Appendix to Hyperparameter Transfer Across Developer Adjustments

Danny Stoll[1], Jörg K.H. Franke[1], Diane Wagner[1], Simon Selg[1] & Frank Hutter[1,2]
[1]University of Freiburg
[2]Bosch Center for Artificial Intelligence
`{stolld,frankej,wagnerd,selgs,fh}@cs.uni-freiburg.de`

A Pseudocode

Algorithm 1 Sampling strategy in transfer TPE

Input: Current hyperparameter space $\mathcal{X}_{\text{new}}$, previous hyperparameter space $\mathcal{X}_{\text{old}}$, config ranking of previous optimization $\mathcal{C}$, budget b_{new}

Decompose $\mathcal{X}_{\text{new}} = (\mathcal{X}_{\text{both}} \cup \mathcal{X}_{\text{both,range-only-new}}) \times \mathcal{X}_{\text{only-new}}$
Discard configs in $\mathcal{C}$ that have hyperparameter values in $\mathcal{X}_{\text{both,range-only-new}}$
Project configs in $\mathcal{C}$ to space $\mathcal{X}_{\text{both}}$, to yield config ranking $\mathcal{C}_{\text{both}}$
Fit TPE model M_{both} for $\mathcal{X}_{\text{both}}$ on $\mathcal{C}_{\text{both}}$
for t **in** $1, \ldots, b_{\text{new}}$ **do**
  if is random fraction **then** ▷ From TPE implementation, e.g., 1/3 of cases
    Sample $\mathbf{x}_{\text{new}}$ from prior on $\mathcal{X}_{\text{new}}$
  else if no model for $\mathcal{X}_{\text{new}}$ **then**
    Sample $\mathbf{x}_{\text{both}}$ from $\mathcal{X}_{\text{both}}$ according to M_{both}
    for hyperparameter range $\mathcal{X}^{H_i}_{\text{both,range-only-new}} \neq \emptyset$ **in** $\mathcal{X}_{\text{both,range-only-new}}$ **do**
      Set $p := \frac{|\mathcal{X}^{H_i}_{\text{both,range-only-new}}|}{|\mathcal{X}^{H_i}_{\text{new}}|}$
      Sample x^i from prior on $\mathcal{X}^{H_i}_{\text{both,range-only-new}}$
      Set $\mathbf{x}^i_{\text{both}} := \mathrm{x}^i$ with probability p
    Sample $\mathbf{x}_{\text{only-new}}$ from prior on $\mathcal{X}_{\text{only-new}}$
    Combine $\mathbf{x}_{\text{both}}$ with $\mathbf{x}_{\text{only-new}}$ to yield sample $\mathbf{x}_{\text{new}}$
  else
    Fit TPE model M_{new} for $\mathcal{X}_{\text{new}}$ on current observations
    Sample $\mathbf{x}_{\text{new}}$ from $\mathcal{X}_{\text{new}}$ according to M_{new}
return

4th Workshop on Meta-Learning at NeurIPS 2020, Vancouver, Canada.

B Transfer TPE Sampling Illustration

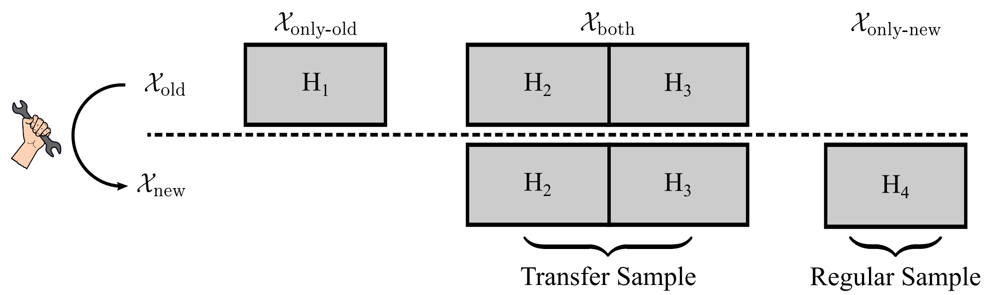

Figure 1: Sampling in T2PE for hyperparameter additions / removals.

