Appendix

A Proofs

A.1 Proof of Theorem 1

Proof. Using the definition of $p(\boldsymbol{\theta}|f, \mathcal{D})$, the probability of improvement is computed as:

$$\begin{aligned} p(f \leq f^\gamma \mid \boldsymbol{\theta}, \mathcal{D}) &= \int_{-\infty}^{f^\gamma} p(f|\boldsymbol{\theta}, \mathcal{D}) df \\ &= \int_{-\infty}^{f^\gamma} \frac{p(\boldsymbol{\theta}|f, \mathcal{D}) p(f|\mathcal{D})}{p(\boldsymbol{\theta}|\mathcal{D})} df \\ &= \frac{l(\boldsymbol{\theta}|\mathcal{D}^{(l)})}{p(\boldsymbol{\theta}|\mathcal{D})} \int_{-\infty}^{f^\gamma} p(f|\mathcal{D}) df \end{aligned}$$

Since the expected improvement for TPE is computed as:

$$\mathrm{EI}[\boldsymbol{\theta}|\mathcal{D}] = \frac{l(\boldsymbol{\theta}|\mathcal{D}^{(l)})}{p(\boldsymbol{\theta}|\mathcal{D})} \int_{-\infty}^{f^\gamma} (f^\gamma - f) p(f|\mathcal{D}) dy$$

and both equations have the common part $l(\boldsymbol{\theta}|\mathcal{D}^{(l)})/p(\boldsymbol{\theta}|\mathcal{D})$, this part is canceled out each other when we take the ratio. For this reason, the derivative of the ratio of two equations with respect to $\boldsymbol{\theta}$ is computed as:

$$\frac{\partial}{\partial \boldsymbol{\theta}} \underbrace{\frac{\int_{-\infty}^{f^\gamma} (f^\gamma - f) p(f|\mathcal{D}) df}{\int_{-\infty}^{f^\gamma} p(f|\mathcal{D}) df}}_{\text{const w.r.t. } \boldsymbol{\theta}} = \mathbf{0} \tag{1}$$

where, since we assume that the support of $p(f \leq f^\gamma|\boldsymbol{\theta}, \mathcal{D})$ covers the whole domain $\boldsymbol{\Theta}$, i.e. $p(f \leq f^\gamma|\boldsymbol{\theta}, \mathcal{D}) \neq 0$, the factor, which is differentiated in Eq. (1), takes a finite constant value. This leads to the partial derivative of zero. □

A.2 Proof of Theorem 2

Proof. Let the priority factor of the i-th constraint over the j-th constraint be $P \in \mathcal{P}_{i,j}(\boldsymbol{\theta})$ such that the following holds:

$$\begin{aligned} \mathcal{P}_{i,j}(\boldsymbol{\theta}) = \Bigg\{ P \Bigg| & \left(\gamma_{c_i^\star} + (1-\gamma_{c_i^\star})(r_i(\boldsymbol{\theta}))^{-1}\right)^{-1} \\ & - \left(\gamma_{c_j^\star} + (1-\gamma_{c_j^\star})(r_i(\boldsymbol{\theta})P)^{-1}\right)^{-1} \geq 0, \\ & P \in \mathbb{R}_+ \cup \{\infty\} \Bigg\}. \end{aligned} \tag{2}$$

Then Theorem 2 will be rephrased into the following:

Proposition 1. *Under the given conditions in Theorem 2, the supremum of the priority factor is:*

$$P_{i,j}(\boldsymbol{\theta}) = \sup_{P \in \mathcal{P}_{i,j}(\boldsymbol{\theta})} P = \lim_{\epsilon \to +0} \frac{1-\gamma_{c_j^\star}}{\max\{\epsilon, 1-\gamma_{c_i^\star} - r_i(\boldsymbol{\theta})\Delta\}}.$$

The supremum is achieved when the equality holds in the condition of Eq. (2). By transforming the condition, we obtain the following:

$$\begin{aligned} \gamma_{c_i^\star} + (1-\gamma_{c_i^\star})(r_i(\boldsymbol{\theta}))^{-1} &\leq \gamma_{c_j^\star} + (1-\gamma_{c_j^\star})(r_i(\boldsymbol{\theta})P)^{-1} \\ \gamma_{c_i^\star} r_i(\boldsymbol{\theta})P + (1-\gamma_{c_i^\star})P &\leq \gamma_{c_j^\star} r_i(\boldsymbol{\theta})P + 1 - \gamma_{c_j^\star} \\ P(1 - \gamma_{c_i^\star} - \Delta r_i(\boldsymbol{\theta})) &\leq 1 - \gamma_{c_j^\star}. \end{aligned}$$

When $1 - \gamma_{c_i^\star} - r_i(\boldsymbol{\theta})\Delta \leq 0$, since LHS will be negative and RHS will be positive, P can be infinitely large. Now, let's assume $1 - \gamma_{c_i^\star} - r_i(\boldsymbol{\theta})\Delta > 0$. Then we obtain the following:

$$P \leq \frac{1-\gamma_{c_j^\star}}{1 - \gamma_{c_i^\star} - r_i(\boldsymbol{\theta})\Delta} = P_{i,j}(\boldsymbol{\theta}).$$

When we assume the equality, Theorem 2 holds. □

Note that since we assume $r_i(\boldsymbol{\theta}) \geq 1$ and $\Delta = \gamma_{c_j^\star} - \gamma_{c_i^\star}$, the denominator of RHS is $1 - \gamma_{c_i^\star} - r_i(\boldsymbol{\theta})\Delta \leq 1 - \gamma_{c_j^\star}$. For this reason, the supremum is always larger than or equal to 1.

A.3 Proof of Theorem 3

To prove Theorem 3, we first show two lemmas.

Lemma 1. *Given a Γ-feasible domain with constraint thresholds of $c_i^\star$ for all $i \in [1, C]$, each constraint satisfies*

$$\forall i \in [1, C], \gamma_{c_i^\star} \geq \Gamma.$$

Proof. Let the feasible domain for the i-th constraint be $\boldsymbol{\Theta}'_{c_i^\star} = \{\boldsymbol{\theta} \in \boldsymbol{\Theta} | c_i \leq c_i^\star\}$. Then the feasible domain is $\boldsymbol{\Theta}' = \bigcap_{i=1}^C \boldsymbol{\Theta}'_{c_i}$. Since $\boldsymbol{\Theta}'_{c_i^\star}$ is a measurable set by definition and $\boldsymbol{\Theta}' \subseteq \boldsymbol{\Theta}'_{c_i^\star}$ holds, $\Gamma/\gamma_{c_i^\star} = \mu(\boldsymbol{\Theta}')/\mu(\boldsymbol{\Theta}'_{c_i}) \leq 1$ holds. Γ is a positive number, so $\gamma_{c_i^\star} \geq \Gamma$. □

Lemma 2. *The domain is $(\Gamma = 1)$-feasible domain iff:*

$$\forall i \in [1, C], \gamma_{c_i^\star} = 1.$$

Proof. Suppose $\gamma_{c_i^\star} < 1$ for some $i \in [1, C]$, we immediately obtain $\Gamma \leq \gamma_{c_i^\star} < 1$ from Lemma 1 and thus the assumption does not hold. For this reason, $\gamma_{c_i^\star} \geq 1$ for all $i \in [1, C]$ and since $\gamma_{c_i^\star} \leq 1$ by definition, $\gamma_{c_i^\star} = 1$ for all $i \in [1, C]$. □

Using Lemma 2, we prove Theorem 3.

