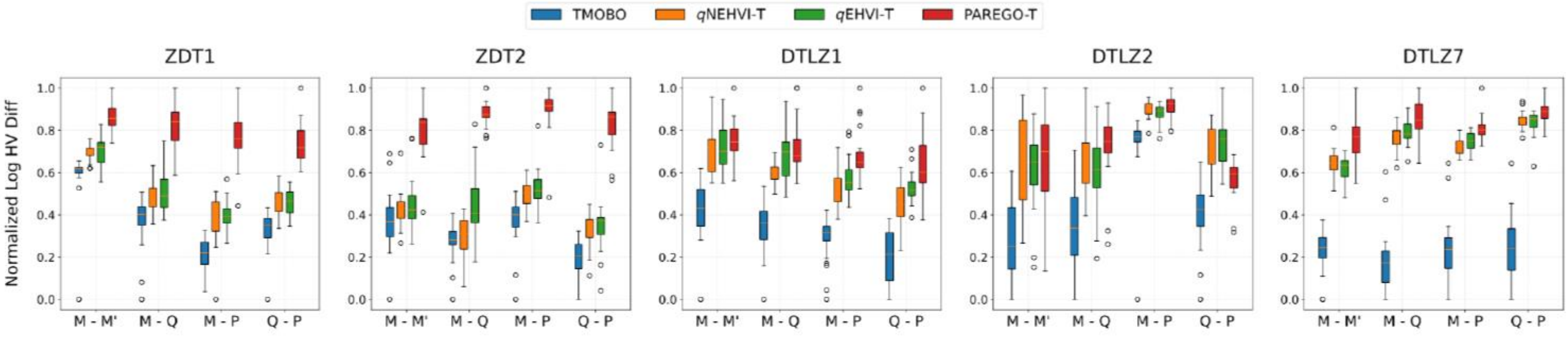
$$t^* = \sup\{t \in \mathbb{T} \mid \boldsymbol{\mu}(\boldsymbol{x}', t) - \beta\boldsymbol{\Sigma}(\boldsymbol{x}', t) \prec \boldsymbol{y}, \exists \boldsymbol{y} \in F\}$$

  - $\boldsymbol{\mu}(\boldsymbol{x}', t) - \beta\boldsymbol{\Sigma}(\boldsymbol{x}', t)$ denotes the lower bound of the performance at t, with β controls the confidence level.
  - Intuitively, t^* is the number of epochs after which future training results are unlikely to improve front F.

- **Early stopping strategy** (t' increases from 1 to t_{max}):
  - if $t' \leq t^*$, continue training with x' for one epoch and let $t' = t' + 1$ and recompute t^*;
  - if $t' > t^*$, terminate the training with x' immediately.

Outline

Results on Synthetic Simulations



- TMOBO consistently achieves the lowest HV difference over 5×4 synthetic problems, which are modeled by

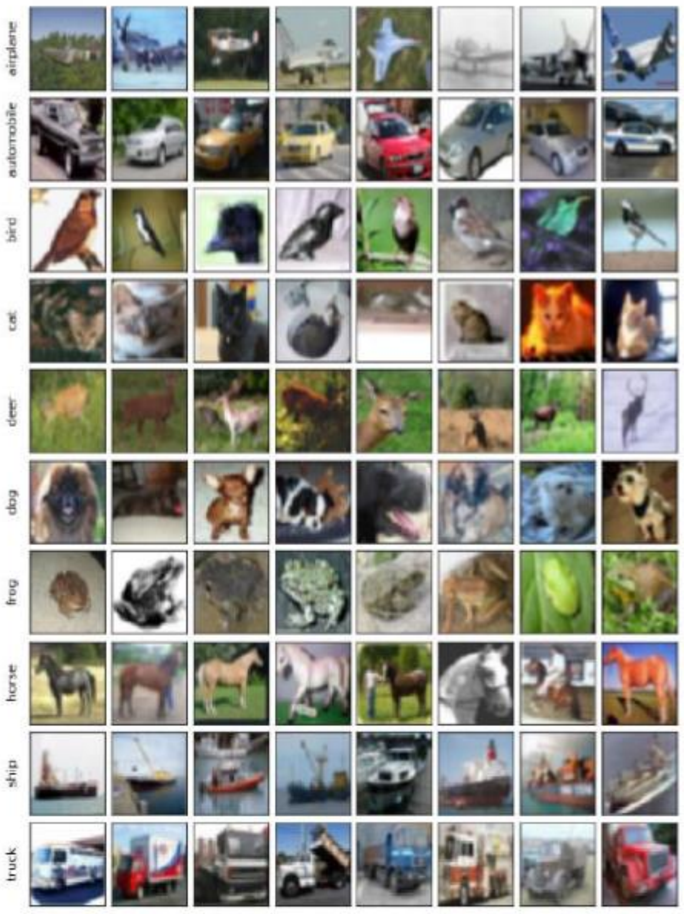
$$\min_{(\boldsymbol{x},t)\,\in\,\mathbb{X}\times\mathbb{T}} \boldsymbol{f}(\boldsymbol{x},t) = [f_1(\boldsymbol{x})\cdot g_1(t),\ldots,f_k(\boldsymbol{x})\cdot g_k(t)]\,.$$

