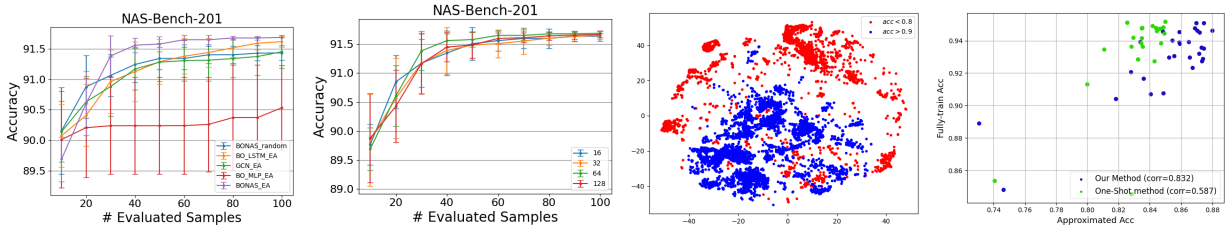

(a) **[R2, R4]** Ablation study. (b) **[R2]** Embedding size. (c) **[R3]** Visualization. (d) **[R3, R4]** Weight sharing.

[R1, R4] *"meaning of top-1 error? Error bar ..."*, *"variance for the open-domain search"*: Top-1 error is simply the classification error (a common terminology in image classification). The ImageNet experiment (Table 3) is expensive, and so results reported are based on one single run (as in other papers). To show error bars, we use the open-domain search experiment on CIFAR-10 (Table 2) instead, and re-train the architectures 5 times. The error bars (mean $\pm$ std) are: BONAS-A (2.69 ± 0.05); BONAS-B (2.54 ± 0.04); BONAS-C (2.46 ± 0.03); BONAS-D (2.43 ± 0.03).
[R1] *"compare with some HPO methods such as BOHB in the closed domain search"*: BONAS outperforms BANANAS, which in turn is better than BOHB (as shown in Figure 5.2 of BANANAS's arxiv version).
[R1] *"Will the code and the LSTM-12K be open-sourced?"*: Yes, as discussed in footnote 3.
[R1] *"The final results in Table 2 is not fair."*: Agree, we will remove it.

[R2] *"robustness against such a hyper-parameter setting"*: We use the same hyper-parameter setting for all 3 benchmarks. Figure (b) above shows that the performance is robust to different GCN embedding sizes.
[R2, R4] *"nice if LSTM- or MLP-based surrogate model is compared"*, *"difficult to disentangle each component...clarify; EA... degree to help"*: Figure (a) shows ablation study on NAS-Bench-201, which varies each component (surrogate model/sampling method/BO) in the model. The other experimental settings are the same as in Section 4.2.

[R3] *"whether the discrete structure is well preserved in the continuous space"*: Figure (c) shows TSNE embedding of the architecture embedding vectors. As can be seen, more accurate architectures are close to each other. Table 1 also shows that one can obtain accurate prediction of the performance using the architecture embedding.
[R3] *"whether it can guarantee the subnets accuracy trained with weight sharing correlates well with its final accuracy ... appreciated if the authors can show sampled subnets trained with weight sharing has high correlation"*: We perform an experiment similar to that reported in Figure 8 of Appendix, and compute the correlation between approximated accuracy by weight-sharing and accuracy of fully-trained model. Instead of using 100 random models as in Figure 8, we use 100 subnets that are trained with weight-sharing in the same round. Figure (d) above shows that the correlation is high (0.832).
[R3] *"which dimensionality you emb the architecture? BO typically works better in low-dimensional..."*: We use 64-d (line 227 in paper). In Table 7 of "Scalable Bayesian optimization using deep neural networks", it also uses $\{50, 100, 200\}$ as embedding dimensionality.
[R3] *"weight-sharing among related architectures are more reliable than those in the one-shot approach"*: Standard one-shot methods perform weight-sharing on a large set of subnets. These subnets can be very different and so sharing their weights may not be a good assumption. Our method performs weight-sharing on a small subset of subnets, which are similar in that they have high UCB scores (step 10 in Algorithm 2). Hence, sharing weights for this smaller subset of similar-performance subnets may be more reasonable (as also verified by the high correlation between the actual and approximate accuracies in Figure (d) mentioned above).
[R3] *"BANANAS's encoding sounds more like a computational-aware encoding ..."*: Agree, and we will clarify it.

[R4] *"not sure that .6 vs a .7 correlation is really that big of a difference"*: In Figure 8 of the Appendix, the models are randomly sampled. Here, in Figure (d) above, we use subnets that are sampled in the same search iteration. The correlation is higher (0.832), demonstrating that weight-sharing among a smaller subset of similar models is better.
[R4] *"a little confused about was the comparison of weight sharing..."*, *"were weights reinitialized each time?"*: Weights are reinitialized each time across sampling rounds. Indeed, trained weights in ENAS is inherited along the whole search process, but this approach may have some flaws: networks evaluated later are trained with longer budget than those evaluated earlier, which may render the evaluation score unfair and cause misleading. Reinitializing the weights in each sampling round can alleviate this issue since each sub-network is trained for the same iterations.
[R4] *"for the GCN, were alternatives to a global node considered? For example, it is common to see pooling"*: Yes, we also tried differentiable pooling. The correlation is similar to adding a global node (0.840 vs 0.841 on NAS-Bench-101). Thus, GCN propagation part is more important than how to add global node.
[R4] *"how was 100 decided upon for the number of candidates? how changing this number"*: This needs to be large enough to accelerate search, but small enough to enable efficient weight sharing and exploration. We simply chose 100 as a convenient choice.
[R4] *"correlation at the beginning and how does it improve over the course of the search"*: The following are correlation results over the course of search on NAS-Bench-201.

# samples	10	50	100	200
correlation	0.093	0.377	0.486	0.634