We thanks all the reviewers very much for the constructive comments. Below please find our response. We hope you could raise your evaluation if you find that we address your concerns.

General Response: Reviewers ask why the practical gain of the local imitation method is not that significant as the theory. Here we give the response.

(i) Local imitation minimizes the surrogate local discrepancy loss while global imitation directly minimizes the original discrepancy loss. The surrogate loss is able to upper bound the original loss up to a constant but this constant still matters in practice. As a result, despite the local imitation has faster rate, both method can not consistently outperform the other one; This justifies the usefulness of comparing and choosing the better method.

(ii) Techniques such as [1] can be easily applied to our framework to reduce the constant gap between the two losses. However, in this paper, we aim at giving a simple algorithm which achieves SOTA both empirically and theoretically and also has future potential. We leave the further algorithm engineering for future work.

(iii) Our way of dealing with the Lipschitz constant is standard and tight. It is unavoidable to have such gap between theory and practice given the complexity of deep learning. But the mathematical structure we found gives stronger justification, deeper understanding of the pruning paradigm, and motivates the new algorithm, which we demonstrate useful in practice.

Reviewer 1: For the ImageNet experiments, the overhead of pruning phase is generally about 0.2x of that of finetuning phase. We will add more discussion on the practical computational cost of the algorithm as well as more discussion on knowledge distillation and NAS in the next version.

Reviewer 2: **(second paragraph in 'correctness')** Optimizing γ (rather than fixing it) is a core component to achieve faster convergence rate. However, using global reconstruction loss, does not necessarily weaken the exponential rate, if we still optimize γ. The proof framework of Theorem 4 can be extended to this case (optimizing γ and minimize global reconstruction loss) and gives exponential decay rate. We will add more discussion on that. Besides, Taylor approximation is not used in local imitation as each iteration is already fast (line 102-107) and thus does not effect the exponential rate. **(a, h)** We will move this statement to main text. **(b)** Lemma 1-3 are self contained in terms of notation. We guess that you find cl$M$ and ri$M$ undefined? Their definition is in the beginning of Appendix (the def of h and $\bar{h}$ are also there). The main intuition is that local imitation actually enjoys good geometric property (line 551), which makes imitating the internal layer's output very efficient (see key inequality between line 555 and 556). **(c)** Please see general response. **(d)** Yes, it is equivalent in practice. The boldsymbol denotes vector. We will improve the clarity. **(e)** Sorry for the typo. $U_i = [-a_i(k)/(1-a_i(k)), 1]$. **(f)** The time and space complexity is small. For local imitation, we only add one extra parameter for each neuron and this parameter will be merged into the scale of neuron after the pruning finish. Following in line 102-107 and sec 5.1 in appendix, executing the pruning algorithm only requires one forward pass and the selection can be done with simple matrix multiplication using O(batch size $\times$ num of channels) space complexity. The Taylor approximation is only applied to global imitation, which only adds 2 extra parameters for each neuron. The main time complexity for selecting one neuron is calculating the gradient of the ancillary variable, which is also small. See sec 5.4 for more details. **(g, i)** Yes, b is the ancillary parameters. But a in the appendix is identical to the a used in the main text (e.g. between line 61 and 62). And a is not the scale on the activation but w_1 is (see line 53). In practice, we regard the weight, activation and batchnorm together as a neuron. **(j)** The bound in Theorem 1 can not be adapted to Ye at al. (2020) and their bound is tight. One key difference is that we optimizing γ during selection, which significantly enlarge the search space of each greedy step and thus improves the rate. **(k)** They are very different. Our method is based on forward selection (starting from empty network and greedily add neurons). Taylor approximation is only a speed-up technique for us. In comparison, Molchanov (2017b) eliminates neurons from full network by looking at neuron importance measured by Taylor expansion.

Reviewer 3: **(1)** Our theoretical improvement over Ye et al. (2020) is discussed in introduction and Table 1. In terms of algorithm, intuitively, our method weights each selected neurons differently instead of simply average them like Ye et al. (2020), which brings the improvement. We will give more detailed discussion on difference between Ye et al. (2020) and other existing papers. **(2)** There are two reasons. Firstly, the final comparison is made after finetuning and thus part of the improvement on pruning phase might be smaller after finetuning. For the second point, please see general response.

Reviewer 4: Please see general response (iii) on the 'Lipschitz continuous' comment. We will add more ablation studies similar to that in sec 2.3 and sec 3.1 to compare with GFS.

Reference: [1] Zhuang, Zhuangwei, et al. "Discrimination-aware channel pruning for deep neural networks." Advances in Neural Information Processing Systems. 2018.