Proof. From the assumption, $\Gamma = 1$ holds. Then $\gamma_{c_i^\star} = 1$ holds for all $i \in [1, C]$ from Lemma 2. By replacing $\gamma_{c_i^\star}$ with 1 in Eq. (6), we obtain the following:

$$\begin{aligned} \mathrm{ECI}_{f^\gamma}[\boldsymbol{\theta}|\mathcal{D}] &\propto \left(\gamma + (1-\gamma)\frac{g(\boldsymbol{\theta}|\mathcal{D}^{(g)})}{l(\boldsymbol{\theta}|\mathcal{D}^{(l)})}\right)^{-1} \prod_{i=1}^{C} 1 \\ &= \left(\gamma + (1-\gamma)\frac{g(\boldsymbol{\theta}|\mathcal{D}^{(g)})}{l(\boldsymbol{\theta}|\mathcal{D}^{(l)})}\right)^{-1} \\ &\propto \mathrm{EI}_{f^\gamma}[\boldsymbol{\theta}|\mathcal{D}] \end{aligned} \tag{3}$$

Since $\mathrm{ECI}_{f^\gamma}[\boldsymbol{\theta}|\mathcal{D}], \mathrm{EI}_{f^\gamma}[\boldsymbol{\theta}|\mathcal{D}] \in \mathbb{R}_{\geq 0}$ holds by definition, Eq. (3) indicates the statement of the theorem. □

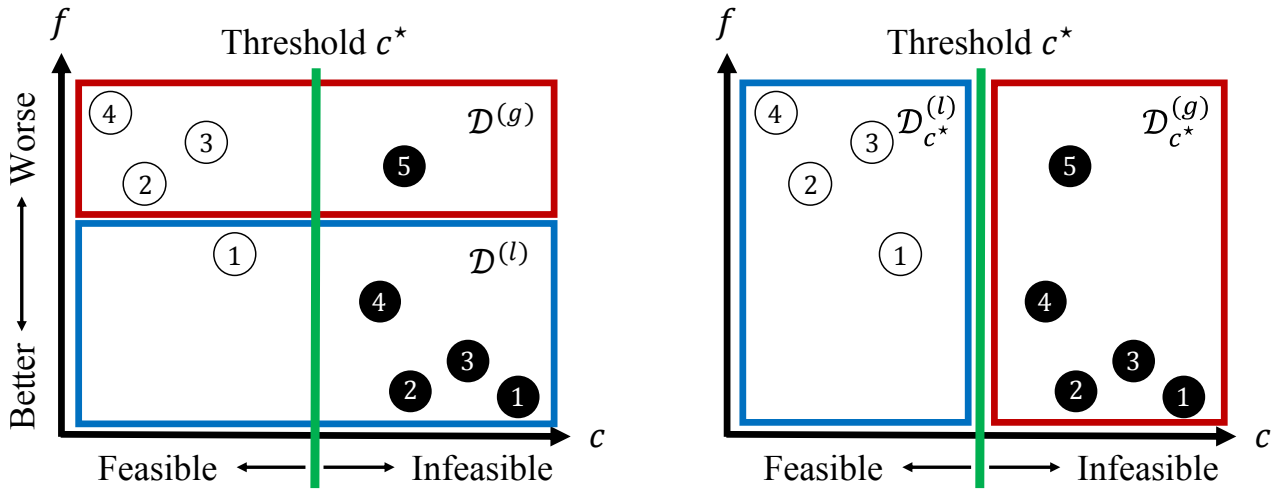

Figure 1: The conceptual visualizations of the split algorithm for the objective (left) and for each constraint (right). The black circles in the figures represent infeasible solutions and the white circles represent feasible solutions. The numberings for white and black objects stand for the ranking of the objective value in feasible and infeasible domains, respectively. The configurations enclosed by the red rectangle belong to the worse group and those enclosed by the blue rectangle belong to the better group.

B Further details of the split algorithm

In this section, we describe the intuition and more details on how the split algorithm works.

B.1 Split criterion of objective

Figure 1 presents how to split observations into better and worse groups. The left figure shows the split for the objective. In this example there are $N = 9$ observations and thus we will include $\lceil\sqrt{N}/4\rceil = \lceil\sqrt{9}/4\rceil = 1$ feasible solution in $\mathcal{D}^{(l)}$. For this reason, we first need to find the feasible observation with the best objective value and the white-circled observation 1 in the figure is the corresponding observation in this example. Then we split observations at this observation along the horizontal axis and $\mathcal{D}^{(l)}$ and $\mathcal{D}^{(g)}$ are obtained. The reason behind this modification is from the fact that observations with the best objective values are often far from the feasible domain (e.g. the black-circled observations 1 and 3 in Figure 1) and this is the case especially for tighter constraints. For example, Figure 2 visualizes the observations by c-TPE and the vanilla TPE on ImageNet16-120 of NAS-Bench-201 with $\gamma^{\text{true}}_{c^\star_i} = 0.1$. As seen in the figure, there are many observations with better performance than the oracle that are far from the feasible domain in the result of the vanilla TPE. When c-TPE prioritizes only such observations, c-TPE ends up searching the infeasible domain. For this reason, we include worse observations compared to the original TPE rather than just enforcing greedy selections of f^γ. This selection guarantees the overlap of the domain of the better group and the feasible domain as long as we already have feasible observations. This property prevents c-TPE from spending many evaluations that are far from feasible domains, but with good objective values. Notice that when the whole domain is feasible, all the observations will be feasible and thus this selection naturally converges to the same behavior as the original TPE.

B.2 Split criterion of each constraint

The right figure of Figure 1 shows the split of each constraint. Note that for simplicity, we show the 1D example and abbreviate $c_i, c^\star_i, \mathcal{D}^{(l)}_{c^\star_i}, \mathcal{D}^{(g)}_{c^\star_i}$ as $c, c^\star, \mathcal{D}^{(l)}_{c^\star}, \mathcal{D}^{(g)}_{c^\star}$, respectively. As illustrated in the figure, we take the observations with constraint values less than $c^\star$ into $\mathcal{D}^{(l)}_{c^\star}$ and vice versa. When the observations in the feasible domain do not exist, we only take the observation with the best constraint value among all the observations into $\mathcal{D}^{(l)}_{c^\star}$ and the rest into $\mathcal{D}^{(g)}_{c^\star}$. This selection increases the priority described in Theorem 2 and thus raises the probability of yielding feasible solutions quickly.

C Additional results for Section 4.2

In this section, we show the additional results for Section 4.2. The main goal of those results is to show how robust c-TPE is over various levels of constraints.

C.1 Performance over number of evaluations for all the datasets

Results on HPOlib

Figures 3, 4, and 5 show the time evolution of absolute percentage loss of each optimization method on HPOlib, NAS-Bench-101, and NAS-Bench-201 with the $\gamma^{\text{true}}_{c^\star_i}$-quantile of 0.1, 0.5, and 0.9 for the network size constraint.

For tighter constraint settings, c-TPE outperformed other methods except for HM2 and knowledge augmentation accelerated c-TPE in the early stage. For looser constraint settings, CNSGA-II improves its performance in the early stage of optimizations although c-TPE still exhibits quicker convergence. On the other hand, the performance of NEI and HM2 was degraded. As mentioned in Theorem 3, c-TPE approaches the performance of the vanilla TPE in the settings

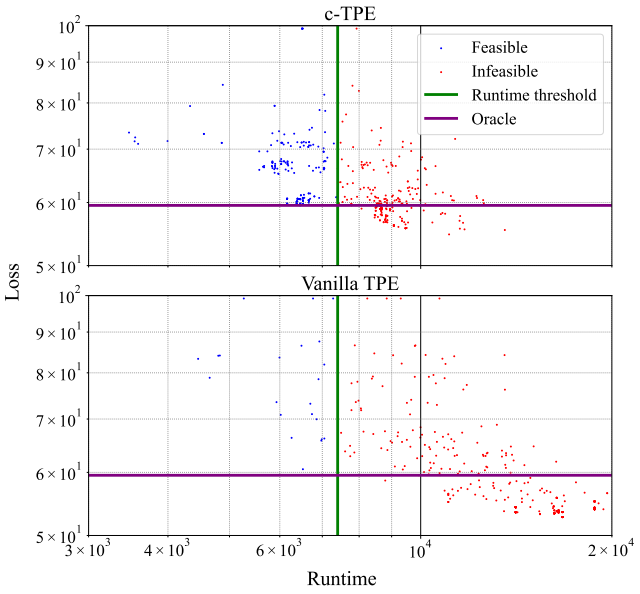

Figure 2: The visualization of the observations obtained by c-TPE (top) and the vanilla TPE (bottom) in the optimization of NAS-Bench-201 on ImageNet16-120 with $\gamma_{c_i^\star}^{\text{true}} = 0.1$. We include the observations in the latter half of each optimization to see the pure learning effect of each method and each optimization was run five times. Runtime threshold is chosen so that $\gamma_{c_i^\star}^{\text{true}} = 0.1$ will hold and oracle is the best loss value that can be achieved given a constraint value. The red dots are the observations that belong to the infeasible domain and the blue dots are the observations that belong to the feasible domain.

of $\gamma_{c_i^\star}^{\text{true}} = 0.9$ and thus such degradation does not happen to c-TPE. Furthermore, the contributions to the acquisition function from looser constraints decay and knowledge augmentation does not disrupt the performance of c-TPE thanks to this property.

