A. Optimization Interpretation of *LeGR*

LeGR can be interpreted as minimizing a surrogate of a derived upper bound for the loss difference between (1) the pruned-and-fine-tuned CNN and (2) the pre-trained CNN. Concretely, we would like to solve for the filter masking binary variables $\mathbf{z} \in \{0,1\}^K$, with K being the number of filters. If a filter k is pruned, the corresponding mask will be zero ($z_k = 0$), otherwise it will be one ($z_k = 1$). Thus, we have the following optimization problem:

$$\begin{aligned} \min_{\mathbf{z}} \; & \mathbb{L}(\mathbf{\Theta} \odot \mathbf{z} - \eta \sum_{j=1}^{\tau} \Delta \boldsymbol{w}^{(j)} \odot \mathbf{z}) - \mathbb{L}(\mathbf{\Theta}) \\ & \text{s.t. } C(\mathbf{z}) \leq \zeta, \end{aligned} \tag{4}$$

where $\mathbf{\Theta}$ denotes all the filters of the CNN, $\mathbb{L}(\mathbf{\Theta}) = \frac{1}{|D|}\sum_{(x,y)\in D} L(f(x|\mathbf{\Theta}), y)$ denotes the loss function of filters where x and y are the input and label, respectively. D denotes the training data, f is the CNN model and L is the loss function for prediction (*e.g.*, cross entropy loss). η denotes the learning rate, τ denotes the number of gradient steps, $\Delta \boldsymbol{w}^{(j)}$ denotes the gradient with respect to the filter weights computed at step j, and $\odot$ denotes element-wise multiplication. On the constraint side, $C(\cdot)$ is the modeling function for FLOP count and ζ is the desired FLOP count constraint. By fine-tuning, we mean updating the filter weights with stochastic gradient descent (SGD) for τ steps.

Let us assume the loss function $\mathbb{L}$ is Ω_l-Lipschitz continuous for the l-th layer of the CNN, then the following holds:

$$\begin{aligned} & \mathbb{L}(\mathbf{\Theta} \odot \mathbf{z} - \eta \sum_{j=1}^{\tau} \Delta \boldsymbol{w}^{(j)} \odot \mathbf{z}) - \mathbb{L}(\mathbf{\Theta}) \\ & \leq \mathbb{L}(\mathbf{\Theta} \odot \mathbf{z}) + \sum_{i=1}^{K} \Omega_{l(i)} \eta \left\| \sum_{j=1}^{\tau} \Delta \boldsymbol{w}_i^{(j)} \odot \mathbf{z}_i \right\| - \mathbb{L}(\mathbf{\Theta}) \\ & \leq \sum_{i=1}^{K} \Omega_{l(i)} \|\mathbf{\Theta}_i\| \mathbf{h}_i + \sum_{i=1}^{K} \Omega_{l(i)}^2 \eta \tau \mathbf{z}_i \\ & = \sum_{i=1}^{K} (\Omega_{l(i)} \|\mathbf{\Theta}_i\| - \Omega_{l(i)}^2 \eta \tau) \mathbf{h}_i + \Omega_{l(i)}^2 \eta \tau, \end{aligned} \tag{5}$$

where $l(i)$ is the layer index for the i-th filter, $\mathbf{h} = \mathbf{1} - \mathbf{z}$, and $\|\cdot\|$ denotes ℓ_2 norms.

On the constraint side of equation (4), let $R_{l(i)}$ be the FLOP count of layer $l(i)$ where filter i resides. Analytically, the FLOP count of a layer depends linearly on the number of filters in its preceding layer:

$$R_{l(i)} = u_{l(i)} \left\| \{\mathbf{z} : \mathbf{z}_j \; \forall j \in P(l(i)\} \right\|_0, \; u_{l(i)} \geq 0, \tag{6}$$

where $P(l(i))$ returns a set of filter indices for the layer that precedes layer $l(i)$ and $u_{l(i)}$ is a layer-dependent positive constant. Let $\hat{R}_{l(i)}$ denote the FLOP count for layer $l(i)$ for the pre-trained network ($\mathbf{z} = \mathbf{1}$), one can see from equation (6) that $R_{l(i)} \leq \hat{R}_{l(i)} \; \forall i, \mathbf{z}$. Thus, the following holds:

$$C(\mathbf{1} - \mathbf{h}) = \sum_{i}^{K} R_{l(i)}(1 - \mathbf{h}_i) \leq \sum_{i}^{K} \hat{R}_{l(i)}(1 - \mathbf{h}_i). \tag{7}$$

Based on equations (5) and (7), instead of minimizing equation (4), we minimize its upper bound in a Lagrangian form. That is,

$$\min_{\mathbf{h}} \; \sum_{i=1}^{K} \left(\alpha_{l(i)} \|\mathbf{\Theta}_i\| + \kappa_{l(i)}\right) \mathbf{h}_i, \tag{8}$$

where $\alpha_{l(i)} = \Omega_{l(i)}$ and $\kappa_{l(i)} = \eta\tau\Omega_{l(i)}^2 - \lambda \hat{R}_{l(i)}$. To guarantee the solution will satisfy the constraint, we rank all filters by their scores $s_i = \alpha_{l(i)} \|\mathbf{\Theta}_i\| + \kappa_{l(i)} \; \forall \, i$ and threshold out the bottom ranked (small in scores) filters such that the constraint $C(\mathbf{1} - \mathbf{h}) \leq \zeta$ is satisfied and $\|\mathbf{h}\|_0$ is maximized. That is, *LeGR* can be viewed as learning to estimate $\boldsymbol{\alpha}$ and $\boldsymbol{\kappa}$ by assuming that better estimates of $\boldsymbol{\alpha}$-$\boldsymbol{\kappa}$ produce a better solution for the original objective (4) by solving the surrogate of the upper bound (8).