C Benchmark Suite Details

C.1 Overview

Table 1: Benchmarks overview

Benchmark	#Hyperparameters Old	#Hyperparameters New	#Tasks
FCN-A	6	5	4
FCN-B	6	8	4
NAS-A	6	6	3
NAS-B	3	6	3
XGB-A	5	9	10
XGB-B	6	6	10
SVM-A	2	2	10
SVM-B	2	2	10

C.2 FCN-A & FCN-B

Budget For FCN-A the budget is set to 100. For FCN-B, additional to the changes in the search space (Table 4), the budget is increased from 50 to 100.

Table 2: Values for integer coded hyperparameters in FCN benchmarks

Hyperparameter	Values
# Units Layer $\{1, 2\}$	$(16, 32, 64, 128, 256, 512)$
Dropout Layer $\{1, 2\}$	$(0.0, 0.3, 0.6)$
Initial Learning Rate	$(0.0005, 0.001, 0.005, 0.01, 0.05, 0.1)$
Batch Size	$(8, 16, 32, 64)$

Table 3: Search spaces in FCN-A. Numerical hyperparameters are encoded as integers, see Table 2 for specific values for these hyperparameters.

Steps	Hyperparameter	Range/Value	Prior
1	# Units Layer 1	1	-
1	# Units Layer 2	1	-
1	Batch Size	$\{0, \ldots, 3\}$	Uniform
1, 2	Dropout Layer 1	$\{0, \ldots, 2\}$	Uniform
1, 2	Dropout Layer 2	$\{0, \ldots, 2\}$	Uniform
1, 2	Activation Layer 1	{ReLu, tanh}	Uniform
1, 2	Activation Layer 2	{ReLu, tanh}	Uniform
1, 2	Initial Learning Rate	$\{0, \ldots, 5\}$	Uniform
1, 2	Learning Rate Schedule	Constant	Uniform
2	# Units Layer 1	5	-
2	# Units Layer 2	5	-
2	Batch Size	1	-

Table 4: Search spaces in FCN-B. Numerical hyperparameters are encoded as integers, see Table 2 for specific values for these hyperparameters.

Steps	Hyperparameter	Range/Value	Prior
1	Activation Layer 1	tanh	-
1	Activation Layer 2	tanh	-
1	Learning Rate Schedule	Constant	-
1, 2	# Units Layer 1	$\{0, \ldots, 5\}$	Uniform
1, 2	# Units Layer 2	$\{0, \ldots, 5\}$	Uniform
1, 2	Dropout Layer 1	$\{0, \ldots, 2\}$	Uniform
1, 2	Dropout Layer 2	$\{0, \ldots, 2\}$	Uniform
1, 2	Initial Learning Rate	$\{0, \ldots, 5\}$	Uniform
1, 2	Batch Size	$\{0, \ldots, 3\}$	Uniform
2	Activation Layer 1	{ReLu, tanh}	Uniform
2	Activation Layer 2	{ReLu, tanh}	Uniform
2	Learning Rate Schedule	Cosine	-

C.3 NAS-A & NAS-B

Table 5: Search spaces in NAS-A.

Steps	Hyperparameter	Range/Value	Prior
1, 2	$0 \rightarrow 2$	{ none, skip-connect, conv1x1, conv3x3, avg-pool3x3 }	Uniform
1, 2	$0 \rightarrow 3$	{ none, skip-connect, conv1x1, conv3x3, avg-pool3x3 }	Uniform
1, 2	$2 \rightarrow 3$	{ none, skip-connect, conv1x1, conv3x3, avg-pool3x3 }	Uniform
2	$0 \rightarrow 1$	{ none, skip-connect, conv1x1, conv3x3, avg-pool3x3 }	Uniform
2	$1 \rightarrow 2$	{ none, skip-connect, conv1x1, conv3x3, avg-pool3x3 }	Uniform
2	$1 \rightarrow 3$	{ none, skip-connect, conv1x1, conv3x3, avg-pool3x3 }	Uniform

Table 6: Search spaces in NAS-B.

