A Appendix

A.1 Scenarios for Parallel Mentoring with Multiple Proxies

A.1.1 Method

In the primary paper, we mainly focus on a scenario with three proxies. Here, we extend our method to incorporate M proxies. We revisit the two essential modules.

Voting-based pairwise supervision. We train M ($M \geq 3$) parallel proxies, $f_{\boldsymbol{\theta}}^{1}(\cdot)$, $f_{\boldsymbol{\theta}}^{2}(\cdot)$, $\cdots$, $f_{\boldsymbol{\theta}}^{M}(\cdot)$, each initialized differently, on the static dataset. Their mean is utilized as the final prediction:

$$f_{\boldsymbol{\theta}}(\cdot) = \frac{1}{M}(f_{\boldsymbol{\theta}}^{1}(\cdot) + f_{\boldsymbol{\theta}}^{2}(\cdot) + \cdots f_{\boldsymbol{\theta}}^{M}(\cdot)). \tag{10}$$

We generate the pairwise comparison labels $\hat{\boldsymbol{y}}^1$, $\hat{\boldsymbol{y}}^2$, $\cdots$, and $\hat{\boldsymbol{y}}^M$ for each proxy in the same way. We extend the subsequent majority voting part and derive the pairwise consensus labels $\hat{\boldsymbol{y}}^V$ via an element-wise majority voting:

$$\hat{\boldsymbol{y}}_{ij}^{V} = \text{majority\_voting}(\hat{\boldsymbol{y}}_{ij}^{1}, \hat{\boldsymbol{y}}_{ij}^{2}, \cdots, \hat{\boldsymbol{y}}_{ij}^{M}). \tag{11}$$

Here, i and j are the indexes of the neighborhood samples.

Adaptive soft-labeling. This module remains the same as it is designed for an individual proxy. We carry out fine-tuning and soft-labeling via bi-level optimization to adaptively mentor the proxy.

Setting on M. In Eq.(11), M can be any positive number greater than 2 as a decision may not be reached with just two proxies. In this study, we consider M as an odd number to ensure a decisive outcome in the voting process. Cases with an even number of proxies can be handled by adopting strategies like maintaining the original labels and skipping the fine-tuning step when the proxies are evenly split in their labels. However, we do not delve into these cases for brevity. We examine scenarios with M equal to 3, 5, 7, 9, and 11.

A.1.2 Experiments

We conduct experiments on the Ant task and the TFB8 task. The performance ratio comparing the performance of *parallel mentoring* to that of *tri-mentoring*, is computed as a function of M (the number of proxies). The results are displayed in Figure 6.

(1) Our observations indicate that as the number of proxies (M) increases, the performance ratios for both tasks initially improve, eventually reaching a plateau. This behavior suggests that an increased number of proxies enhances the robustness of the ensemble due to the increased diversity. However, this impact lessens as the number of proxies increases further, with the ensemble's robustness plateauing after a certain point. (2) Somewhat unexpectedly, the performance with $M = 7$ shows a slight dip on the Ant task. A possible explanation for this could be the dynamics of the voting system. When we have $M = 3$, some voting happens when two proxies agree but conflict with the third. However, when M increases to 7, voting may occur when four proxies align with one another but dissent with the remaining three. Such a scenario can make consensus labels less reliable, potentially explaining the poor performance of the $M = 7$ case on the Ant task. (3) Finally, it's important to note that adding more proxies also amplifies computational complexity. This increase could become a restricting factor when trying to scale the method to include a larger number of proxies.

A.2 Additional Results on 50th Percentile Scores

In the main paper, we presented the 100^{th} percentile scores. Here, we offer supplementary results on the 50^{th} percentile scores, which have been previously utilized in the design-bench work [1], to further validate the efficacy of *tri-mentoring*. Continuous task results can be found in Table 4 while discrete task results and ranking statistics are shown in Table 5. A review of Table 5 reveals that *tri-mentoring* achieves the highest ranking, demonstrating its effectiveness in this context.