B. LeGR-DDPG

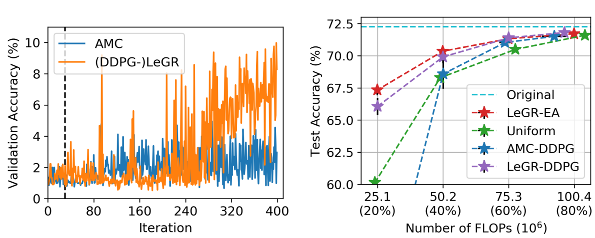
We have also tried learning the layer-wise affine transformations with actor-critic policy gradient (DDPG), which is adopted in prior art [20]. We use DDPG in a sequential fashion that follows [20]. *LeGR* requires two continuous actions (*i.e.*, α_l and κ_l) for layer l while *AMC* needs only one action (*i.e.*, percentage). We conduct the comparison of pruning ResNet-56 to 50% of its original FLOP count targeting CIFAR-100 with $\hat{\tau} = 0$ and hyper-parameters following [20]. As shown in Fig. 9a, while both *LeGR* and *AMC* outperform random search (iterations before the vertical black-dotted line), *LeGR* converges faster to a better solution. Beyond comparing the progress of searching, we also compare the performance of the final pruned networks. As shown in Fig. 9b, searching layer-wise affine transformations is more efficient and effective compared to searching the layer-wise filter percentages. Comparing *LeGR* using the two policy improvement methods, we empirically find that DDPG incurs larger variance on the final network than evolutionary algorithm.

C. ImageNet Result Detail

The comparison of *LeGR* with prior art on ImageNet is detailed in Table 2.

Table 2: Summary of pruning on ImageNet. The sections are defined based on the FLOP count left. The accuracy is represented in the format of *pre-trained* ↦ *pruned-and-fine-tuned*.

Network	Method	Top-1	Top-1 Diff	Top-5	Top-5 Diff	FLOP count (%)
ResNet-50	NISP [64]	- → -	-0.2	- → -	-	73
	LeGR	76.1 → **76.2**	**+0.1**	92.9 → **93.0**	**+0.1**	73
	SSS [28]	76.1 → 74.2	-1.9	92.9 → 91.9	-1.0	69
	ThiNet [40]	72.9 → 72.0	-0.9	91.1 → 90.7	-0.4	63
	C-SGD-70 [13]	75.3 → 75.3	+0.0	92.6 → 92.5	-0.1	63
	Variational [66]	75.1 → 75.2	+0.1	92.8 → 92.1	-0.7	60
	GDP [34]	75.1 → 72.6	-2.5	92.3 → 91.1	-1.2	58
	SFP [19]	76.2 → 74.6	-1.6	92.9 → 92.1	-0.8	58
	FPGM [21]	76.2 → 75.6	-0.6	92.9 → 92.6	-0.3	58
	LeGR	76.1 → **75.7**	-0.4	92.9 → **92.7**	-0.2	58
	GAL-0.5 [35]	76.2 → 72.0	-4.2	92.9 → 91.8	-1.1	57
	AOFP-C1 [14]	75.3 → 75.6	**+0.3**	92.6 → **92.7**	**+0.1**	57
	NISP [64]	- → -	-0.9	- → -	-	56
	Taylor-FO-BN [42]	76.2 → 74.5	-1.7	- → -	-	55
	CP [23]	- → -	-	92.2 → 90.8	-1.4	50
	SPP [59]	- → -	-	91.2 → 90.4	-0.8	50
	LeGR	76.1 → **75.3**	**-0.8**	92.9 → **92.4**	-0.5	47
	CCP-AC [46]	76.2 → **75.3**	-0.9	92.9 → 92.6	**-0.3**	44
	RRBP [70]	76.1 → 73.0	-3.0	92.9 → 91.0	-1.9	45
	C-SGD-50 [13]	75.3 → 74.5	**-0.8**	92.6 → 92.1	**-0.5**	45
	DCP [72]	76.0 → 74.9	-1.1	92.9 → 92.3	-0.6	44
MobileNetV2	AMC [20]	71.8 → 70.8	-1.0	→ -	-	70
	LeGR	71.8 → **71.4**	**-0.4**	→ **-**	**-**	70
	LeGR	71.8 → 70.8	-1.0	→ **-**	**-**	60
	DCP [72]	70.1 → 64.2	-5.9	→ -	-	55
	MetaPruning [37]	72.7 → 68.2	-4.5	→ -	-	50
	LeGR	71.8 → **69.4**	**-2.4**	→ **-**	**-**	50



(a) (b)

Figure 9: Comparison between searching the layer-wise filter norms and searching the layer-wise filter percentage. (a) compares the searching progress for 50% FLOP count ResNet-56 and (b) compares the final performance for ResNet-56 with various constraint levels.