Steps	Hyperparameter	Range/Value	Prior
1	$0 \to 1$	{ none, skip-connect, conv1x1, conv3x3 }	Uniform
1	$0 \to 2$	{ none, skip-connect, conv1x1, conv3x3 }	Uniform
1	$0 \to 3$	{ none, skip-connect, conv1x1, conv3x3 }	Uniform
1	$1 \to 2$	{ none, skip-connect, conv1x1, conv3x3 }	Uniform
1	$1 \to 3$	{ none, skip-connect, conv1x1, conv3x3 }	Uniform
1	$2 \to 3$	{ none, skip-connect, conv1x1, conv3x3 }	Uniform
2	$0 \to 1$	{ none, skip-connect, conv1x1, conv3x3, avg-pool3x3 }	Uniform
2	$0 \to 2$	{ none, skip-connect, conv1x1, conv3x3, avg-pool3x3 }	Uniform
2	$0 \to 3$	{ none, skip-connect, conv1x1, conv3x3, avg-pool3x3 }	Uniform
2	$1 \to 2$	{ none, skip-connect, conv1x1, conv3x3, avg-pool3x3 }	Uniform
2	$1 \to 3$	{ none, skip-connect, conv1x1, conv3x3, avg-pool3x3 }	Uniform
2	$2 \to 3$	{ none, skip-connect, conv1x1, conv3x3, avg-pool3x3 }	Uniform

C.4 SVM-A & SVM-B

Table 7: Search spaces in SVM-A.

Steps	Hyperparameter	Range/Value	Prior
1	Kernel	Radial	-
1	Degree	$\{2, \ldots, 5\}$	Uniform
1, 2	Cost	$[2^{-10}, 2^{10}]$	Log-uniform
2	Kernel	Polynomial	-
2	γ	$[2^{-5}, 2^{5}]$	Log-uniform

Table 8: Search spaces in SVM-B.

Steps	Hyperparameter	Range/Value	Prior
1	Cost	$[2^{-5}, 2^{5}]$	Log-uniform
1, 2	γ	1	-
1, 2	Degree	5	-
1, 2	Kernel	{Polynomial, Linear, Radial}	Uniform
2	Cost	$[2^{-10}, 2^{10}]$	Log-uniform

C.5 XGB-A & XGB-B

Table 9: Search spaces in XGB-A

Steps	Hyperparameter	Range/Value	Prior
1	Colsample-by-tree	1	-
1	Colsample-by-level	1	-
1	Minimum child weight	1	-
1	Maximum depth	6	-
1, 2	Booster	Tree	-
1, 2	# Rounds	$\{1, \ldots, 5,000\}$	Uniform
1, 2	Subsample	$[0, 1]$	Uniform
1, 2	Eta	$[2^{-10}, 2^{0}]$	Log-uniform
1, 2	Lambda	$[2^{-10}, 2^{10}]$	Log-uniform
1, 2	Alpha	$[2^{-10}, 2^{10}]$	Log-uniform
2	Colsample-by-tree	$[0, 1]$	Uniform
2	Colsample-by-level	$[0, 1]$	Uniform
2	Minimum child weight	$[2^{0}, 2^{7}]$	Log-uniform
2	Maximum depth	$\{1, \ldots, 15\}$	Uniform

Table 10: Search spaces in XGB-B

Steps	Hyperparameter	Range/Value	Prior
1	Colsample-by-tree	1	-
1	Colsample-by-level	1	-
1	Minimum child weight	1	-
1	Maximum depth	6	-
1, 2	Booster	{ Linear, Tree }	-
1, 2	# Rounds	$\{1, \ldots, 5,000\}$	Uniform
1, 2	Subsample	$[0, 1]$	Uniform
1, 2	Eta	$[2^{-10}, 2^{0}]$	Log-uniform
1, 2	Lambda	$[2^{-10}, 2^{10}]$	Log-uniform
1, 2	Alpha	$[2^{-10}, 2^{10}]$	Log-uniform
2	Colsample-by-tree	1	-
2	Colsample-by-level	0.5	-
2	Minimum child weight	10	-
2	Maximum depth	10	-