For multiple constraints settings shown in Figure 5, both CNSGA-II and HM2 show slower convergence compared to single constraint settings. On the other hand, c-TPE shows quicker convergence in the settings as well.

Results on NAS-Bench-101

Figures 6, 7, and 8 show the time evolution of absolute percentage loss of each optimization method on HPOlib, NAS-Bench-101, and NAS-Bench-201 with the $\gamma_{c_i^\star}^{\text{true}}$-quantile of 0.1, 0.5, and 0.9 for the runtime constraint. Note that since we could not run NEI and HM2 on CIFAR10C in our environment, the results for CIFAR10C do not have the performance curves of NEI and HM2.

The results on NAS-Bench-101 look different from those on HPOlib and NAS-Bench201. For example, random search outperforms other methods on the tighter constraint settings of CIFAR10C. This is because the high-dimensional search space and tighter constraints made the information collection harder and thus each method could not guide itself although c-TPE still outperformed other methods on average. Additionally, knowledge augmentation still helps to yield better configurations quickly except CIFAR10C with runtime and network size constraints. As seen in the figures, the vanilla TPE exhibited better performance on looser constraint settings and thanks to the c-TPE's property that enables it to approach the original TPE formulation, c-TPE improves its performance in looser constraint settings.

Results on NAS-Bench-201

Figures 9, 10, and 11 show the time evolution of absolute percentage loss of each optimization method on HPOlib, NAS-Bench-101, and NAS-Bench-201 with the $\gamma_{c_i^\star}^{\text{true}}$-quantile of 0.1, 0.5, and 0.9 for the runtime and network size constraints. Note that the search space of NAS-Bench-201 is composed of six categorical parameters.

According to the figures, the discrepancy between c-TPE and the vanilla TPE is larger than HPOlib and NAS-Bench-101 settings. This means that there are many violated configurations that exhibit good performance. For this reason, the tighter constraint settings on NAS-Bench-201 are harder than the other benchmarks. However, c-TPE and HM2 showed better performance on tighter constraint settings. Additionally, c-TPE maintained the performance even over looser constraint settings while CNSGA-II and HM2 did not. This robustness is also from the property mentioned in Theorem 3.

C.2 Performance over $\gamma_{c_i^\star}^{\text{true}}$-quantile for all the datasets

Figures 12, 13, and 14 show the mean and standard error of the best performance in each optimization with various $\gamma_{c_i^\star}^{\text{true}}$. The results are visualized for each benchmark problem unlike other visualization so that we can see the best optimization methods for each problem and how close the final performance of each method is. In other words, a method is better when it is always close to the best performance rather than exhibiting only the best performance for some settings and the worst settings for others. Note that since we plot the final performance, knowledge augmentation, which accelerates the optimizations in the early stage, does not make big difference.

Results on HPOlib

According to Figure 12, while c-TPE was outperformed for some settings, the differences from the best performance are not large compared to those for CNSGA-II and HM2. CNSGA-II typically has larger differences in tighter constraints and HM2 has them in looser settings. Another point is that the performance of NEI and HM2 was not superior to the vanilla TPE when constraint quantiles are higher. This is due to the fact that NEI and HM2 do not converge to a single objective optimization unless the approximated probability improvements of each constraint return 1 over the whole domain. On the other hand, such performance degradation does not happen to c-TPE due to Theorem 3. However, knowledge augmentation for c-TPE does not help, except in tighter constraint settings although it does not disrupt the optimizations. This is because the results show the final performance of 200 evaluations and thus when c-TPE reaches 200 evaluations, it is competent enough to be able to find good solutions by itself.

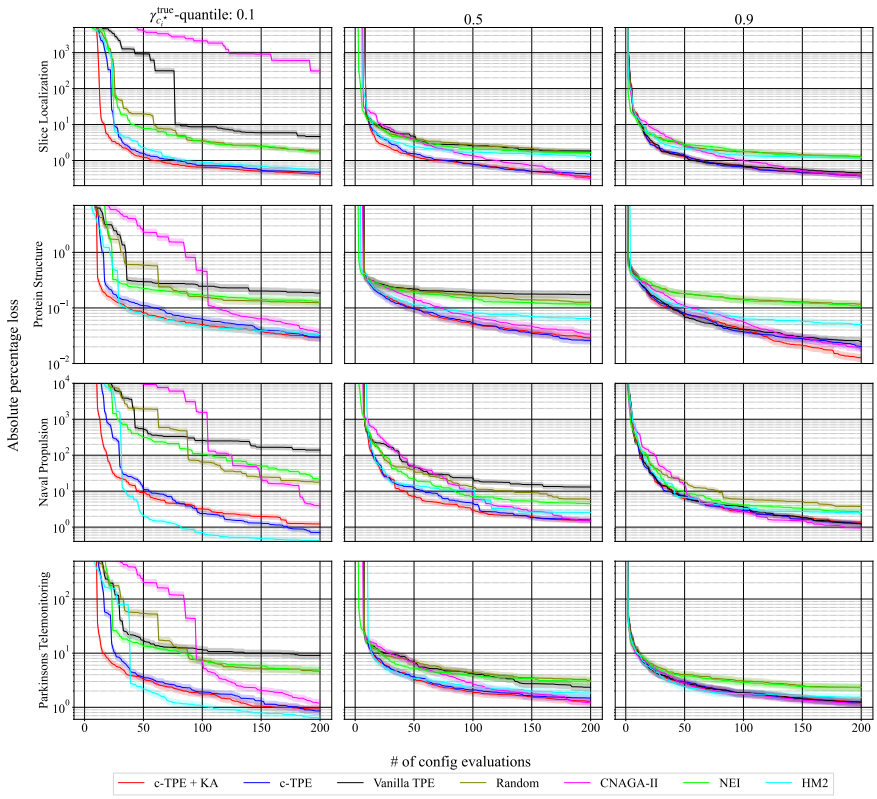

Figure 3: Figures show the performance curves on four benchmarks in HPOlib with a constraint of network size. We picked $\gamma_{c_i^\star}^{\text{true}} = 0.1$ (left), 0.5 (center), 0.9 (right). The horizontal axis shows the number of evaluated configurations in optimizations and the vertical axis shows the absolute percentage error in each experiment.

Results on NAS-Bench-101

Figure 13 shows the performance on NAS-Bench-101. As seen in the figures, c-TPE or c-TPE with knowledge augmentation exhibited the best performance except on CIFAR10C with runtime and network size constraints. This is because NAS-Bench-101 has high-dimensional search spaces and tighter constraints enforce each method to have only a fraction of observations that are useful to guide the optimization. The experiments on NAS-Bench-101 conclude that although the combination of tight constraints and higher dimensions might block the quick convergence of c-TPE, it is better to use c-TPE rather than other methods because the performance of c-TPE is stable over all the constraint levels on average.

Results on NAS-Bench-201

Figure 14 presents the results on NAS-Bench-201. As seen in the figure, while c-TPE and c-TPE with knowledge augmentation yielded the best performance for most settings or competitive performance if they are not the best, CNSGA-II and HM2 exhibited larger differences in some settings. This result implies that c-TPE is more robust compared to other methods although they also yield good performances in some settings.

D Additional results for Section 4.3

Figures 15, 16, and 17 show the average rank of each method over the number of evaluations. Each figure shows the performance of different constraint settings with 0.1 to 0.9 of $\gamma_{c_i^\star}^{\text{true}}$.

