We sincerely thank all reviewers for their constructive comments. The primary goal of this paper is to analyze the effect of parameter sharing on channel number search and to provide a tangible and controllable measurement for parameter sharing. We hope this would shed some light on a better understanding of parameter sharing in NAS.

.... TO REVIEWER # 1 ....

Q1: Not particularly strong empirical results.
We sincerely appreciate your recognition of our technical contributions. For the weak empirical results, one potential reason could be the coarse search space. For CIFAR10 experiments, we use the same candidate width for all layers (Line 181). Unlike previous works [2, 8] that doubles the candidate width at each pooling layer, our search space is free from domain expertise, and can better challenge the searching ability of the algorithm. For ImageNet experiments, a more fine-grained search space may similarly lead to better results with improved implementation, see **Q3** of **R3** for details. Meanwhile, as you pointed out, different optimization of APS would be interesting to explore in the future.

Q2: Writing issues. Thanks, and we will fix these typos. In Equation 2, we use mode-d multiplication rather than matrix multiplication for 4-D convolutional kernels. Details of mode-d multiplication can be found in [11].

.... TO REVIEWER # 2 ....

Q1: How can the proposed method improve one-shot NAS.
APS-T can be readily extended to one-shot NAS. We may first train weights and anneal Φ simultaneously with uniformly sampled architectures, and then update the RL controller based on the decoupled candidate parameters. We need to highlight that our analysis on parameter sharing is fundamental in the training process of NAS, and is agnostic to one-shot NAS or traditional NAS. We will add a paragraph of explanation in the revision.

Q2: More description on RL controller and parameter update.
We largely follow ENAS [17] in controller design. A two-layer LSTM with 100 hidden units is used, and the width decisions are made auto-regressively. The RL states contain the previous layer width decision, the one-hot layer index encoding, and available FLOPs left. We will detail these as well as the parameter update in the revision, thanks.

Q3: Illustrating the calculation of Φ.
Calculation of Φ can be found in Equation 9 of Appendix B. We will move it to Section 3.4 for better readability.

Q4: The CIFAR-10 dataset needs 600 epochs, will the calculation amount explode for large structure space?
On CIFAR-10, we use 600 epochs in order to be consistent with TAS [2]. In fact, fewer epochs yield similar performance. On ImageNet we take the common 160 epochs. The searching time is 6.9 hours for ResNet-20 (Line 177), 24 hours for ResNet-18 and 48 hours for MobileNet-v2 (Line 189), all of which are acceptable.

Q5: How to measure the discrimination of architecture; some clarity issues.
The architecture discrimination is measured by the probability gap of different candidates, as outlined in Line 205 and visualized in Figure 4. We will explain Figure 1, 2 in more details in revision. Please refer to [11] for $\times_1$ and $\times_2$.

.... TO REVIEWER # 3 ....

Q1: Detailed discussion and comparison to MetaPruning.
Thanks for the suggestion. MetaPruning uses a meta-network to provide more flexible patterns of parameter sharing. However, the sharing scheme is less interpretable and controllable. Our semi-orthogonal projections enable quantitative measurement and explicit control of parameter sharing, which helps to better understand its role in architecture search. A more detailed discussion will be incorporated in the revised version.

Q2: How about directly optimizing $\mathcal{P}$ and $\mathcal{Q}$ with the task loss instead of Equation 4?
Optimizing $\mathcal{P}$, $\mathcal{Q}$ with task loss leads to much lower validation acc than our approach ($\sim 70\%$ v.s. $\sim 90\%$), and the architecture with maximum capacity in Line 228 cannot be safely found. We suspect it due to the break of orthogonality constraints inside $\mathcal{P}$ $\mathcal{Q}$, leading to linearly correlated filters within the single kernel and thus decreases model capacity.

Q3: I doubt about the efficiency of scaling up.
The number of parameters in $\mathcal{P}$, $\mathcal{Q}$ grows linearly with the number of candidates, which hinders more fine-grained search space on ImageNet that could lead to better results. The problem can be alleviated by better implementation: we can compute Φ and update $\mathcal{P}$ and $\mathcal{Q}$ layer-wisely rather than feeding all $\mathcal{P}$ and $\mathcal{Q}$ into the memory. In this way, the space complexity can be effectively reduced, allowing for more fine-grained search space. To reduce the computational overhead of Φ, we can alternatively update $\mathcal{P}$ and $\mathcal{Q}$ less frequently, as outlined in Line 414, Appendix C.

.... TO REVIEWER # 4 ....

Q1: Φ depends on both meta-weights and $\mathcal{P}$, $\mathcal{Q}$, thus simply optimizing $\mathcal{P}$, $\mathcal{Q}$ does not ensure decreasing Φ.
As proposed in Definition 3.1, weights are assumed to follow the standard normal distribution. Thus Φ only depends on $\mathcal{P}$ and $\mathcal{Q}$, as shown in Equation 9 of Appendix B. Consequently, updating P and Q is sufficient to minimize Φ, which is also empirically observed in the rightmost sub-figure in Figure 3 and 4.

Q2: All candidates are fully-shared with meta-weights.
There can be no parameter sharing among different candidates when vectors in $\boldsymbol{P}_1$ $\boldsymbol{Q}_1$ are orthogonal to vectors of $\boldsymbol{P}_2$ $\boldsymbol{Q}_2$. The right part of Figure 1(b) and 1(c) gives an example of candidate weights being non-overlapping sub-tensors of meta-weights when $\mathcal{P}$ $\mathcal{Q}$ are composed of disjoint standard basis. We use linear correlation to depict the sharing level Φ. According to Theorem 3.1, the maximum and minimum of Φ corresponds to maximally overlapped sub-tensors (APS-O) and non-overlapping sub-tensors (APS-I) respectively, which verifies that the definition is proper and legible.