A.3 Accuracy of Pairwise Consensus Labels

In addition to the performance results presented in the main paper, we also examine the accuracy of the optimized consensus labels $\hat{\boldsymbol{y}}^{S'}$. This analysis further substantiates the effectiveness of our

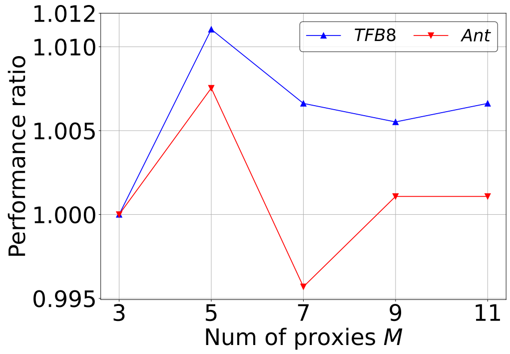

Figure 6: **Ratio of** performance with M proxies **to** performance with $M = 3$ proxies.

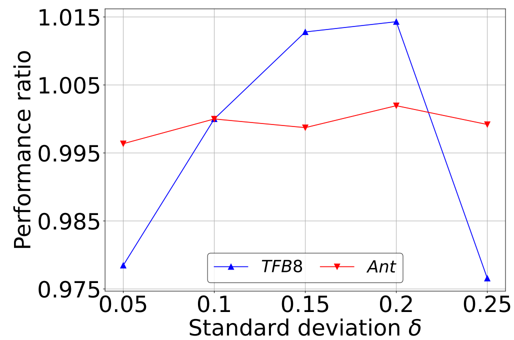

Figure 7: **Ratio of** performance with standard deviation δ **to** performance with $\delta = 0.10$.

Table 4: Results (median normalized score) on continuous tasks.

Method	Superconductor	Ant Morphology	D'Kitty Morphology	Hopper Controller
$\mathcal{D}$(**best**)	0.399	0.565	0.884	1.000
BO-qEI	0.300 ± 0.015	0.567 ± 0.000	0.883 ± 0.000	0.343 ± 0.010
CMA-ES	0.379 ± 0.003	-0.045 ± 0.004	0.684 ± 0.016	-0.033 ± 0.005
REINFORCE	$\mathbf{0.463 \pm 0.016}$	0.138 ± 0.032	0.356 ± 0.131	-0.064 ± 0.003
CbAS	0.111 ± 0.017	0.384 ± 0.016	0.753 ± 0.008	0.015 ± 0.002
Auto.CbAS	0.131 ± 0.010	0.364 ± 0.014	0.736 ± 0.025	0.019 ± 0.008
MIN	0.336 ± 0.016	$\mathbf{0.618 \pm 0.040}$	$\mathbf{0.887 \pm 0.004}$	0.352 ± 0.058
Grad	0.339 ± 0.015	0.564 ± 0.014	0.877 ± 0.005	0.384 ± 0.004
DE	0.333 ± 0.004	0.570 ± 0.011	0.875 ± 0.004	0.385 ± 0.007
GB	0.373 ± 0.013	0.550 ± 0.021	0.869 ± 0.009	0.374 ± 0.008
COMs	0.316 ± 0.022	0.568 ± 0.002	0.883 ± 0.002	0.346 ± 0.009
ROMA	0.368 ± 0.019	0.475 ± 0.036	0.856 ± 0.008	0.388 ± 0.007
NEMO	0.322 ± 0.008	0.593 ± 0.000	0.885 ± 0.000	0.361 ± 0.001
IOM	0.348 ± 0.022	0.516 ± 0.037	0.876 ± 0.007	0.368 ± 0.008
BDI	0.412 ± 0.000	0.474 ± 0.000	0.855 ± 0.000	$\mathbf{0.408 \pm 0.000}$
tri-mentoring	0.355 ± 0.003	0.606 ± 0.007	$\mathbf{0.886 \pm 0.001}$	0.391 ± 0.004

Table 5: Results (median normalized score) on discrete tasks & ranking on all tasks.