As the constraint becomes tighter, c-TPE converges quicker in the early stage of the optimizations in all the settings due to knowledge augmentation. On the other hand, knowledge augmentation does not accelerate the optimizations as constraints become looser. This is because it is easy to obtain information about feasible domains even by random samplings. However, knowledge augmentation does not degrade the performance of c-TPE and thus it is recommended to add knowledge augmentation as much as possible.

Furthermore, it is worth noting that although the performance of HM2 and NEI outperformed the vanilla TPE in the tighter constraint settings, their performance is degraded as constraints become looser and they did not exhibit better performance than the vanilla TPE with $\gamma_{c_i^\star}^{\text{true}} = 0.9$. On the other hand, c-TPE adapts the optimization based on the estimated $\gamma_{c^\star}$-quantile and thus it exhibited better performance than the vanilla TPE even in the settings of $\gamma_{c_i^\star}^{\text{true}} = 0.9$.

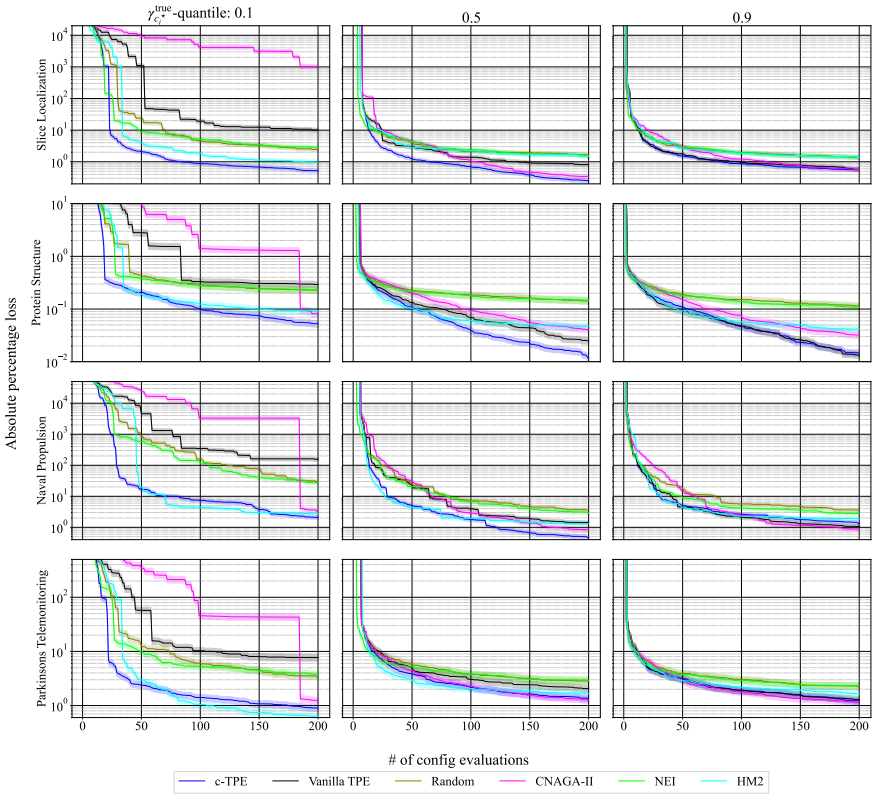

Figure 4: Figures show the performance curves on four benchmarks in HPOlib with a constraint of runtime.

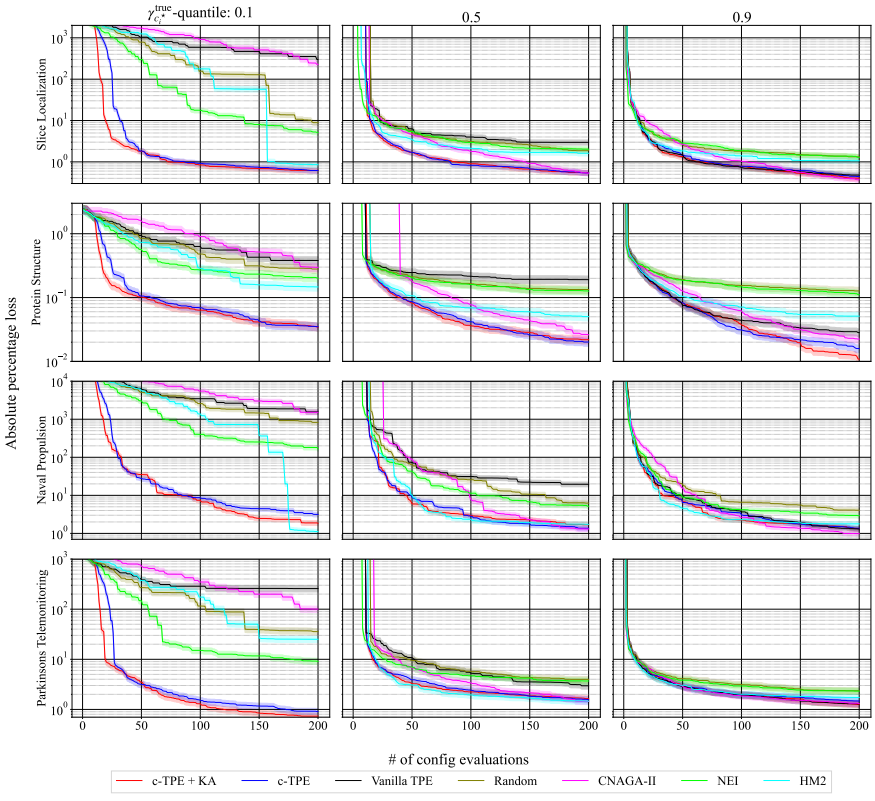

Figure 5: Figures show the performance curves on four benchmarks in HPOlib with constraints of runtime and network size.

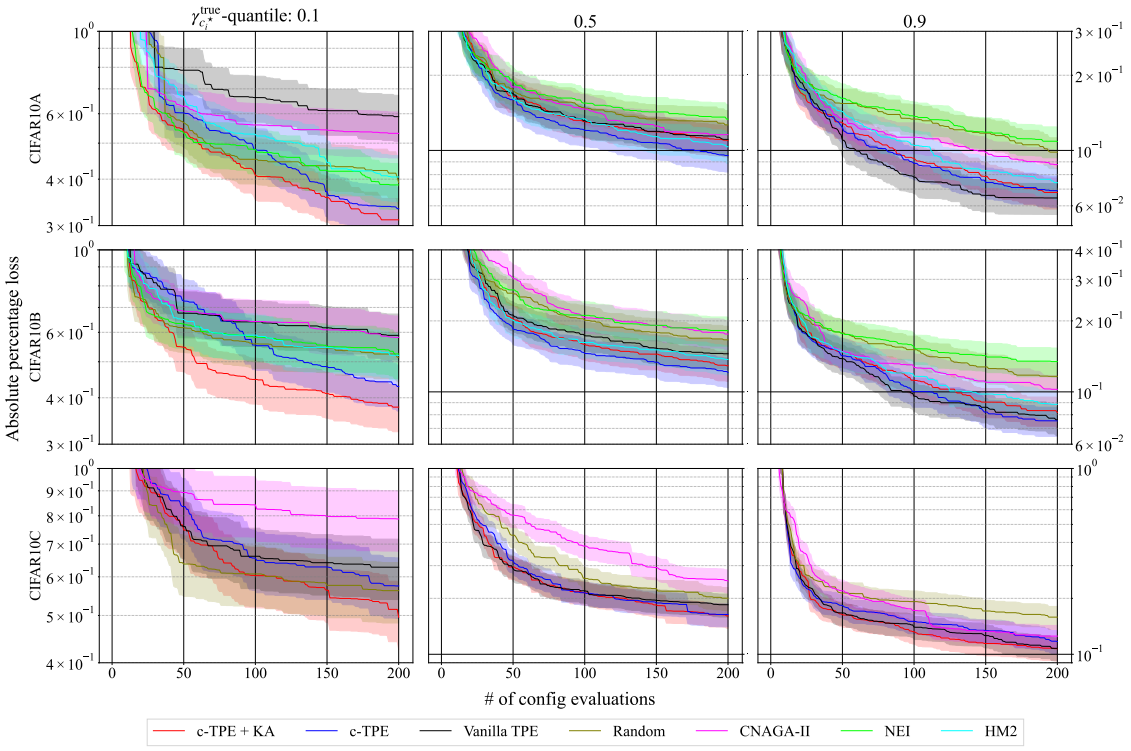

Figure 6: Figures show the performance curves on three benchmarks in NAS-Bench-101 with a constraint of network size. Note that since the scale of the results in $\gamma^{\text{true}}_{c^\star_i}$-quantile of 0.1 is different from others, we separately scaled for the readability.

