To Reviewer#1 Q1. *The method needs to search through paths for long-term information, it's like to find conflict facts or relations. How to deal with the conflict is not mentioned.* **Reply:** The conflict problem in BootEA [42] is by their bootstrapping training strategy, i.e., adding generated facts (not necessarily true) in training data. Same as baselines [11,19,46,49] and survey [48], we work on KG with standard training sets (i.e., with only true facts and no conflicts).

Q2. *It would be interesting to include 2 extra methods, SRAP with only macro and SRAP with only micro, to illustrate the effectiveness of macro/micro split.* **Reply:** All the architectures are defined by micro+macro part and thus cannot be evaluated separately. In Fig. 5, we have "search with only macro stage" (*Random*, *Reinforce*, *Bayes*) and "with only micro stage" (*ASNG*, *DARTS*). We will elaborate more on this in Sec. 4.4. Besides, in appendix Sec. B.2 (Fig. 8), we show that the split is reasonable.

Q3. *Some recent work on graph alignment were not included in the comparison.* **Reply:** For the three papers, the first one BootEA [42] and the third one GCN-Align [49] are included in our submission and are carefully compared in Sec. 4.3 (see Tab. 4). The code of VR-GNN [Ye et al. 2019] is not publicly available. Thus, we report the performance (H@1, H@10, MRR) by our implementation in the right table. As can be seen, SRAP is much better.

	DBP-WD	DBP-YG	EN-FR	EN-DE
VR-GCN	19.4 55.5 0.32	33.0 68.7 0.45	26.9 63.9 0.38	40.6 73.1 0.52
SRAP	**40.7 71.2 0.51**	**40.2 72.0 0.51**	**35.5 67.9 0.46**	**50.1 75.6 0.59**

To Reviewer#2 Q1. *The search cost still takes tens of hours. The search algorithm is conducted independently on each data set.* **Reply:** As in Tab. 6, the searching cost is already comparable with fine-tuning and SRAP already performs best. This is the best case that NAS can do [1,28,55]. Considering data similarity is a promising extension direction.

Q2. *It is not sure whether the search problem will be influenced when the paths change.* **Reply:** No. Paths just serve as the input to the search problem but will not change the problem definition.

Minor. *(i) natural policy gradient is costly.* **Re:** Unlike the single-level optimization [2,32], NG is used in the bi-level problem Eq.(2) and is cheaper compared with updating embeddings F^*. *(ii) It's better to include Path-ranking as a reference.* **Re:** Thanks for the suggestion to provide a good reference to support our motivation. *(iii) why not sample the path and train the model simultaneously?* **Re:** The sampling process is expensive [18,19] and the model already performs well when the paths are fixed. *(iv) If the graph evolves, can this algorithm adapt to the new one?* **Re:** Same as the baselines, SRAP is on the static KG. Adapting to evolving KG is a future direction. We will add above discussions.

To Reviewer#3 Q1. *It seems that the model does not do well in terms of efficiency. How does it scale to large datasets?* **Reply:** In Tab. 1, SRAP has the same complexity as path-based models [11,19,27]. In Tab. 6, we show the search cost is comparable with the fine-tune cost. These evidences show that efficiency is not an issue. We will add results on YAGO3-10 dataset [Mahdisoltani et al. 2015] (41.4 66.6 0.50 for SRAP in Tab. 5 within 100h) with millions of triplets.

Minor. *(i) What does the sentence "based on the relational paths ... individually" mean?* **Re:** It means that the relational paths can be the input of triplet-based models and these models work on each triplet in the paths separately. We will clarify these statements to be more precise. *(ii) CompGCN and AutoSF should be added.* **Re:** We will add the two works for comparison and discussion. The scalability of CompGCN is a problem due to the complexity $O(|\mathcal{S}|d)$, $|\mathcal{S}|$ is triplet number. AutoSF is a triplet-based model, which can be combined with SRAP to form a much larger search space. A combination of the searched model of AutoSF and SRAP on WN18RR gives 44.9, 55.8, 0.49 in Tab. 5, which can further improve upon SRAP. More results will be elaborated in the final version.

To Reviewer#4 Q1. *The trade-off of computational complexity and performance has been explored a lot, e.g. arxiv-1909.10815. The NAS algorithm in this paper is not very novel actually.* **Reply:** This paper Balanced-NAO [Luo et. al.] does not weaken the novelty of our hybrid-search algorithm. It supports us for the problem of one-shot methods. They give different resources to different models. The similar idea has been previously explored by Hyperband [Li et al. JMLR 2018]. Instead, we use micro/macro algorithms for different parameters in the search space. The two approaches are orthogonal and can be combined together. We will add more references for discussion.

Q2. *What is the difference between S1 and S2 in Section 4.2? Why do p2 and p3 behave more differently on S1 and behave the same on S2?* **Reply:** Indeed, the difference in AUC-PR score is small. We carefully check details in S1 and S2, and find that the queries in S1 has two support rules $neighbor \wedge locatedin \rightarrow locatedin$ and $locatedin \wedge locatedin \rightarrow locatedin$, whereas S2 only has the first one. P2 models on two triplets, and thus learns well in both S1 and S2. But for P3, the additional rule in S1 gives more paths and S3 may overfit on the training data. This slightly weakens the performance on S1 than S2. We will include this discussion in the final version.

Q3. *Is it fair to compare DARTS on training loss with the algorithm optimized based on validation metric?* **Reply:** Please note that optimizing architectures on the training set is also an alternative practice in the NAS literature, see SNAS [Xie et al. ICLR2019], Auto-DeepLab [Liu et al. CVPR2019], MiLeNAS [He et al. CVPR2020].

Q4. *If the article claims that the proposed NAS algorithm is sufficiently innovative, it's necessary to be compared with more benchmarks elaborately.* **Reply:** As stated in Q1, the algorithm is novel. Our main target is not to propose an universal NAS algorithm, but one can explore domain-information in KG well. Main baselines are compared with those in Tab. 4&5, which are benchmarks in the KG literature [19,41,48].