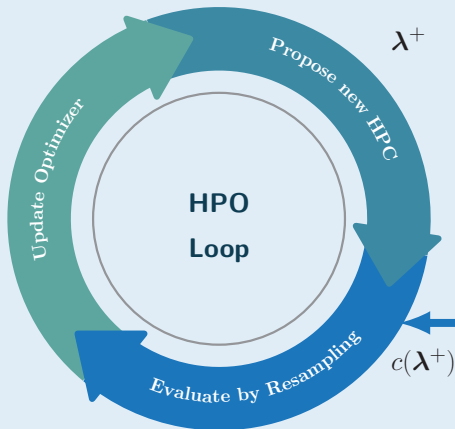


Input

Training Data $\mathcal{D}_{\text{train}}$, Inducer \mathcal{I} ,
metric ρ , splits \mathcal{J} , search space $\tilde{\Lambda}$

Self-Tuning learner $\mathcal{T}_{\mathcal{I}, \tilde{\Lambda}, \rho, \mathcal{J}}$

Tuning/2nd Level Inference



1st Level Inference

for configuration λ^+
 $\mathcal{J} = ((J_{\text{train},1}, J_{\text{test},1}), \dots, (J_{\text{train},B}, J_{\text{test},B}))$
for $k = 1, \dots, B$
 $\hat{f}_{\hat{\theta},k} = \mathcal{I}_{\lambda^+}(\mathcal{D}_{\text{train},k})$
 $= \arg \min_{f \in \mathcal{H}} \sum_{(\mathbf{x}, y) \in \mathcal{D}_{\text{train},k}} L(y, f_{\theta}(\mathbf{x})),$
where λ^+ configures the above e.g. by
modifying \mathcal{H} , L or aspects of the optimizer.

$$\hat{\lambda} = \tau(\mathcal{D}, \mathcal{I}, \tilde{\Lambda}, \rho)$$

\cap

$$\arg \min_{\lambda \in \tilde{\Lambda}} c(\lambda)$$

Final Model Fit

Fit \mathcal{I} with $\hat{\lambda}$ on $\mathcal{D}_{\text{train}}$

Return

Model \hat{f} , HPC $\hat{\lambda}$