A. ImageNet Training Configuration

For training PeLK, we use 8 GPUs and a total batch size of 4096 to train for 300 epochs. The optimizer is AdamW [4] with momentum of 0.9 and weight decay of 0.05. The learning rate setting includes an initial value of 4×10^{-3}, cosine annealing and 20-epoch warm-up. For the data augmentation and regularization, we use RandAugment [1] ("rand-m9-mstd0.5-inc1" as implemented by timm [9]), label smoothing coefficient of 0.1, mixup [11] with $\alpha = 0.8$, CutMix [10] with $\alpha = 1.0$, Rand Erasing [12] with probability of 25% and Stochastic Depth with a drop-path rate of 10%/40%/50% for PeLK-T/S/B respectively.

B. ERF Quantitation Comparison

Following RepLKNet [2] and SLaK [3], we report the high-contribution area ratio r to give a quantitation analysis of ERF comparison in Table 1. Here, r denotes the proportion of the smallest rectangle to the overall input area that can encompass the contribution scores above a specified threshold t. For instance, given an area of $\mathrm{R} \times \mathrm{R}$ at the center can cover $\mathrm{t} = 20\%$ contribution scores of a 1024×1024 input, the corresponding area ratio of t = 20% is $r = (\mathrm{R}/1024)^2$. Larger r indicates a smoother distribution of high-contribution pixels. Compared with previous CNN paradigms, our PeLK naturally takes a larger range of pixels into accout to make decisions, which continues to demonstrate the intuitive effect of the extremely large kernel on enlarging the receptive field.

Models	Kernel Size	t=20%	t=30%	t=50%
ResNet	3-3-3-3	1.1%	1.8%	3.9%
ConvNeXt	7-7-7-7	2.0%	3.6%	7.7%
RepLKNet	31-29-27-13	4.0%	9.1%	19.1%
SLaK	51-49-47-13	6.9%	11.5%	23.4%
PeLK	51-49-47-13	**7.5%**	**12.8%**	**25.9%**
PeLK-101	101-69-67-13	**8.1%**	**13.7%**	**26.5%**

Table 1. **Quantitative comparison of ERF.** We use ResNet-152 and tiny size model for the other methods. larger values indicate larger ERFs and smoother distribution of high-contribution pixels.

C. Ablation on Re-parameterization

According to RepLKNet [2], directly optimizing large kernel convnets can be difficult and leads to performance degradation. Therefore, existing large kernel paradigms re-parameterize a small kernel (e.g., 5×5) to alleviate this issue. In this part, we remove the re-parameterization trick to see how degraded the models are. We train tiny model for 120 epochs on ImageNet as the same in Section 3. As shown in Table 2, our peripheral convolution still sustain a good performance after removing the small convolution, while the dense convolution suffers a significant degradation. This phenomenon implies that our peripheral convolution can alleviate the optimization difficulty of large dense convolution by reducing the number of parameters required.

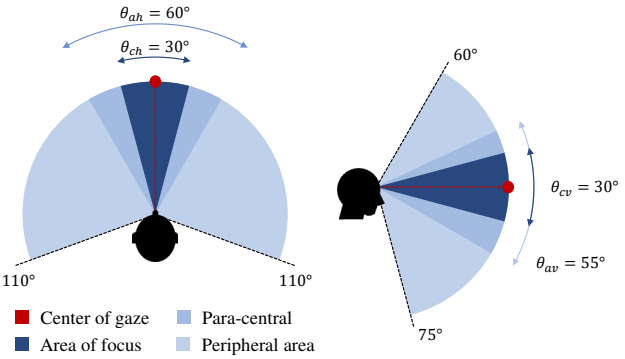

Figure 1. **Illustration of peripheral vision.** Human vision possesses distinct clarity within a confined focus area, contrasted by merely vague perception in the extensive peripheral area. Note that all numbers are approximate values.

Models	Conv Form	w/ Rep	w/o Rep	Δ
RepLK	dense	81.6	80.2	-1.4
PeLK	peripheral	81.6	80.9	**-0.7**

Table 2. **Ablation on Re-parameterization.** We compare the degradation of the model after removing the rep technique.

D. Peripheral Vision

The human visual field showcases a phenomenon known as "central focus and peripheral blur" [6, 7]. According to vision science literature[5, 8], the human visual field can be modeled as fan-shaped figures as shown in Fig 1. It consists of three segments from the center to the periphery: central area (θ_{ch} and θ_{cv}), para-central area (θ_{ah} and θ_{av}) and peripheral area (θ_{ph} and θ_{pv}). The central area is the primary part used for clear perception. Therefore, the proportion of the focused area in the human visual system can be calculated as:

$$P_{human} = \frac{\pi\theta_{ch} \cdot \theta_{cv}}{\pi\theta_{ph} \cdot \theta_{pv}} = 2.72\% \tag{1}$$

Similarly, our peripheral convolution keeps fine-grained parameters in the center, the central proportion for PeLK is:

$$P_{pelk} = \frac{5 \times 5}{51 \times 51} = 0.96\% \tag{2}$$

Although the values for PeLK and human are not strictly equivalent, they both are very small ratios ($< 5\%$), indicating that an efficient visual mechanism only requires a very small proportion of fine-grained perception.

References

[1] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020. 1

[2] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11963–11975, 2022. 1

[3] Shiwei Liu, Tianlong Chen, Xiaohan Chen, Xuxi Chen, Qiao Xiao, Boqian Wu, Mykola Pechenizkiy, Decebal Mocanu, and Zhangyang Wang. More convnets in the 2020s: Scaling up kernels beyond 51x51 using sparsity. *arXiv preprint arXiv:2207.03620*, 2022. 1

[4] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1

[5] Elaine Nicpon Marieb and Katja Hoehn. *Human anatomy & physiology*. Pearson education, 2007. 1

[6] Juhong Min, Yucheng Zhao, Chong Luo, and Minsu Cho. Peripheral vision transformer. *Advances in Neural Information Processing Systems*, 35:32097–32111, 2022. 1

[7] RT Pramod, Harish Katti, and SP Arun. Human peripheral blur is optimal for object recognition. *Vision research*, 200: 108083, 2022. 1

[8] Hans Strasburger, Ingo Rentschler, and Martin Jüttner. Peripheral vision and pattern recognition: A review. *Journal of vision*, 11(5):13–13, 2011. 1

[9] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. 1

[10] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019. 1

[11] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 1

[12] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, pages 13001–13008, 2020. 1