D Detailed Speedups

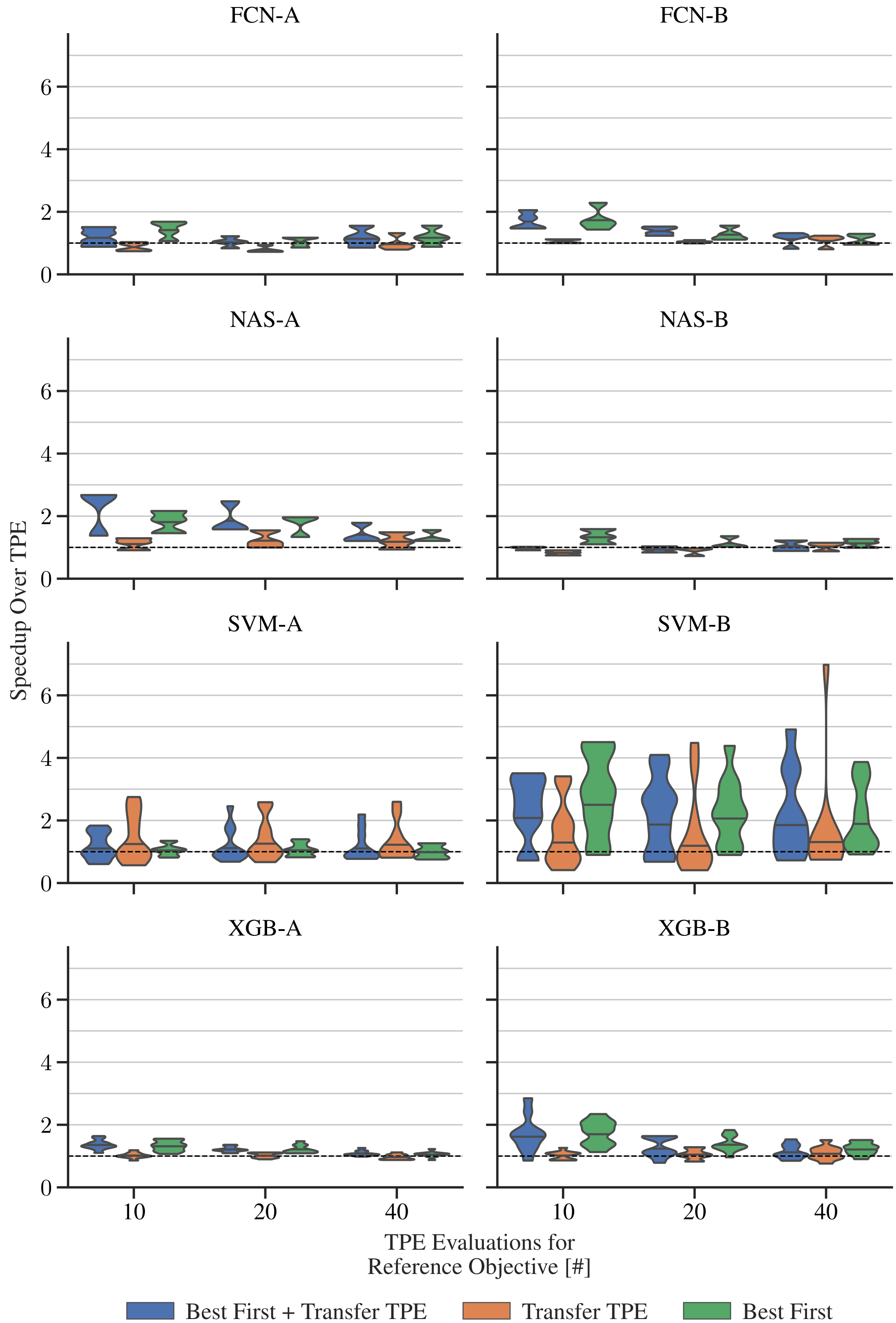

Figure 2: Speedup of transfer TPE, best-first, and their combination, over TPE across tasks for each of 8 benchmarks. The previous HPO has a budget of 10 evaluations. The violins estimate densities of the task means. The horizontal line in each violin shows the mean across these task means. In each plot, the budget for the TPE reference increases.