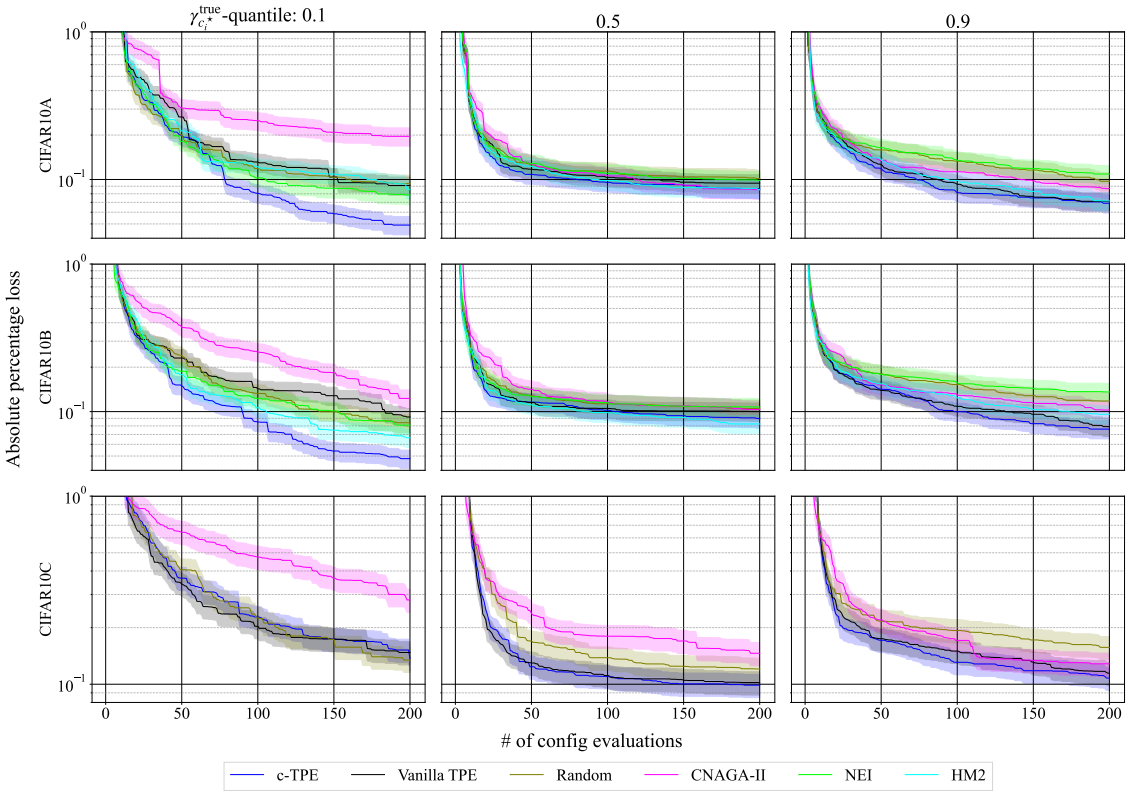

Figure 7: Figures show the performance curves on three benchmarks in NAS-Bench-101 with a constraint of runtime.

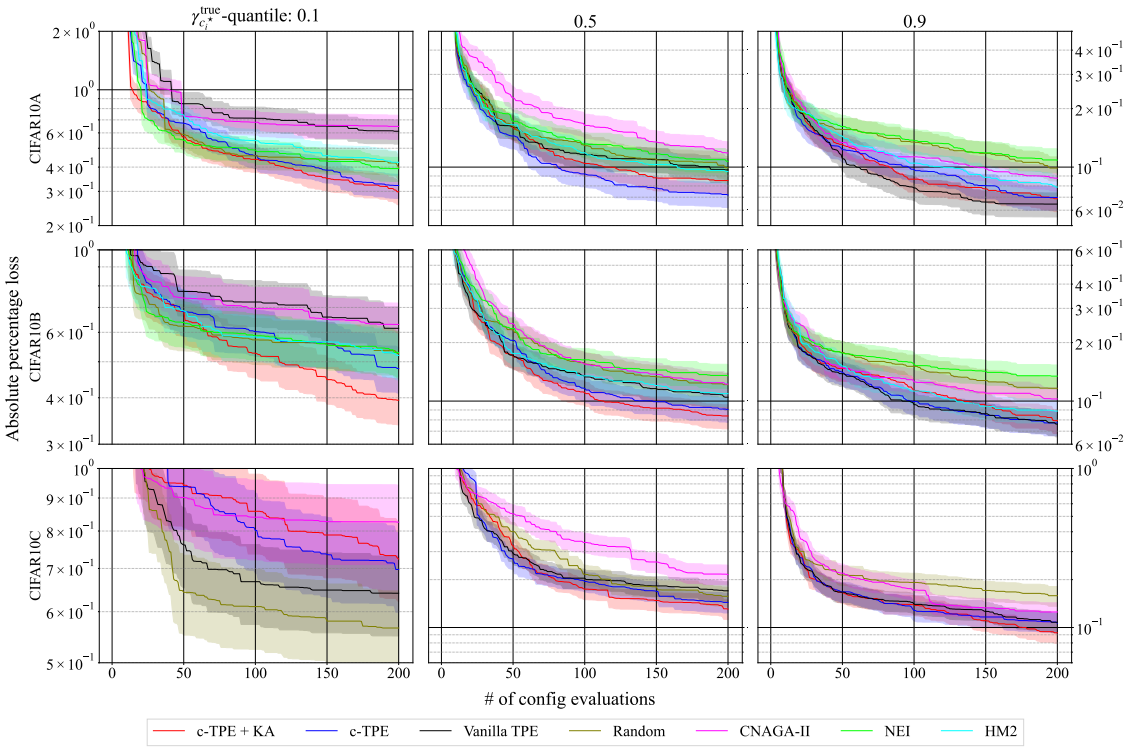

Figure 8: Figures show the performance curves on three benchmarks in NAS-Bench-101 with constraints of runtime and network size. Note that since the scale of the results in $\gamma^{\text{true}}_{c^\star_i}$-quantile of 0.1 is different from others, we separately scaled for the readability.

