We thank all the reviewers for the valuable comments and suggestions.

To Reviewer #1: **(1) [*Why semi-supervised method helps?*]** We have a framework consisting of an encoder, a predictor and a decoder, where the encoder and decoder together act as an autoencoder to learn the representation of architectures via reconstruction task. This enables improvements when training with additional pseudo-labeled architecture-accuracy pairs. Besides, we indeed use dropout as in NoisyStudent (the paper you mentioned) to help generalization. We will add these discussions in the paper. **(2) [*Training on 50 pairs*]** It leads to severe performance drop.

To Reviewer #2: **[*Why SemiNAS but not SemiNAO*]** The basic idea of SemiNAS, leveraging unlabeled architectures via the encoder-predictor-decoder framework and predicting the accuracy of candidate architectures to boost the search process, is general and can be applied to various NAS algorithms as discussed in Section 3.3. NAO is only chosen as a demonstration example. We also combine SemiNAS with other NAS algorithm (e.g., Regularized Evolution) and conduct experiments in Table 1 to further verify its effectiveness. It is also easy to apply SemiNAS to RL based NAS methods, by predicting the accuracy of an architecture as the reward. We will add such experiments in the new version.

To Reviewer #3: **(1) [*Results in Table 1*]** SemiNAS (RE) only uses 1000 (half of original RE uses) architecture-accuracy pairs to achieve comparable accuracy, which is to show that SemiNAS can reduce the resources required. We also run SemiNAS (RE) consuming 2000 pairs to compare with RE under the same number of queries, and it achieves 94.03% test accuracy which outperforms RE. **(2) [*Standard deviation on CIFAR-10*]** Though NASBench-101 is conducted on CIFAR-10, there exist some differences. It runs each model for 3 times and collect the 3 results to reduce the variance. Moreover we run the experiment for 500 times suggested by the authors of NASBench-101 to further reduce the variance. We show that even 0.1% is already a significant improvement on NASBench-101 via statistical method in line 191. More discussions on how to compare different algorithms via test regret and ranking for better interpretation are included in lines 191-203. **(3) [*Comparison with EfficientNet*]** Thanks for the suggestion! We will add EfficietNet-B0 for a comparison. **(4) [*Why built upon NAO?*]** Please refer to our response to reviewer #2. As for one-shot search, it usually uses weight sharing to reduce the time of training architectures as in ENAS and is orthogonal to the core search algorithm. Our experiments on ImageNet and TTS use one-shot search with weight sharing. **(5) [*The novelty and comparison with other SSL works*]** For the novelty, please refer to our response to Reviewer #1 and #2. The related SSL works you mentioned focus on proposing novel SSL methods/algorithms; while our focus is to boost NAS by SSL rather than proposing a new SSL algorithm (we can choose any SSL method) and mainly compare with other NAS works. We will cite and adopt the mentioned SSL methods in the new version, and further explore more advanced SSL techniques into NAS in future works. **(6) [*Lines 130-141*]** You are right! We will polish this part to make it clearer. **(7) [*Parameterization of encoder/ predictor/decoder*]** The weights of encoder, predictor and decoder are independent without sharing. As in line 109, the predictor (f_p) is a multi-layer fully connected network with relu activation. **(8) [*Unlimited unlabeled architectures and extremely few labeled architectures?*]** The gain from unlabeled architectures will become saturated when the number of unlabeled architectures continuously increases, as shown in the Appendix. We only use 100 architecture-accuracy pairs (N=100) in our experiments on ImageNet and TTS, and further reducing labeled architectures leads to performance drop.

To Reviewer #4: **(1) [*Selection of N, M, K*]** We mainly study different M in the Appendix. For N and K, it is obvious that larger values will result in better performances. Considering the resources constraints and our motivation, we do not explore larger N and K. We mainly explore how small N can be to achieve comparable performance. We find that N should be at least 100 and smaller N leads to severe performance drop. **(2) [*"Top-ranked 42"*]** Seems you misunderstood Table 1. The ranking in Table 1 indicates the ranking of the discovered architecture among all the candidate architectures in NASBench-101, rather than the ranking of specific NAS algorithm (in your comments). It does not mean that there exist 42 other NAS algorithms that are better than SemiNAS. **(3) [*Comparison on ImageNet with other works.*]** We follow the search space and tricks in ProxylessNAS, and mainly compare to works with the same setting, while some other works use additional tricks/modules (e.g., swish activation, squeeze-and-excitation, auto data augmentation) that can improve the accuracy for several points. We will add more missing related works to Table 2 for general comparisons. **(4) [*Experiments on TTS*]** We evaluate the MOS for the robustness test experiment. The MOS for Transformer TTS, NAO and SemiNAS are respectively 2.03, 2.22, 2.43, which shows the advantages of SemiNAS. Note that we just use Griffin-Lim as the vocoder for quick comparison, and will use neural vocoder to improve the MOS score in the new version of the paper. **(5) [*Multiple runs on NASBench*]** Running for 500 times is suggested in the original NASBench-101 paper by its authors. We just follow this to fairly compare with other works. **(6) [*Algorithm 1*]** We simplify the processes of NAO in Algorithm 1 by using simple description instead of complicated equations to let the reader focus on the semi-supervised learning method rather than the underlying search algorithm. We will add more details and explanations to make this part clearer and easier to understand. **(7) [*Confidence interval in line 192*]** It is based on bootstrap resampling. **(8) [*Best test accuracy in NASBench-101*]** NASBench-101 contains 423k architectures and their evaluated test accuracy on CIFAR-10, among which the highest test accuracy is 94.32%, which is the goal for NAS algorithms to achieve.