We thank all reviewers for the valuable comments.

1. To Reviewer #2

Sufficient to explore the entire search space? We note that even keeping the inactive paths does not indicate that the exploration of search space is extended. When the sparsity constraint is not enforced, the search process just optimizes a larger super-net that incurs inconsistency with the finally derived architecture, instead of exploring a larger search space. The proportion of explored architectures is decided by how many architectures are covered by the search process. In order to analyze quantitatively, we run the search of DARTS and our ISTA-NAS on CIFAR-10 for three times, with 50 epochs for each run. We count how many different architectures are covered in the 50 epochs. For DARTS, we need to perform the architecture derivation (filtering inactive paths) for each epoch to get its architecture. We find that the average number of architectures for the three-times experiments of DARTS is 35.7, while the number of ISAT-NAS is 34.3. This shows that there is not a significant difference. We hope this result could resolve your concern.

The results on ImageNet. As shown in Tables 3 and 4, ISTA-NAS has higher accuracies than PC-DARTS on both CIFAR-10 and ImageNet with less search cost. The superiority of accuracy on ImageNet is not so significant, maybe because that PC-DARTS has been close to the upper limitation of performance on ImageNet in the DARTS-based space.

Relations to AtomNAS and MTL-NAS. Thanks for reminding us. We will cite and discuss the relations to AtomNAS (ICLR 2020) and MTL-NAS (CVPR 2020) in the revised version.

2. To Reviewer #3

Motivation. We think that the decoupling method that you mentioned refers to sampling based methods, such as SPOS. We need to point out that our method belongs to differentiable NAS methods, instead of sampling-based ones that could have no architecture parameter (e.g. by the uniform sampling in SPOS). In the scope of differentiable NAS, without enforcing sparsity, the super-net will be dense and incur inconsistency and redundant search cost. Please refer to lines 32 to 40 for our motivations. Besides, GDAS that minimizes the entropy of edges has similar motivations as ours.

Kendall calculation. Note that the methods of the chain-based space, such as SPOS, need to re-calculate BN layers for measuring the accuracy of the sub-network extracted from a super-net. But our method belongs to the cell-based search space. We just measure the correlation between the accuracies of the super-net after search and the target-net after retraining, so do not have the BN problem. The explanation of Table 2 is shown in lines 246 to 255 of our paper.

Comparison with GDAS and One-shot. We compare with One-shot (Bender, et al. ICML 2018) and GDAS. On CIFAR-10, GDAS has an error rate of 2.93% with 0.21 GPU-day, while ISTA-NAS has an error rate of 2.54% with 0.05 GPU day. On ImageNet, the top-1 accuracies of One-shot, GDAS, and ISTA-NAS are 75.2%, 74.0%, and 76%, respectively. Besides, GDAS did not have a one-stage method and direct search on ImageNet in their paper.

Discussion of Once for All (OFA). OFA searches in the chain-based space so its result could not be directly compared with ours of the cell-based space. We will cite and discuss it in detail in the revised version.

3. To Reviewer #6

Clear details. Thanks for your suggestions. The arrows in Eq. (8) just denote how x_j is computed in the network $\mathcal{N}(W, Z)$ and how to get $\mathcal{N}(W, B)$. We will make the description clearer in the revised version. As shown in "Input" of Algorithm 1, $\mathbf{A}_j$ ($1 < j \leq n$) are sampled as fixed matrices. They construct the relations between the original space and the compressed space, and are not learnable in Eq. (9). The proof of Proposition 1 does not unfold some easily-derived steps due to the limited page space. We will include the complete proof in the revised version.

Implementations. The implementation details are described in the supplementary material. The code is based on PyTorch. But the sparse coding part is achieved on CPU by a Python interface of CVX using the MOSEK solver. So the whole framework is still differentiable for training. Code address will be released.

4. To Reviewer #7

Structure. Thanks for your suggestion. We start with the two-stage method because it offers the basis of the one-stage method. The one-stage method enjoys better consistency but performs search directly in the evaluation settings, which consumes more time in general. So we keep both versions. We will consider your suggestion and refine the structure.

Chain-based space. We develop our method for the cell-based space because it has a higher dimension for intermediate nodes, so suffers more than the chain-based space that mainly searches for the configurations of MBConv layers. But it is a good suggestion to extend our method to chain-based space. Following the space of ProxylessNAS on ImageNet, we get an architecture with 75.8% top-1 accuracy, which is a little better than ProxylessNAS (75.1%). It has less FLOPs (410M) than our ISTA-NAS (cell-based) maybe due to the design of chain-based space. We think the result will be better with more hyper-parameter tunning or if it can be combined with more advanced space, such as Once for all.

Hyper-parameter. As shown in Figure 2 (right), the difference of $\mathbf{z}$ in neighboring epochs decays fast. Even without the termination condition, we observe that once the value is smaller than 1e-3, it will stay close to 0 and does not change the architecture. So ϵ needs to be small, but no matter ϵ is 1e-3 or 1e-4, it has little effect on the search results.

Discussion of Once for All. Thanks for reminding us. We will cite and discuss it in detail in the revised version.