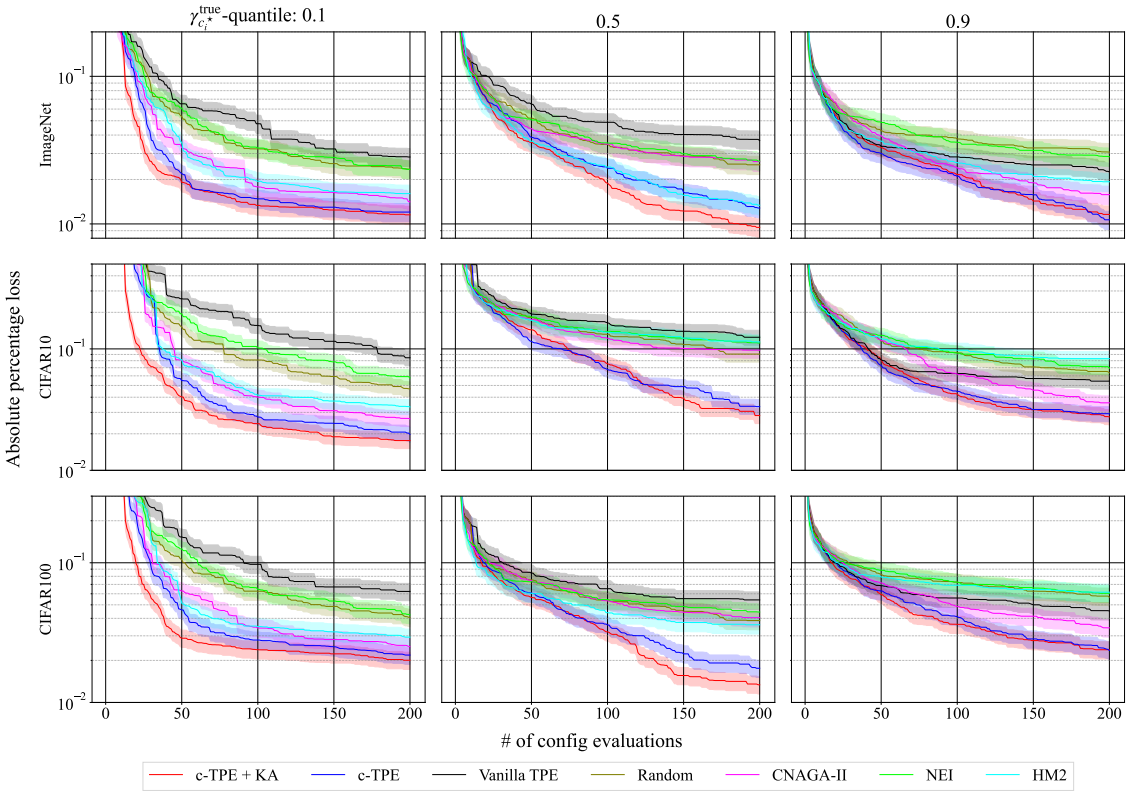

Figure 9: Figures show the performance curves on three benchmarks in NAS-Bench-201 with a constraint of network size.

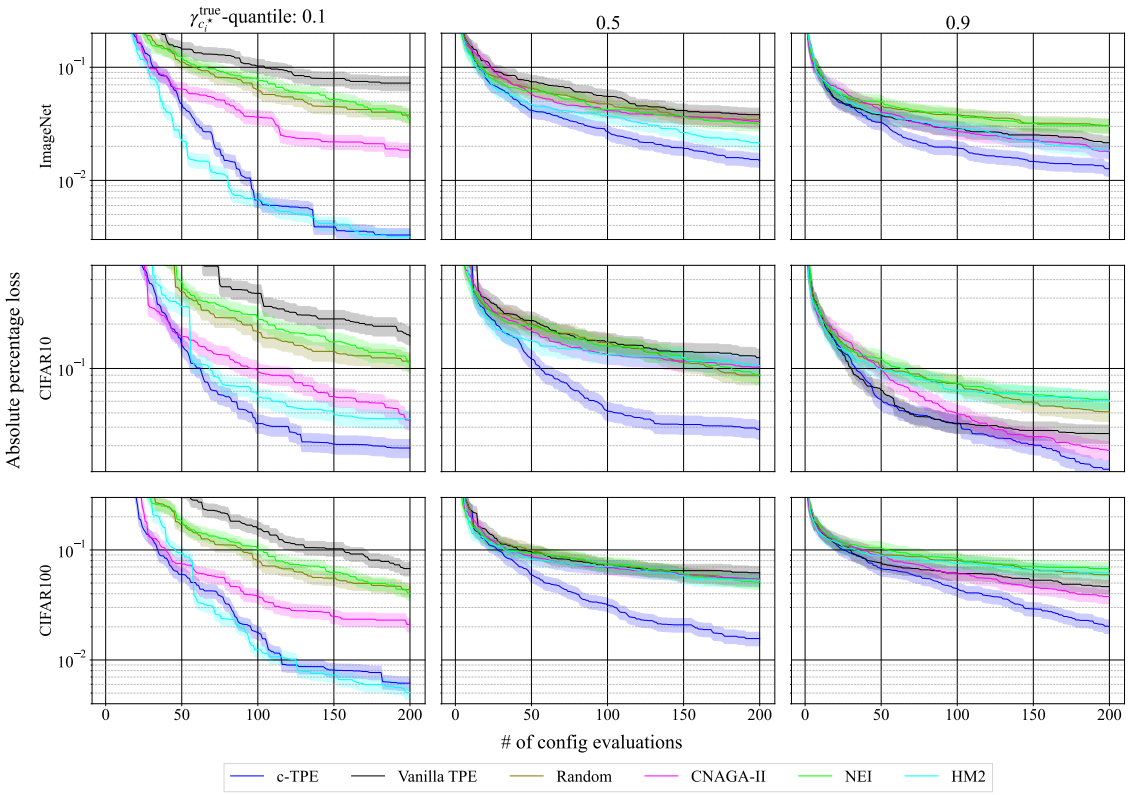

Figure 10: Figures show the performance curves on three benchmarks in NAS-Bench-201 with a constraint of runtime.

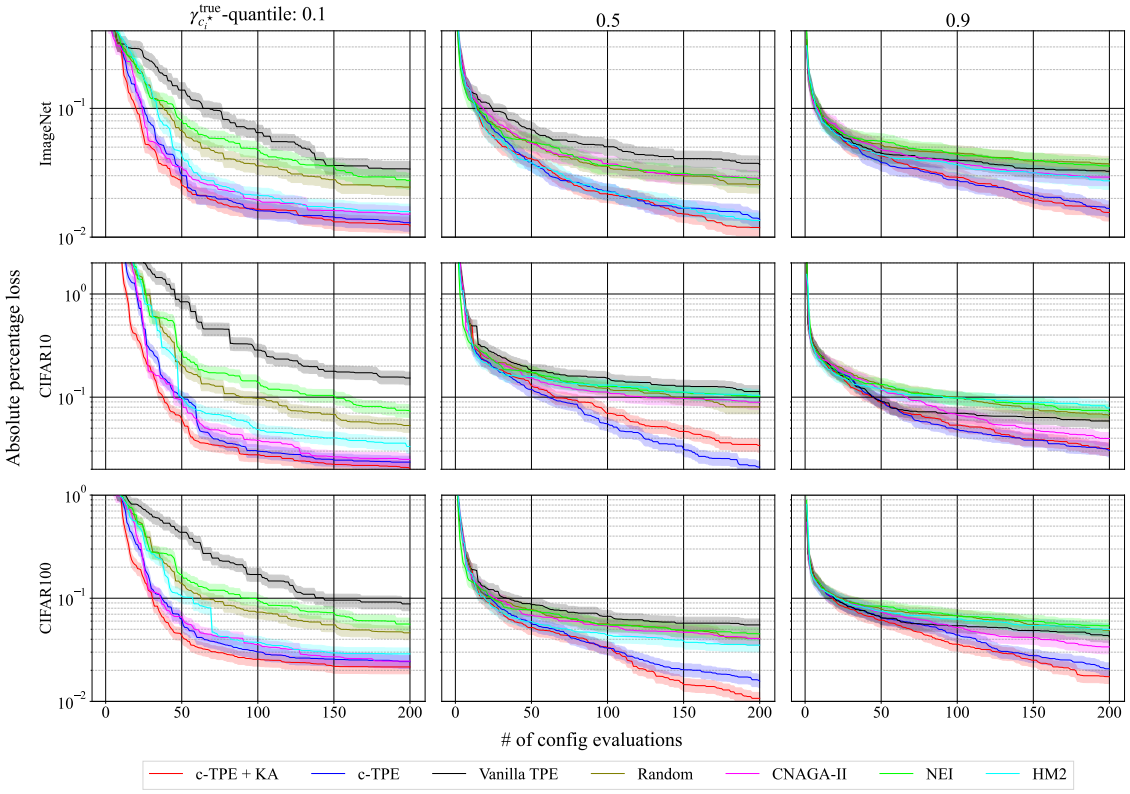

Figure 11: Figures show the performance curves on three benchmarks in NAS-Bench-201 with constraints of runtime and network size.