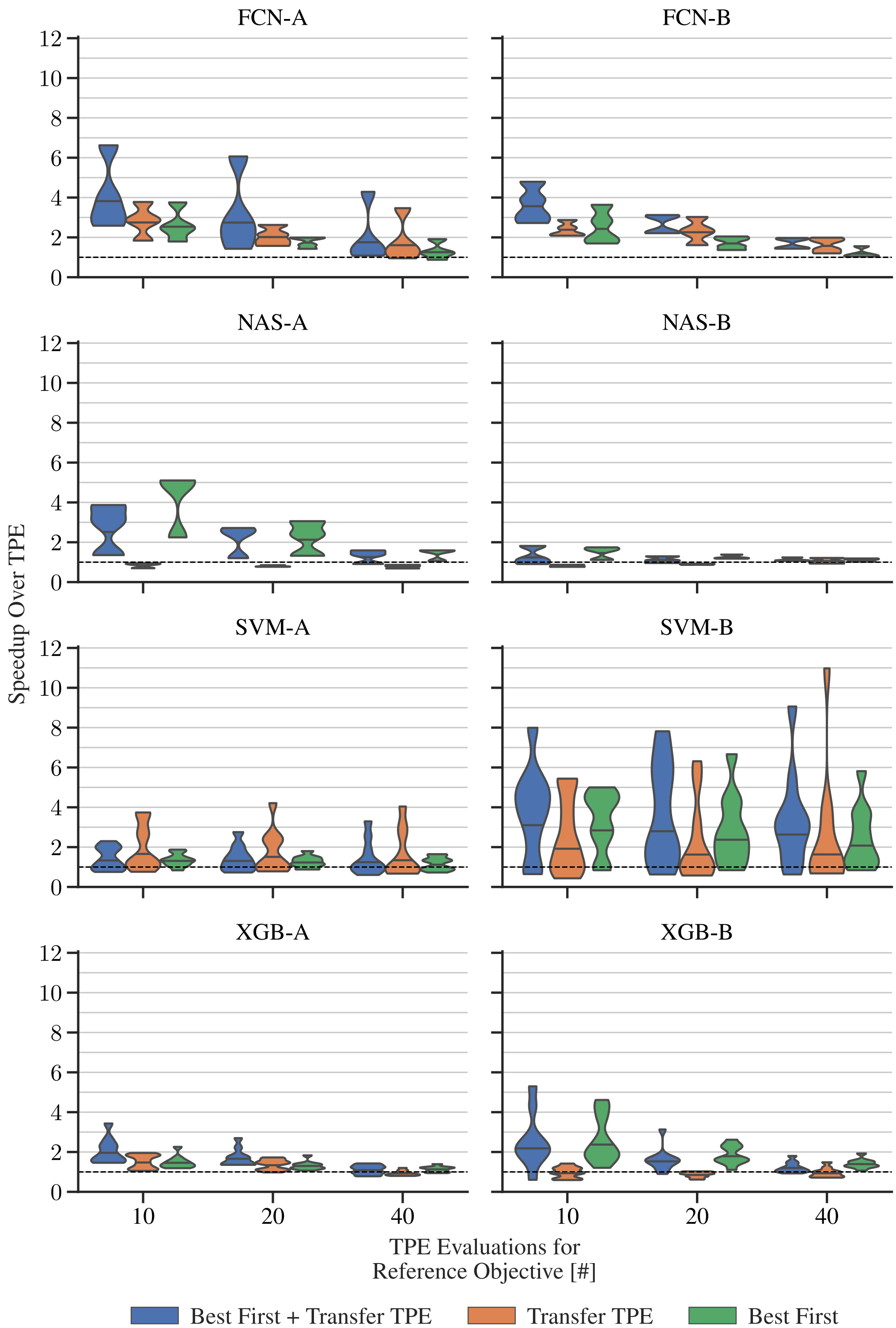

Figure 3: Speedup of transfer TPE, best-first, and their combination, over TPE across tasks for each of 8 benchmarks. The previous HPO has a budget of 20 evaluations. The violins estimate densities of the task means. The horizontal line in each violin shows the mean across these task means. In each plot, the budget for the TPE reference increases.

