We sincerely thank the anonymous reviewers for their support and constructive comments.

To Reviewer #1. Q1: *Presentation.* **A**: Treatment group denotes the features to be researched while control group is the features for controlling irrelevant factors, which are instantiated as real and knockoff features respectively (Line 81-90 in the main paper). We will refine the presentation and add a dedicated section for revisiting related-works.

Q2: *Different results in the baseline.* **A**: We collect the baseline results from the published papers and their test errors of the 'original' networks are slightly different. We re-implement the competing methods under the same baseline, and the results of ResNet-56 on CIFAR-10 with mean/std are shown below. More results will be included in the final version.

Method	Original Error (%)	Pruned Error (%)	Gap (%)	Params. ↓ (%)	FLOPs ↓ (%)
GAL (2019) [17]	6.30 ± 0.24	6.91 ± 0.14	0.61	42.4	50.0
SCP (Ours)	6.30 ± 0.24	**6.36** ± 0.09	**0.06**	**56.3**	**56.0**

Q3: *Effectiveness of knockoffs.* **A**: Results of ResNet-50 on ImageNet using noise or knockoffs are shown below. Since ImageNet is more complex, *e.g.*, 1,000 categories and images of high resolution (224×224), utilizing noise data is hard to obtain good results. More results will be included in the final version.

Method	Error (%)	Gap (%)	FLOPs ↓ (%)
No control	26.22	2.37	54.6
Noise	25.86	2.01	54.6
Ours with bias	24.74	0.89	54.6

Q4: *The term 'filter'.* **A**: There are M filters in Eq. (1), and each of them is an $N \times k \times k$ tensor, where $k \times k$ is the kernel size (*e.g.*, 3×3) and N is the number of input channels. We will refine the presentation around the definitions.

Q5: *Efficiency.* **A**: Learning based approaches (*e.g.*, CP [8], GAL [17]) are compared in Table 1 and 2 in the main paper. These methods re-train the original network to learn the importance of filters. Compared with them, our method fixes the network weights and only tunes the control scales when discovering redundant filters, which is more efficient. The practical consuming time of pruning ResNet-56 (no fine-tuning) on CIFAR-10 is shown below (A V100 GPU).

Method	CP	GAP	Ours
Time (min)	46	25	16

To Reviewer #2. Q1: *Clarity issues.* **A**: 1) The term 'scales' denotes the magnitudes of $\boldsymbol{\beta}_l$ and $\tilde{\boldsymbol{\beta}}_l$, which will be replaced with 'scaling factors' and defined at the beginning of the paper. 2) We adopt similar loss function for the discriminator as Knockoff-gan [9] and propose to aggregate multiple elements for efficiently generating knockoff data, which will be fully described in the final version. The novel contributions will also be discussed at the end of introduction as your suggestion.

Q2: *Pruning procedure in practice.* **A**: Unimportant filters in each layer will be pruned. As discussed in the main body (Line 184-186), for an arbitrary convolutional layer, filters with small $(\beta_j^l - \tilde{\beta}_j^l)$ will be recognized as redundancy.

Q3: *Concept of knockoff data.* **A**: Specifically, knockoff data do not contain real objects of any category (*e.g.*, goldfish, snail) in the real dataset, as shown in Figure 3 of the main body. We will add more explanations in the final version.

Q4: *Limitation.* **A**: Thanks for this nice concern. 1) Control scales are distributed in range [0,1] as shown in Figure R1. 2) It does not matter when scales are close to 0.5/0.5, since we focus on sorting scales of different filters, rather than $\boldsymbol{\beta}^l$ and $\tilde{\boldsymbol{\beta}}^l$ of the same filter.

To Reviewer #3. Q1: *Figure 1.* **A**: We will replace the art picture in Fig. 1 with the actually generated knockoff data.

Q2: *Example for swapping operation.* **A**: Suppose that $\mathcal{A}$=[0.1,0.18,-0.1],$\tilde{\mathcal{A}}$=[0.13,0.16,-0.15], and then $[\mathcal{A}, \tilde{\mathcal{A}}]$=[0.1,0.18,-0.1,0.13,0.16,-0.15]. If $\tilde{S} = \{2\}$, the swapped feature $[\mathcal{A}, \tilde{\mathcal{A}}]_{swap(\tilde{S})}$=[0.1,0.16,-0.1,0.13,0.18,-0.15].

To Reviewer #4. Q1: *Optimization procedure.* **A**: Eq. (6) is the general formulation for feature selection. In practice, we use Eq. (8) to avoid ℓ_0-norm and then Adam optimizer can be applied.

Q2: *$\boldsymbol{\beta}^l$,$\tilde{\boldsymbol{\beta}}^l$ in the feature selection layer.* **A**: Thanks for this constructive comment. To have an explicit understanding, we illustrate the change of $\boldsymbol{\beta}^l$ (*i.e.*, the first conv layer in last stage of ResNet-56 on CIFAR-10) during the optimization in Figure R1. Wherein, each curve denotes the control scale ($\beta_j^l \in \boldsymbol{\beta}^l$) of a specific convolution filter. Similarly, $\tilde{\boldsymbol{\beta}}^l = \mathbf{1} - \boldsymbol{\beta}^l$ has the opposite phenomenon. Since most of existing deep neural networks are of heavy design for the accuracy reason, they will not collapse to $\boldsymbol{\beta}^l = \mathbf{1}$. Visualizations on $\boldsymbol{\beta}^l$, $\tilde{\boldsymbol{\beta}}^l$ and corresponding discussions will be included in the final version.

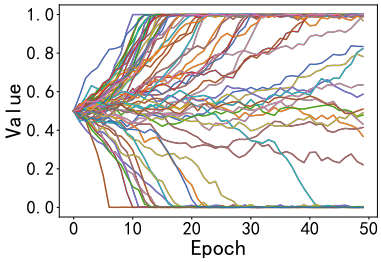

Figure R1: $\boldsymbol{\beta}^l$ *w.r.t.* epoch.

Q3: *Notions.* **A**: 1) We need to calculate knockoff feature $\tilde{\mathcal{A}}^{l+1}$, which is the $l+1$-th layer's feature map of the network with knockoff data as input, *i.e.*, $\tilde{\mathcal{A}}^{l+1} = f^{l+1}(\tilde{X}, \mathcal{W}^{1:l+1})$. 2) Biases $\boldsymbol{b}^l$ and $\tilde{\boldsymbol{b}}^l$ are the modified terms from theoretical derivation, which ensure that the knockoff condition satisfies in the forward propagation of neural network from a theoretical perspective. These notions will be clarified more detailedly in the final version.

Q4: *Different methods and backbone models.* **A**: We apply the proposed method on different backbone models to verify its generalization ability. The results of competing methods on different backbones are collected from their original papers for fair comparison. These methods have their own experimental settings and lack results on some backbones. For example, Hrank [16] did not report results on ResNet20, ResNet32 and MobileNetV2. Thus we do not include it.

Q5: *Clarity.* **A**: Thanks, all of these typos and minor comments will be carefully fixed in the final version, and more discussions on the broader impact will be included.