This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

UniRepLKNet: A Universal Perception Large-Kernel ConvNet for Audio, Video, Point Cloud, Time-Series and Image Recognition

Xiaohan Ding[1*] Yiyuan Zhang[2*] Yixiao Ge[1]
Sijie Zhao[1] Lin Song[1] Xiangyu Yue[2] Ying Shan[1]
[1] Tencent AI Lab [2] The Chinese University of Hong Kong
https://github.com/AILab-CVC/UniRepLKNet

Abstract

Large-kernel convolutional neural networks (ConvNets) have recently received extensive research attention, but two unresolved and critical issues demand further investigation. 1) The architectures of existing large-kernel ConvNets largely follow the design principles of conventional ConvNets or transformers, while the architectural design for large-kernel ConvNets remains under-addressed. 2) As transformers have dominated multiple modalities, it remains to be investigated whether ConvNets also have a strong universal perception ability in domains beyond vision. In this paper, we contribute from two aspects. 1) We propose four architectural guidelines for designing large-kernel ConvNets, the core of which is to exploit the essential characteristics of large kernels that distinguish them from small kernels - they can see wide without going deep. Following such guidelines, our proposed large-kernel ConvNet shows leading performance in image recognition (ImageNet accuracy of 88.0%, ADE20K mIoU of 55.6%, and COCO box AP of 56.4%), demonstrating better performance and higher speed than the recent powerful competitors. 2) We discover large kernels are the key to unlocking the exceptional performance of ConvNets in domains where they were originally not proficient. With certain modality-related preprocessing approaches, the proposed model achieves state-of-the-art performance on time-series forecasting and audio recognition tasks even without modality-specific customization to the architecture. All the code and models are publicly available on GitHub and Huggingface.

1. Introduction

The design paradigm of convolutional neural networks (ConvNets) with very large kernels originated from RepLKNet [19] when the status of ConvNets was challenged by Vision Transformers (ViTs) [20, 22, 50, 74, 80]. Inspired by ViTs that use global attention [20, 67, 80] or attention with large windows [50, 62, 78], RepLKNet proposed to use very large conv kernels. In contrast to the common practice using small kernels (*e.g.*, 3×3) [28, 31, 34, 61, 66, 71, 92], which fails to obtain a large Effective Receptive Field (ERF) [55] even with numerous small-kernel layers, RepLKNet realizes large ERF and impressive performance, especially on tasks such as object detection and semantic segmentation.

Nowadays, ConvNets with very large kernels become popular, which mostly focus on making the large kernels even larger [48], ways to apply them to multiple tasks [9, 54, 90], *etc*. However, we note that most architectures of the existing large-kernel ConvNets simply follow other models, *e.g.*, RepLKNet [19] follows the architecture of Swin Transformer [49], and SLaK [48] follows ConvNeXt, which is a powerful architecture with medium-sized (7×7) kernels. The architectural design for large-kernel ConvNets remains under-explored.

We explore large-kernel ConvNet architecture by rethinking the design of conventional models that employ a deep stack of small kernels. As we add a 3×3 conv to a small-kernel ConvNet, we expect it to take three effects simultaneously - **1)** make the receptive field larger, **2)** increase the abstract hierarchy of spatial patterns (*e.g.*, from angles and textures to shapes of objects), and **3)** improve the model's general representational capability via making it deeper, bringing in more learnable parameters and nonlinearities. In contrast, we argue that such three effects in a large-kernel architecture should be decoupled as the model should utilize the substantial strength of a large kernel - *the ability to see wide without going deep*. Since increasing the kernel size is much more effective than stacking more layers in enlarging the ERF [55], a sufficient ERF can be built up with a small number of large-kernel layers, so that the compute budget can be saved for other efficient structures that are more effective in increasing the abstract hierarchy of spatial patterns or generally increasing the depth. For example, when the objective is to extract higher-level local

*Equal contributions.

spatial patterns from lower-level ones, a 3×3 might be a more suitable option than a large-kernel conv layer. The reason is that the latter demands more computations and may result in patterns no longer restricted to smaller local regions, which could be undesirable in specific scenarios.

Concretely, we propose **four architectural guidelines** for large-kernel ConvNets - **1)** use efficient structures such as SE Blocks [33] to increase the depth, **2)** use a proposed *Dilated Reparam Block* to re-parameterize the large-kernel conv layer to *improve the performance without inference costs*, **3)** decide the kernel size by the downstream task and usually use large kernels only in the middle- and high-level layers, and **4)** add 3×3 conv instead of more large kernels while scaling up the model's depth. A ConvNet built up following such guidelines (Fig. 1) realizes the aforementioned three effects separately, as it uses a modest number of large kernels to guarantee a large ERF, small kernels to extract more complicated spatial patterns more efficiently, and multiple lightweight blocks to further increase the depth to enhance the representational capacity.