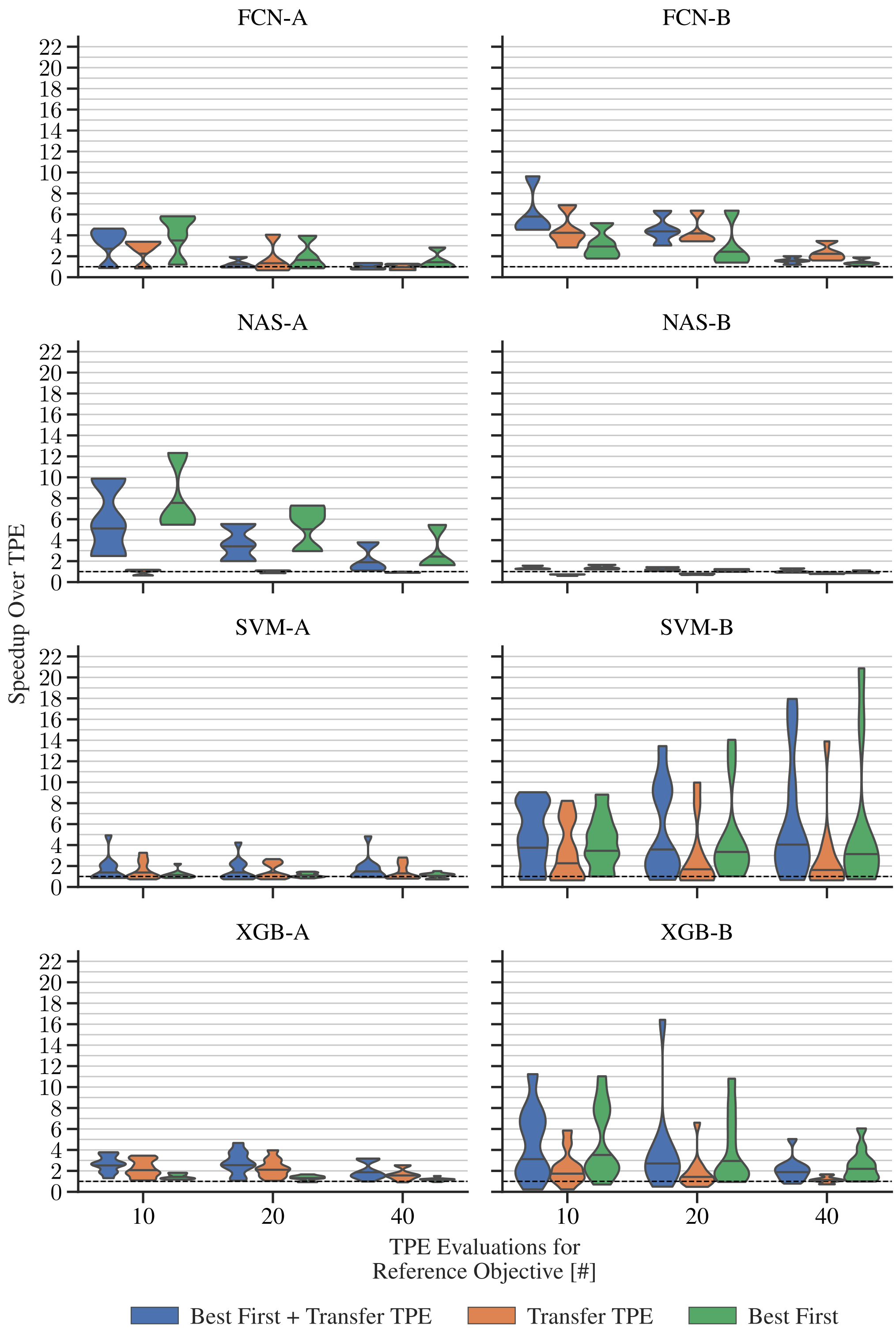

Figure 4: Speedup of transfer TPE, best-first, and their combination, over TPE across tasks for each of 8 benchmarks. The previous HPO has a budget of 40 evaluations. The violins estimate densities of the task means. The horizontal line in each violin shows the mean across these task means. In each plot, the budget for the TPE reference increases.

E Failure Rates

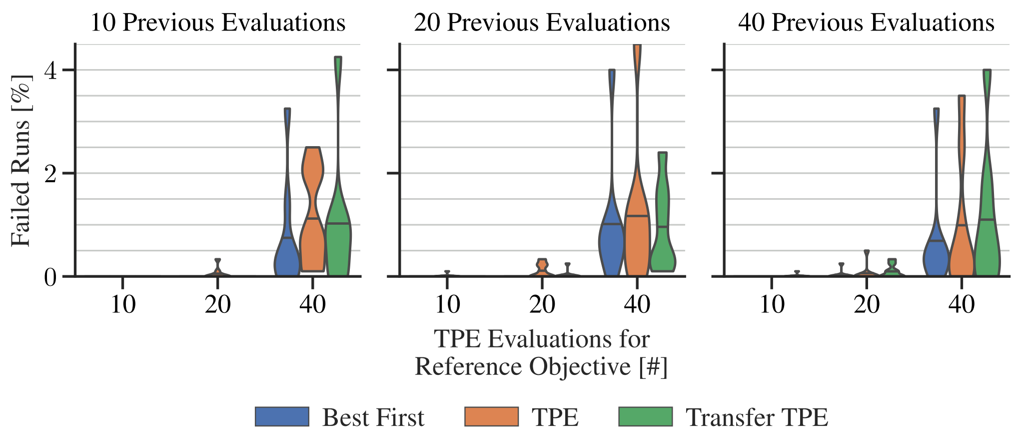

Figure 5: Failure rates for transfer TPE, best-first, and TPE across 8 benchmarks. The violins estimate densities of the task means. The horizontal line in each violin shows the mean across these