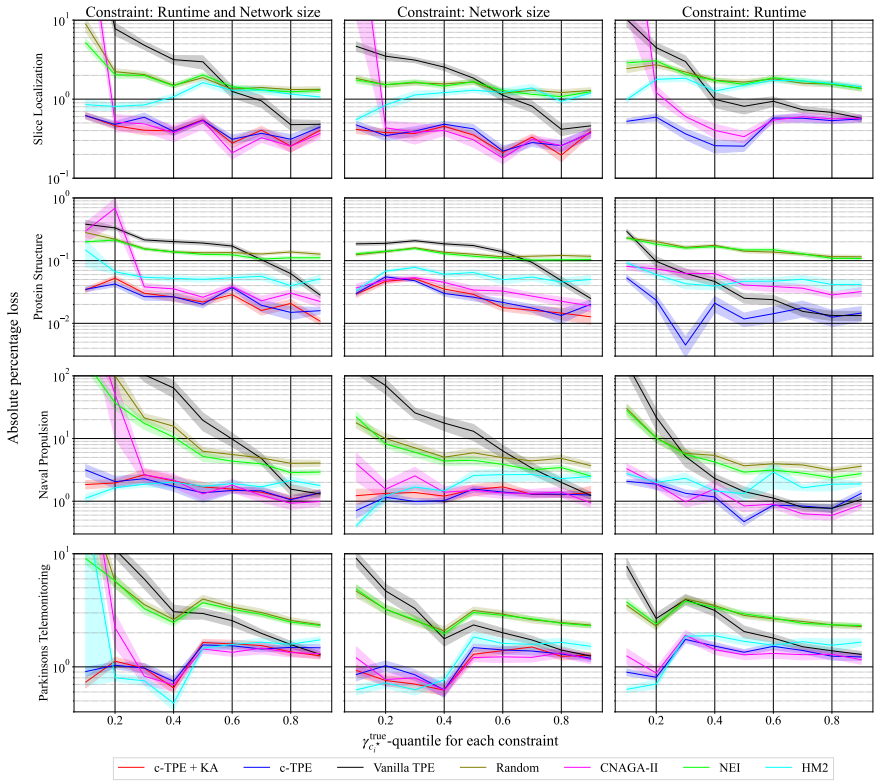

Figure 12: The best performance on HPOlib over various constraint settings and various constraint levels. Each optimization evaluated 200 configurations. Weak-color bands represent the standard error of the best performance over 50 random seeds. The horizontal axis represents $\gamma_{c_i^\star}^{\text{true}}$-quantile and the vertical axis represents the absolute percentage loss.

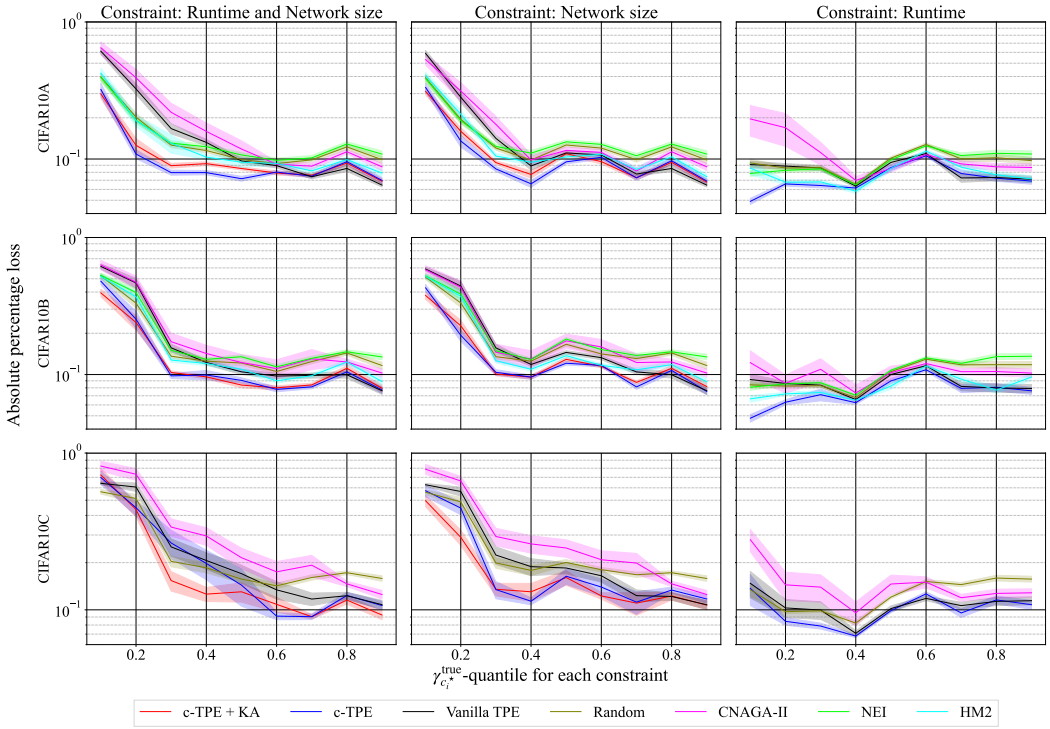

Figure 13: The best performance on NAS-Bench-101 over various constraint settings and various constraint levels. Each optimization evaluated 200 configurations. Weak-color bands represent the standard error of the best performance over 50 random seeds.

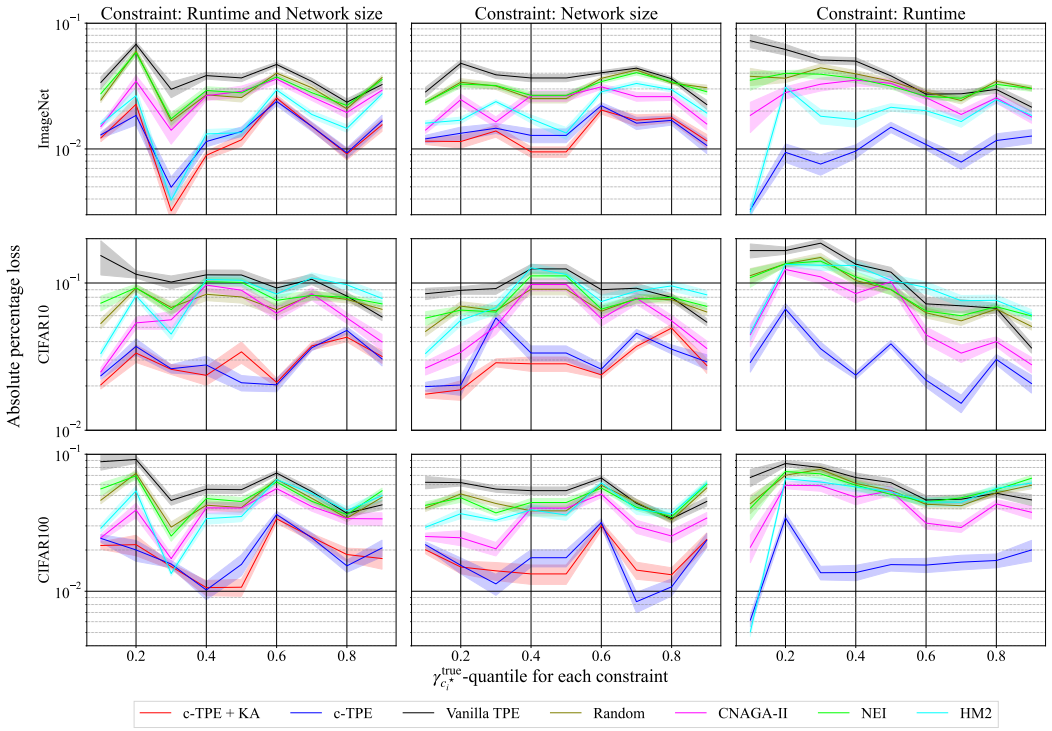

Figure 14: The best performance on NAS-Bench-201 over various constraint settings and various constraint levels. Each optimization evaluated 200 configurations. Weak-color bands represent the standard error of the best performance over 50 random seeds.