Our architecture achieves leading performance on ImageNet classification [12], ADE20K semantic segmentation [98], and COCO object detection [44], outperforming the existing large-kernel ConvNets such as RepLKNet [19], SLaK [48], and recent powerful architectures including ConvNeXt V2 [85], FastViT [77], Swin V2 [51] and DeiT III [75] in terms of both accuracy and efficiency. Moreover, our architecture demonstrates significantly higher shape bias [3, 76] than existing ConvNets and ViTs, *i.e.*, it makes predictions more based on the overall shapes of objects than the textures, which agrees with the human visual system and results in better generalization. This may explain its superiority in downstream tasks. See the Appendix for details.

RepLKNet [19] was proposed partly "in defense of ConvNets" as ViTs dominated multiple image recognition tasks that were once dominated by ConvNets. Moreover, considering transformers have shown universal perception capability in multiple modalities [93, 94], in this work, we seek to not only reclaim the leading position in image recognition tasks by surpassing ViTs' performance but also contribute to areas where ConvNets were not traditionally dominant. Specifically, on audio, video, point cloud, and time-series tasks, we achieve impressive performance with amazingly universal and simple solutions. We use modality-specific preprocessing approaches to transform all the data into 3D embedding maps just like what we do with images and use the same architecture as the backbone to process the embedding maps. Our model shows ***uni*versal perception ability across multiple modalities with a *uni*fied architecture** so it is named **UniRepLKNet**.

Impressively, UniRepLKNet achieves remarkable results even on modalities that were not considered the stronghold of ConvNet, *e.g.*, audio and temporal data. On a huge-scale time-series forecasting task that predicts the global temperature and wind speed, UniRepLKNet, a generalist model originally designed for image recognition, even outperforms the latest state-of-the-art transformer customized for the task. Such results not only signify a ***"comeback"*** for ConvNet in its original domain but also showcase large-kernel ConvNet's potential to ***"conquer"*** new territories, expanding its applicability and versatility in various tasks.

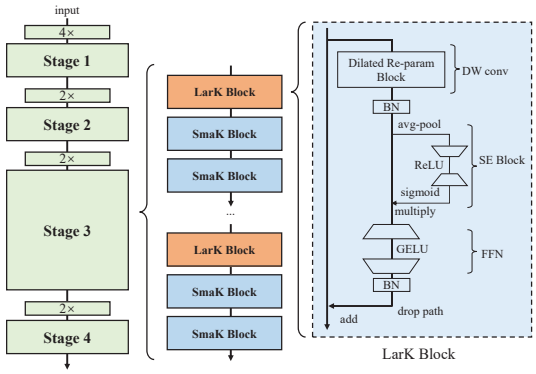

Figure 1. Architectural design of UniRepLKNet. A LarK Block comprises a Dilated Reparam Block proposed in this paper, an SE Block [33], an FFN, and Batch Normalization (BN) [37] layers. The only difference between a SmaK Block and a LarK Block is that the former uses a depth-wise 3×3 conv layer in replacement of the Dilated Reparam Block in the latter. Stages are connected by downsampling blocks implemented by stride-2 dense 3×3 conv layers. We may flexibly arrange the blocks in different stages and the details of our provided instances are shown in Table 5.

2. Related Work

Large kernels in early ConvNets. Classic ConvNets such as AlexNet [42] and Inceptions [68–70] used 7×7 or 11×11 in the low-level layers, but large kernels became not popular after VGG-Net [66]. Global Convolution Network (GCN) [57] used very large conv layers (1×K followed by K×1) for semantic segmentation. Local Relation Networks (LR-Net) [32] adopted a spatial aggregation operator (LR-Layer) to replace the standard conv layer, which can be viewed as a dynamic convolution. LR-Net benefited from a kernel size of 7×7 but degraded with 9×9. With a kernel size as large as the feature map, its top-1 accuracy significantly reduced from 75.7% to 68.4%.