task means. The plots from left to right utilize increasing budget for the pre-adjustment hyperparameter. In each plot, the budget for the TPE reference increases.

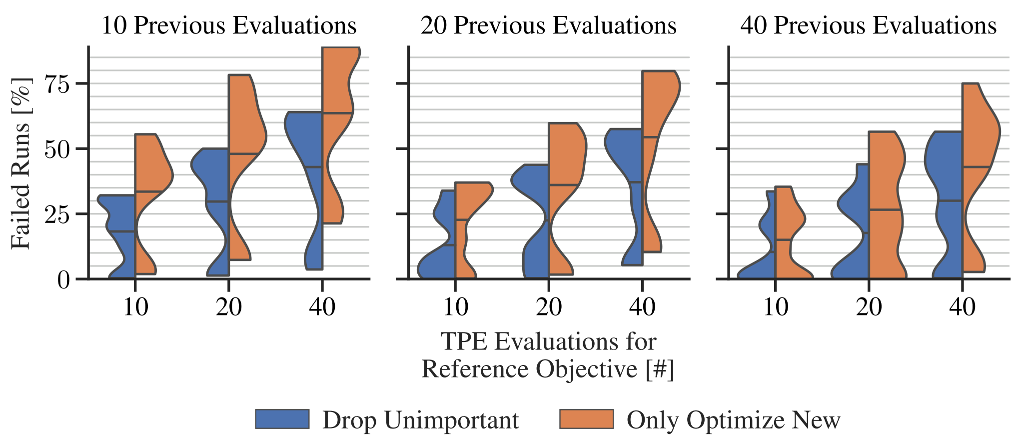

Figure 6: Percent of runs that never reach the reference objective for the drop-unimporant and only-optimize-new approach. Each data point for the violins represents the mean percentage of failures for a benchmark. The line in each violin shows the mean across these benchmark means. Plots from left to right increase in budget for the pre-adjustment hyperparameter optimization. In each plot, the budget of the TPE reference increases.