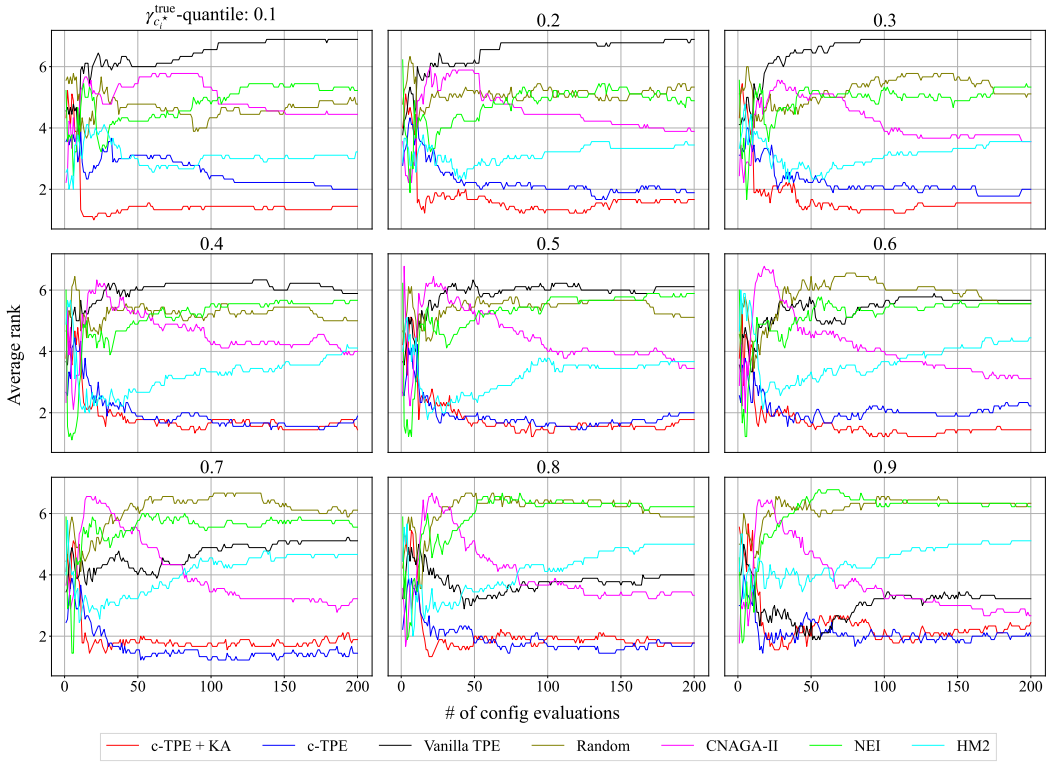

Figure 15: The average rank of each method over the number of evaluations. We evaluated each method on nine benchmarks with the network size constraint and each optimization was repeated over 50 random seeds. Each figure presents the results for $\gamma_{c_i^\star}^{\text{true}}$ of 0.1 to 0.9, respectively.

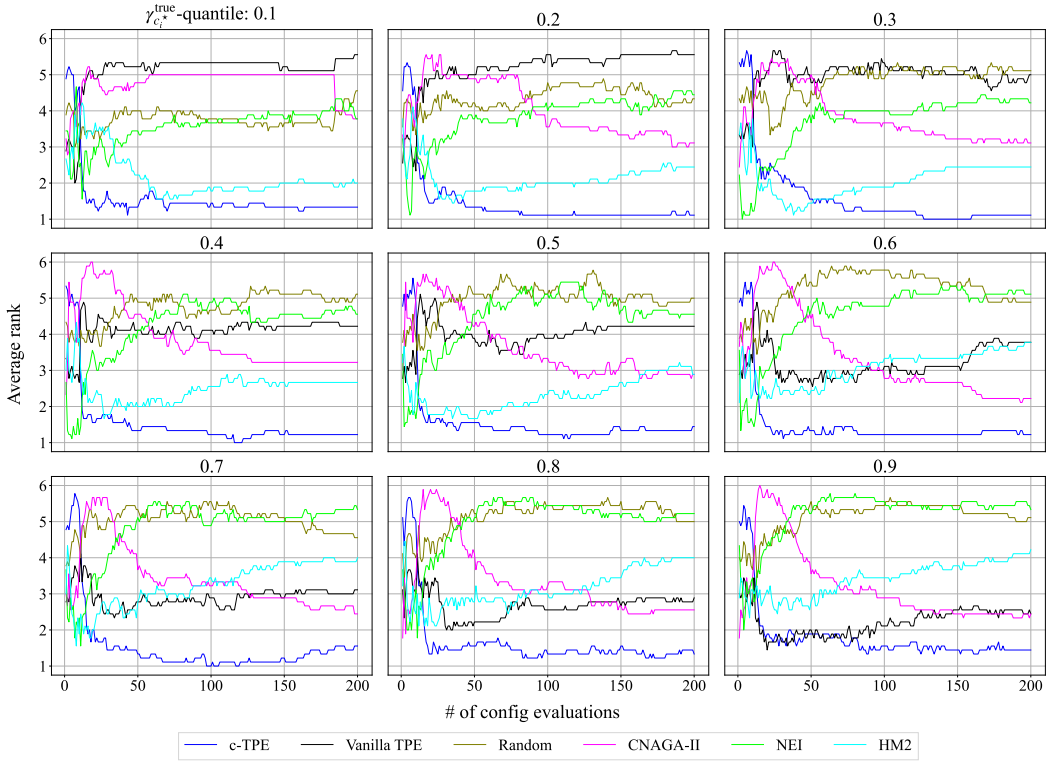

Figure 16: The average rank of each method over the number of evaluations. We evaluated each method on nine benchmarks with the runtime constraint and each optimization was repeated over 50 random seeds.

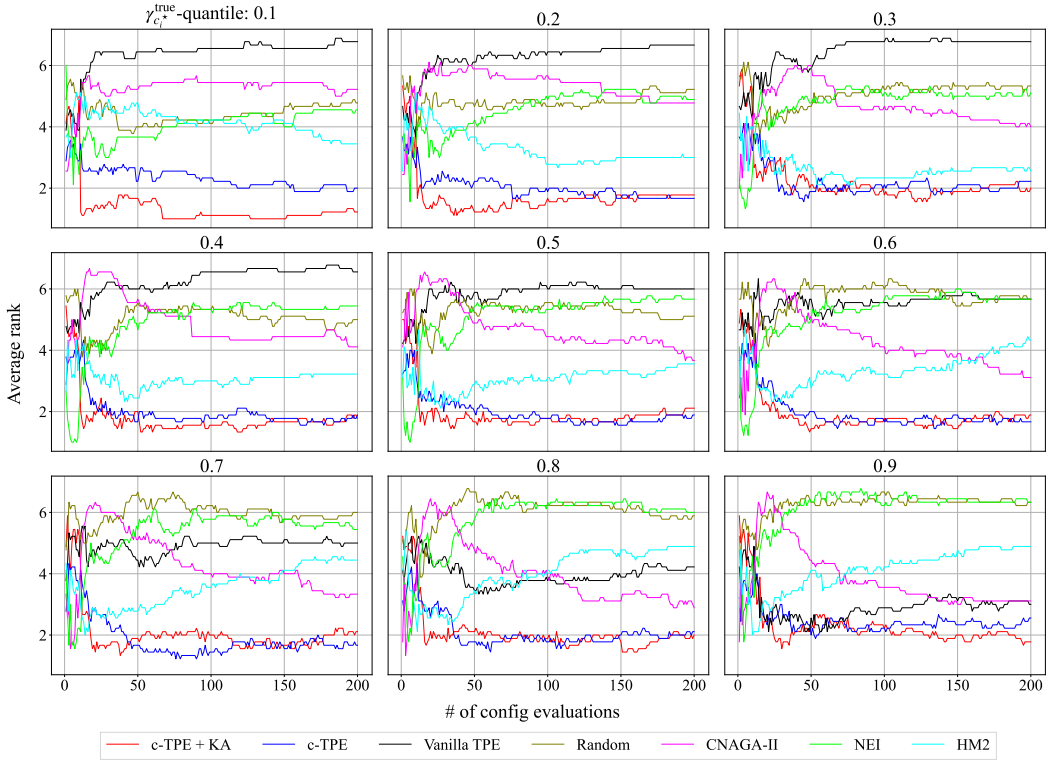

Figure 17: The average rank of each method over the number of evaluations. We evaluated each method on nine benchmarks with the runtime and the network size constraints and each optimization was repeated over 50 random seeds.