Explorations with large kernels. The concept of kernel may be generalized beyond spatial convolution. Swin Transformer [50] used shifted attention with window sizes ranging from 7 to 12, which can be seen as a dynamic kernel. Han et al. [27] replaced the attention layers in Swin with static or dynamic 7×7 conv and still maintained comparable results. MetaFormer [91] suggested large-kernel pooling layer was an alternative to self-attention. Another representative work was Global Filter Network

(GFNet) [63], which optimized the spatial connection weights in the Fourier domain. It is equivalent to circular global convolutions in the spatial domain.

Modern ConvNets with very large kernels. RepLKNet first proposed that simply scaling up the kernel size of existing ConvNets resulted in improvements, especially on downstream tasks [19]. It proposed several guidelines while using large kernels, which were focused on the microstructural design (*e.g.*, using shortcut alongside large kernel) and application (large-kernel ConvNets should be evaluated on downstream tasks). In terms of the architecture, RepLKNet merely followed Swin Transformer for simplicity. In the past two years, large-kernel ConvNets have been intensively studied. Some works succeeded in further enlarging the kernel sizes [48], generalizing the idea to 3D scenarios [9] and many downstream tasks, *e.g.*, image dehazing [54] and super-resolution [90]. However, we note that the architectural design for ConvNets with very large kernels remains under-explored. For example, SLaK [48] followed the architecture developed by ConvNeXt, which is a powerful architecture of medium-sized (7×7) kernels.

3. Architectural Design of UniRepLKNet

We first summarize the architectural guidelines as follows. **1)** Block design: use efficient structures that perform both inter-channel communications and spatial aggregations to increase the depth. **2)** Re-parameterization: use dilated small kernels to re-parameterize a large kernel. **3)** Kernel size: decide kernel size according to the downstream task and usually use large kernels in middle- and high-level layers. **4)** Scaling Rule: while scaling up the depth, the added blocks should use small kernels. We describe the proposed Dilated Reparam Block in Sec. 3.1 and details in Sec. 3.2.

3.1. Dilated Reparam Block

It is reported a large-kernel conv should be used with a parallel small-kernel one because the latter helps capture the small-scale patterns during training [19]. Their outputs are added up after two respective Batch Normalization (BN) [37] layers. After training, with the Structural Re-parameterization [13–18] methodology, we merge the BN layers into the conv layers so the small-kernel conv can be equivalently merged into the large-kernel one for inference. In this work, we note that except for small-scale patterns, enhancing the large kernel's capability to capture sparse patterns (*i.e.*, a pixel on a feature map may be more related to some distant pixels than its neighbors) may yield features of higher quality. The need to capture such patterns exactly matches the mechanism of dilated convolution - from a sliding-window perspective, a dilated conv layer with a dilation rate of r scans the input channel to capture spatial patterns where each pixel of interest is $r-1$ pixels away from its neighbor. Therefore, we use dilated conv layers parallel to the large kernel and add up their outputs.

To eliminate the inference costs of the extra dilated conv layers, we propose to equivalently transform the whole block into a single non-dilated conv layer for inference. Since *ignoring pixels of the input is equivalent to inserting extra zero entries into the conv kernel*, *a dilated conv layer with a small kernel can be equivalently converted into a non-dilated* (i.e., $r=1$) *layer with a sparse larger kernel*. Let k be the kernel size of the dilated layer, by inserting zero entries, the kernel size of the corresponding non-dilated layer will be $(k-1)r+1$, which is referred to as the *equivalent kernel size* for brevity. We further note that such transformation from the former kernel $\mathrm{W} \in \mathcal{R}^{k\times k}$ to the latter $\mathrm{W}' \in \mathcal{R}^{((k-1)r+1)\times((k-1)r+1)}$ can be elegantly realized by a transpose convolution with a stride of r and an identity kernel $\mathrm{I} \in \mathcal{R}^{1\times 1}$, which is scalar 1 but viewed as a kernel tensor. [1] With pytorch-style pseudo code, that is

$$\mathrm{W}' = \texttt{conv\_transpose2d}(\mathrm{W}, \mathrm{I}, \text{stride} = r)\,. \quad (1)$$

The equivalency can be easily verified - given an arbitrary $\mathrm{W} \in \mathcal{R}^{k\times k}$ and an arbitrary input channel, a convolution with W and a dilation rate r always yields identical results to a non-dilated convolution with W'. [2]