MO benchmark's functions

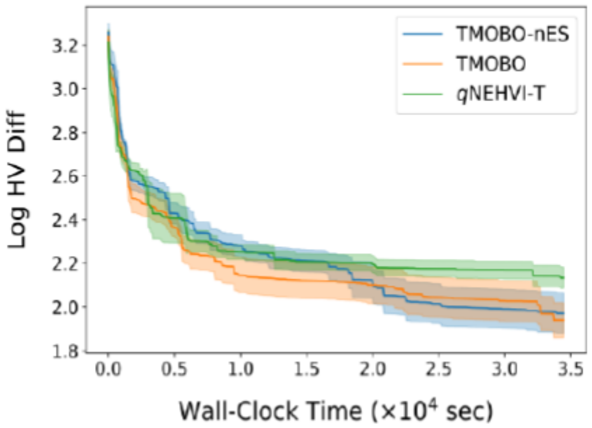
Functions to simulate iterative learning

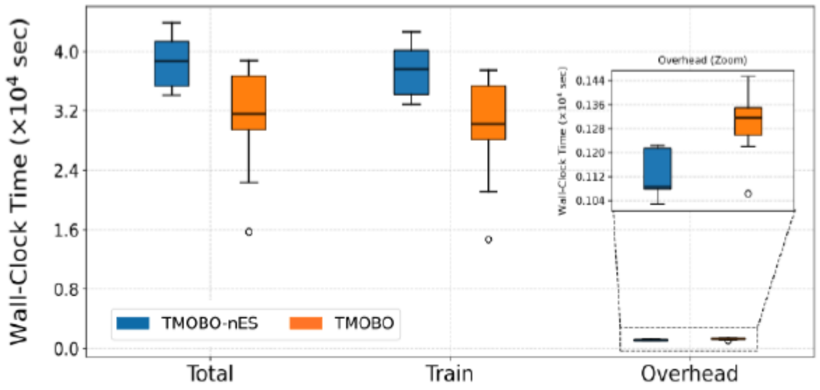
- Over 20 independent trials, the solutions obtained by TMOBO generally dominate a large proportion of those obtained by three alternative enhanced multi-objective optimizers.
  - E.g., qNEHVI-T denotes the enhanced qNEHVI[3] by collecting all trajectory observations, similarly for qEHVI-T and ParEGO-T.

Results on Real-World Benchmarks



- Tuning a more complex MobileNetV2 model[11] with iterative learning on CIFAR-10 image datasets:





  - [Left] TMOBO and its variant TMOBO-nES outperform qNEHVI-T, with TMOBO demonstrating faster early convergence.
  - [Right] TMOBO reduces model training time more than TMOBO-nES, though it incurs slightly higher (but negligible) computation overhead.

Outline

Conclusions

- Considering MOHPO with iterative learning, our interest centers on (1) how trajectory information affects the distribution of trade-offs and (2) how to leverage this information to search trade-offs.

Problem Definition:	Introduce EMOHPO problem by including the number of training epoch as an explicit decision variable to reveal the trade-offs that may occur along trajectories.
Methodology:	Propose TMOBO method that iteratively samples setting based on trajectory-based contribution and decides when to stop training based on trajectory predictions.
Numerical Study:	Show the advantage of TMOBO over alternative methods in locating trade-offs for EMOHPO through synthetic and real-world benchmarks.

- For future research of this study
  - Development of the analytical form or more efficient approximation for the computation of TEHVI.
  - Application of scalable GP and more advanced data augmentation strategy in large-scale applications.
  - Extend the EMOHPO framework to other scenarios, such as drug design and material engineering where an iterative procedure typically exists.

References

1. Knowles, J. (2006). ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE transactions on evolutionary computation*, *10*(1), 50-66.
2. Daulton, S., Balandat, M., & Bakshy, E. (2020). Differentiable expected hypervolume improvement for parallel multi-objective Bayesian optimization. *Advances in Neural Information Processing Systems*, *33*, 9851-9864.
3. Daulton, S., Balandat, M., & Bakshy, E. (2021). Parallel bayesian optimization of multiple noisy objectives with expected hypervolume improvement. *Advances in Neural Information Processing Systems*, *34*, 2187-2200.
4. Swersky, K., Snoek, J., & Adams, R. P. (2014). Freeze-thaw Bayesian optimization. *arXiv preprint arXiv:1406.3896*.
5. Falkner, S., Klein, A., & Hutter, F. (2018, July). BOHB: Robust and efficient hyperparameter optimization at scale. In *International conference on machine learning* (pp. 1437-1446). PMLR.
6. Nguyen, V., Schulze, S., & Osborne, M. (2020). Bayesian optimization for iterative learning. *Advances in Neural Information Processing Systems*, *33*, 9361-9371.
7. Dai, Z., Yu, H., Low, B. K. H., & Jaillet, P. (2019, May). Bayesian optimization meets Bayesian optimal stopping. In *International conference on machine learning* (pp. 1496-1506). PMLR.
8. Belakaria, S., Deshwal, A., & Doppa, J. R. (2020, April). Multi-fidelity multi-objective Bayesian optimization: An output space entropy search approach. In *Proceedings of the AAAI Conference on artificial intelligence* (Vol. 34, No. 06, pp. 10035-10043).
9. Schmucker, R., Donini, M., Zafar, M. B., Salinas, D., & Archambeau, C. (2021). Multi-objective asynchronous successive halving. *arXiv preprint arXiv:2106.12639*.
10. Zimmer, L., Lindauer, M., & Hutter, F. (2021). Auto-pytorch: Multi-fidelity metalearning for efficient and robust autodl. *IEEE transactions on pattern analysis and machine intelligence*, *43*(9), 3079-3090.
11. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510-4520).

Thanks for Your Attention!

Q & A

Wenyu Wang[1], Zheyi Fan[2,3], Szu Hui Ng[1,*]

[1]Industrial Systems Engineering and Management, National University of Singapore, Singapore
[2]Academy of Mathematics and System Science, Chinese Academy of Sciences, China
[3]School of Mathematical Sciences, University of Chinese Academy of Sciences, China