We thank all the reviewers for their insightful comments! All the responses will be incorporated into our revision.

R1: **(1)** We designed a variational graph isomorphism network to injectively encode structural information of networks in the latent space and accurately remap to original structures after latent space optimization. **(2)** The observations are pretrained embeddings of the selected neural architectures. **(3)** The searched networks are trained from scratch.

R2: **(1)** Details of supervised learning approach: architecture embeddings and search strategies (e.g., BO) are jointly optimized in a supervised manner. The supervision signal for embedding learning comes from the accuracies of architectures selected by the search strategies. In addition to accuracy, NAO takes the reconstruction loss of $\hat{A}$ and $\hat{X}$ into account. However, as reported in our submission or Table 1 below, its performance is inferior to our unsupervised approach as it cannot necessarily improve embedding learning due to entangling structure reconstruction and accuracy prediction together. **(2)** Superiority of pretrained embeddings: compared to supervised embeddings, the pretrained embeddings are able to better capture the structural information (e.g. edit distance measures) of neural networks. This is because the optimization objective in pretraining is structure reconstruction only. As we showed in Figure 3 and 4 in the submission, compared to supervised learning, pretraining makes similar architectures clustered better (Figure 3), and hence the accuracies are clustered and distributed more smoothly in the latent space (Figure 4). Conducting architecture search in such smooth performance surface is much easier and is hence more efficient. Note that we only use the accuracy of architecture as supervision in the search phase. **(3)** How pretrained embeddings are used with BO and RL for architecture search: for BO, the pretrained embeddings are passed to Bayesian optimization algorithm (DNGO) to select the top-K architectures in each round of search. For RL, the pretrained embeddings are passed to the Policy LSTM to sample the action and obtain the next state (valid architecture embedding) using nearest-neighborhood retrieval to maximize accuracy as reward. We covered some details in Supplementary A. We will add a thorough description of how pretrained embeddings are used with search strategies in the revision. **(4)** Fine-tuning: we did not fine-tune the embeddings during search based on the performance of the architectures. This is also because it biases the structural clustering obtained from pretraining, which leads to inferior search performance. We will add this result in the revised version. **(5)** Colorscale jumps (red and black) in Figure 4: we overlaid the original colorscale with red (>92% accuracy) and black (<82% accuracy) for highlighting purpose. **(6)** Naming observations: we will name our observations to reflect their nature in the revision. **(7)** Reproducibility: to facilitate fully reproducing our results, we attached the source code in our submitted supplementary material.

R3: **(1)** We report the result of GD on NAS101 in terms of test regret in Figure 1 and number of samples in Table 1. We have two observations. First, for GD, NAS with pretrained embeddings outperforms supervised embeddings. This aligns with our results in RL and BO. Second, GD performs worse than RL and BO in both unsupervised and supervised methods. This could be attributed to how GD minimizes the prediction error, which could easily enter the local minimum. We will add this result in the revised version. **(2)** Supervised embeddings are less capable of preserving the structural information due to the learning bias introduced by predicted accuracy, and thus are distributed less smoothly in the latent space which results in more overlapped (or blank) areas. **(3)** Thanks for suggesting [1,2]. [1] focuses on network generators that output relational graphs, and the predictive performance highly depends on the structure measures of the relational graphs. In contrast, we encode structural information of neural networks into compact continuous embeddings, and the predictive performance depends on how well the structure is injected into the embeddings. [2] focuses on transforming adjacency matrix-based encoding to path-based encoding in the discrete space. In contrast, we focus on encoding adjacency matrix-based architectures to low-dimensional embeddings in the continuous space. We will add the discussions on [1,2] in the revised version.

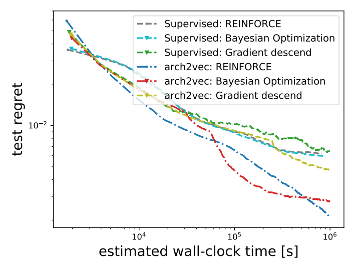

Figure 1: Test regret of GD & others.

R4: **(1)** Thanks for suggesting the related work. While the related work tackles the generative problems, our work focuses on mapping the finite discrete neural architectures into the continuous latent space regularized by KL-divergence such that each architecture is encoded into a unique area in the latent space. Importantly, we systematically investigate how pretraining preserves the structure of neural networks and affects their predictive performance in NAS. We will emphasize this distinction in the revised version. **(2)** The KL term is used to regularize the mapping from the discrete space to the continuous latent space. It helps to perform a better inference and to preserve the validity performance of the model. We show the effectiveness of using KL for pretraining on three search spaces in Table 2 below. We will add this result in the revision.

NAS Methods	**#Queries**	**Accuracy (%)**	**Encoding**	**Search Method**
NAO	1000	93.74	Supervised	GD
GD (ours)	400	93.69	Supervised	GD
RL (ours)	400	93.74	Supervised	REINFORCE
BO (ours)	400	93.79	Supervised	BO
arch2vec-GD	400	93.85	Unsupervised	GD
arch2vec-RL	**400**	**94.10**	Unsupervised	REINFORCE
arch2vec-BO	400	94.05	Unsupervised	BO

Table 1: Number of samples of GD & others.

Method	**NAS-Bench-101**			**NAS-Bench-201**			**DARTS**		
	Accuracy	Validity	Uniqueness	Accuracy	Validity	Uniqueness	Accuracy	Validity	Uniqueness
arch2vec (w.o. KL)	**100**	30.31	99.20	**100**	77.09	96.57	99.46	16.01	99.51
arch2vec	**100**	**51.33**	**99.36**	**100**	**79.41**	**98.72**	**99.79**	**33.36**	**100**

Table 2: An ablation study on the effectiveness of KL for pretraining.