Based on such equivalent transformations, we propose a Dilated Reparam Block, which uses a non-dilated small-kernel and multiple dilated small-kernel layers to enhance a non-dilated large-kernel conv layer. Its hyper-parameters include the size of large kernel K, sizes of parallel conv layers $\boldsymbol{k}$, and the dilation rates $\boldsymbol{r}$. The shown case (Fig. 2) with four parallel layers is denoted by K=9, $\boldsymbol{r}$=(1,2,3,4), $\boldsymbol{k}$=(5,5,3,3). For a larger K, we may use more dilated layers with larger kernel sizes or dilation rates. The kernel sizes and dilation rates of the parallel branches are flexible and the only constraint is $(k-1)r+1 \leq K$. For example, with K=13 (the default setting in our experiments), we use five layers with $\boldsymbol{k}$=(5,7,3,3,3), $\boldsymbol{r}$=(1,2,3,4,5), so the equivalent kernel sizes will be (5,13,7,9,11), respectively. To convert a Dialted Reparam Block into a large-kernel conv layer for inference, we first merge every BN into the preceding conv layer, convert every layer with dilation $r>1$ with function 1, and add up all the resultant kernels with appropriate zero-paddings. For example, the layer in Fig. 2 with k=3,r=3 is converted into a sparse 7×7 kernel and added to the 9×9 kernel with one-pixel zero paddings on each side.

[1] We showcase a single-channel conv and it is easy to generalize the transformation to multi-channel cases. See the Appendix for details.

[2] In common cases where the shape of output equals that of input, *i.e.*, the padding of the former is $\frac{k-1}{2}$, note the padding of the latter should be $\frac{(k-1)r}{2}$ since the size of the equivalent sparse kernel is $(k-1)r+1$.

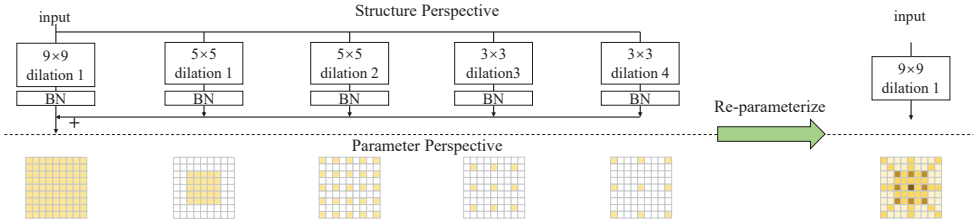

Figure 2. Dilated Reparam Block uses dilated small-kernel conv layers to enhance a non-dilated large-kernel layer. Such dilated layers are equivalent to a non-dilated conv layer with a larger sparse kernel, as shown from the parameter perspective, so that the whole block can be equivalently transformed into a single large-kernel conv. This example shows K=9, and we may use more dilated layers for larger K.

3.2. Architectural Guidelines for Large Kernels

Vanilla architecture. We first construct the vanilla architecture to experiment on. As a common practice, the main body of the model is split into four stages connected by downsampling blocks. Specifically, the first downsampling block uses two stride-2 3×3 conv layers to transform the raw input into C-channel feature maps, where C is an architectural hyper-parameter and the other three downsampling blocks each use one stride-2 3×3 conv layer performing 2× channel expansion so that the numbers of channels of the four stages are C, $2C$, $4C$, and $8C$, respectively. A stage comprises blocks whose vanilla design resembles ConvNeXt, *i.e.*, a *depthwise (DW)* conv layer and a *Feed-Forward Network (FFN)* with GRN unit [85], but we use BN instead of LayerNorm [1] after the conv layer as BN can be equivalently merged into the conv layer to eliminate its inference costs. We use another BN after the FFN, which can also be equivalently merged into the preceding layer (*i.e.*, the second linear layer in FFN). The numbers of such blocks in the four stages are denoted by $\boldsymbol{N}$ (N_1,N_2,N_3,N_4), respectively. Following ConvNeXt-T, the vanilla architecture uses C=96 and $\boldsymbol{N}$=(3,3,9,3). By default, the last three stages use 13×13 Dilated Reparam Block as the DW layer, which means K=13, $\boldsymbol{k}$=(5,7,3,3,3) and $\boldsymbol{r}$=(1,2,3,4,5); the first stage uses DW 3×3 conv as the DW layer.

Experimental settings and metrics. It has been emphasized in the literature [19] that large-kernel ConvNets should be evaluated on downstream tasks, as their full potential may not be accurately reflected by the ImageNet accuracy alone. Therefore, except for the ImageNet-1K accuracy after 100-epoch training, we transfer the trained model with UPerNet [89] to ADE20K to examine its performance on semantic segmentation and report the single-scale mIoU after a 160k-iteration standard finetuning process [10]. Besides the parameters and FLOPs, we test the actual throughput on an A100 GPU with a batch size of 128 and input resolution of 224×224, which is measured in images per second (img/s). See the Appendix for detailed configurations.