F Control Study: TPE for Different Random Seed Ranges

As a sanity check, and to gauge the influence of random seeds, we compare TPE to itself with different seed ranges. In general we observe little differences in TPE and TPE2, with the exception of one outlier task (Figure 7).

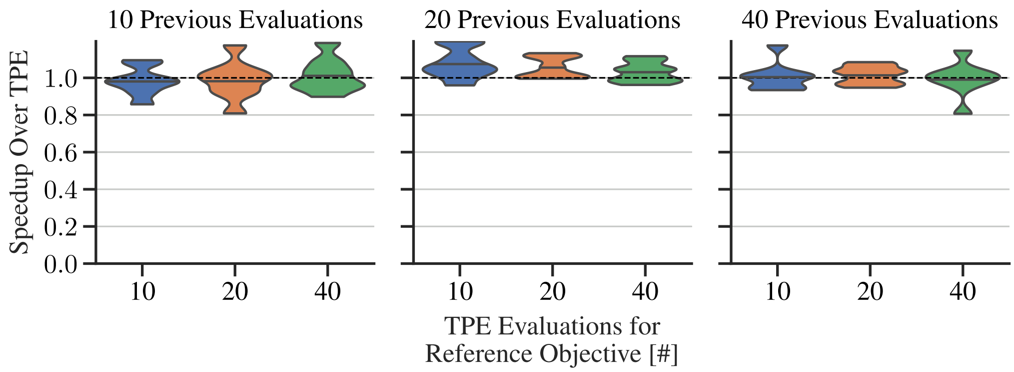

Figure 7: Speedup of TPE over TPE2 across 8 benchmarks. The violins estimate densities of the benchmark means. The horizontal line in each violin shows the mean across these benchmark means. The plots from left to right utilize increasing budget for the pre-adjustment hyperparameter optimization. In each plot, the budget for the TPE reference increases.

G Control Study: Random Search vs TPE

As a sanity check, and for context, we compare TPE to random search (Figure 8).

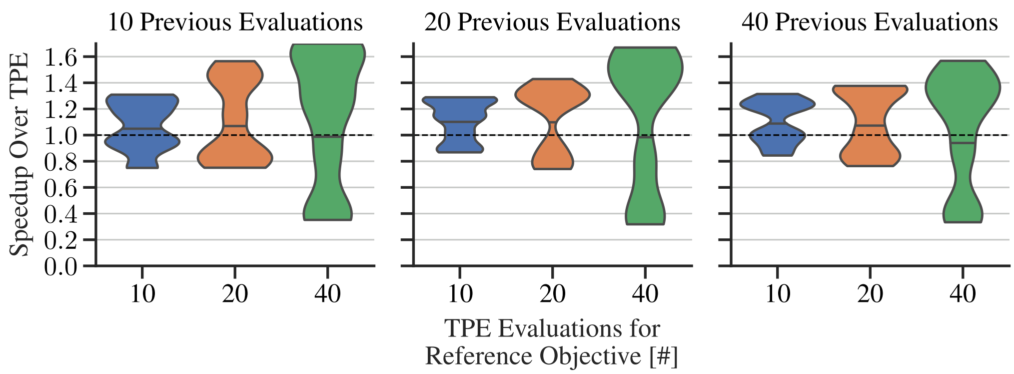

Figure 8: Speedup of random search over TPE across 8 benchmarks. The violins estimate densities of the benchmark means. The horizontal line in each violin shows the mean across these benchmark means. The plots from left to right utilize increasing budget for the pre-adjustment hyperparameter optimization. In each plot, the budget for the TPE reference increases.