Method	TF Bind 8	TF Bind 10	NAS	Rank Mean	Rank Median
$\mathcal{D}$(**best**)	0.439	0.467	0.436		
BO-qEI	0.439 ± 0.000	0.467 ± 0.000	0.544 ± 0.099	$8.0/15$	$8/15$
CMA-ES	0.537 ± 0.014	0.484 ± 0.014	0.591 ± 0.102	$8.0/15$	$5/15$
REINFORCE	0.462 ± 0.021	0.475 ± 0.008	-1.895 ± 0.000	$10.6/15$	$14/15$
CbAS	0.428 ± 0.010	0.463 ± 0.007	0.292 ± 0.027	$12.7/15$	$12/15$
Auto.CbAS	0.419 ± 0.007	0.461 ± 0.007	0.217 ± 0.005	$13.3/15$	$13/15$
MIN	0.421 ± 0.015	0.468 ± 0.006	0.433 ± 0.000	$7.7/15$	$9/15$
Grad	0.532 ± 0.017	$\mathbf{0.529 \pm 0.027}$	0.443 ± 0.126	$6.1/15$	$6/15$
DE	$\mathbf{0.581 \pm 0.034}$	$\mathbf{0.534 \pm 0.014}$	0.474 ± 0.085	$5.4/15$	$4/15$
GB	0.503 ± 0.054	0.455 ± 0.020	$\mathbf{0.559 \pm 0.090}$	$7.3/15$	$6/15$
COMs	0.439 ± 0.000	0.466 ± 0.002	0.529 ± 0.003	$7.9/15$	$8/15$
ROMA	0.548 ± 0.017	$\mathbf{0.516 \pm 0.020}$	0.529 ± 0.008	$5.7/15$	$5/15$
NEMO	0.439 ± 0.018	0.456 ± 0.015	$\mathbf{0.568 \pm 0.021}$	$7.0/15$	$8/15$
IOM	0.437 ± 0.010	0.475 ± 0.010	-0.083 ± 0.012	$9.0/15$	$7/15$
BDI	0.439 ± 0.000	0.476 ± 0.000	0.517 ± 0.000	$6.6/15$	$7/15$
tri-mentoring	$\mathbf{0.609 \pm 0.021}$	$\mathbf{0.527 \pm 0.008}$	0.516 ± 0.028	**3.4/15**	**2/15**

method. For the D'Kitty and TFB8 tasks, we utilize the ground-truth function to determine the ground-truth pairwise labels. This enables us to assess the accuracy of $\hat{\boldsymbol{y}}^{S'}$. For easier accuracy computation, these labels are converted into one-hot labels.

(1) Recall that the pairwise comparison labels of a single proxy serve as its ranking supervision signals. In our analysis, we found that for a single proxy, 13.45% of pairwise comparison labels for the D'Kitty task and 8.38% for the TFB8 task differ from the optimized consensus labels $\hat{\boldsymbol{y}}^{S'}$. This reveals the extent to which our method modifies the original labels. (2) Further analysis shows that, of the conflicting optimized labels, 62.91% are accurate for D'Kitty and 63.16% are accurate for TFB8. These results reinforce the overall efficacy of our method. (3) When we remove the *voting-based pairwise supervision* module, we note a decrease in accuracy from 62.91% to 52.21% for D'Kitty and from 63.16% to 55.63% for TFB8. Similarly, omitting the *adaptive soft-labeling* module leads to a drop in accuracy from 62.91% to 57.16% for D'Kitty and from 63.16% to 60.86% for TFB8. These experiments underscore the crucial role of both modules in preserving the label accuracy.

A.4 Additional Analysis on Sensitivity to the Standard Deviation Hyperparameter

We further delve into how the standard deviation hyperparameter δ in neighborhood sampling, impacts the performance of our method. We experiment with δ values of 0.05, 0.10, 0.15, 0.20, and 0.25, with 0.10 being the default value employed in this paper. The results are normalized by dividing them by the result obtained for $\delta = 0.10$. As demonstrated in Figure 7, *tri-mentoring* exhibits remarkable robustness to variations in δ for both the continuous Ant and the discrete TFB8 tasks.

A.5 Broader Impacts

Our work could potentially expedite the development of new materials, biomedical innovations, or robotics technologies, leading to significant advancements in these areas. However, as with all powerful tools, there are potential risks if misused. One potential negative impact could be the misuse of this technology in designing objects or entities for harmful purposes. For instance, in the wrong hands, the ability to optimize designs could be used to create more efficient weapons or harmful biological agents. Therefore, it is crucial to implement appropriate safeguards and regulations on the use of such technology, particularly in sensitive areas.

A.6 Limitations

Despite the promising results demonstrated by our method, its performance is largely dependent on the accuracy of the design encoding. For tasks of high complexity, such as Neural Architecture Search (NAS) - which represents each design as a 64-length sequence of 5-category one-hot vectors - the performance of *tri-mentoring* is somewhat limited. This shortfall could be due to the default encoding technique of design-bench [1], which may fail to adequately capture the sequential and hierarchical nature of neural architectures, leading to ineffective gradient updates. This challenge suggests that, while our method provides a general framework for offline model-based optimization, task-specific techniques might be necessary for effective design encoding, especially in the context of complex tasks. Potential future research could explore ways of integrating problem-specific knowledge into the design encoding process to address these complexities more effectively.