Architectural Guideline 1 on Block Design: use efficient structures that perform both inter-channel communications and spatial aggregations to increase the depth. We first seek to insert some structures to universally boost the model's representational capacity, which is required to comprise nonlinearity and efficient trainable transformations. We naturally try a bottleneck composed of a 1×1 conv that reduces the channels to 1/4, a DW 3×3 conv, and another 1×1 conv to expand the channels back (Fig. 3). We use BN and ReLU after conv layers as a common practice. Table 1 shows that the performance improves with acceptable costs (+1.2 mIoU with 12% slow down). The performance degrades as we remove the DW 3×3 conv so that only two 1×1 conv layers remain, or replace the bottleneck structure with two DW 3×3 layers, suggesting that such structures require both spatial aggregation transformations and channel mixing. Motivated by this, considering that SE Block [33] elegantly realizes both transformations in a more efficient way (*i.e.*, global average pooling and nonlinear mapping of the pooled vectors), we try it also with 1/4 channel reduction and observe a better performance and higher throughput. We therefore use SE Block as a substructure of our block design in the following explorations.

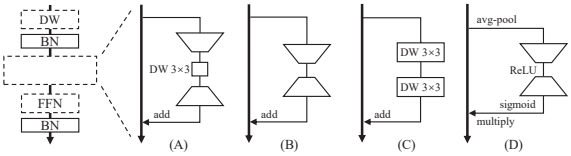

Figure 3. Options of the extra structures to increase the depth.

Table 1. Models with different efficient extra structures to increase the depth. We report the ImageNet accuracy (Acc), ADE20K mIoU, and actual throughput (Img/s).

Extra structure	Params	FLOPs	Img/s	Acc	mIoU
None	31.3M	4.92G	1954	81.2	45.1
(A) Bottleneck	32.9M	5.18G	1716	81.5	46.3
(B) Two 1×1	32.9M	5.17G	1745	81.3	46.2
(C) Two DW 3×3	31.4M	4.96G	1659	81.3	45.4
(D) SE Block	32.9M	4.92G	1863	**81.6**	**46.5**

Architectural Guideline 2 on Re-parameterization: use dilated small kernels to re-parameterize a large kernel. For a fair comparison with Dilated Reparam Block, we try two variants with the same numbers of parallel branches composed of non-dilated layers with **A)** the same kernel sizes or **B)** the same equivalent kernel sizes. For our default setting of K=13, $\boldsymbol{r}$=(1,2,3,4,5), $\boldsymbol{k}$=(5,7,3,3,3), the kernel sizes of the five branches will be $\boldsymbol{k}$=(5,7,3,3,3) or

Table 2. Different forms of Structural Re-parameterization on the 13×13 conv layers. We report the mean±std of three runs.

Re-param	k	r	Acc	mIoU
None	N/A	N/A	81.44±0.04	45.78±0.05
Dilated Reparam	5,7,3,3,3	1,2,3,4,5	81.63±0.02	**46.37**±0.10
Same kernel size	5,7,3,3,3	1,1,1,1,1	81.55±0.01	46.07±0.07
Same eq kernel size	5,13,7,9,11	1,1,1,1,1	81.59±0.02	46.17±0.04

(5,13,7,9,11) for the two variants, respectively. All the models end up with the same inference structure but the training structures differ. Table 2 shows lower performance of variants, suggesting that large kernel benefits from the parallel dilated conv layers' abilities to capture sparse patterns, rather than merely the extra small kernels (variant A) or the combination of different receptive fields (variant B). We use Dilated Reparam Block in the following explorations. [3]

Architectural Guideline 3 on Kernel Size: decide kernel size according to the downstream task and usually use large kernels in middle- and high-level layers. As introduced above, the baseline model uses 3×3 conv in the first stage and 13×13 in the last three stages. Table 3 shows that replacing the large kernels in the last three stages with 3×3 or changing K from 13 to 11 degrades the models, especially in the ADE20K mIoU, which highlights the significance of large kernels. Interestingly, using 13×13 in Stage 1 or enlarging K from 13 to 15 makes almost no difference in the ImageNet accuracy but reduces the ADE20K mIoU.

Remark. We argue that this phenomenon does not mean larger kernels result in lower feature quality. It is due to the structural priors of UPerNet, which takes the features extracted by the low-level layers of the backbone and assumes they should only encode local information so that combining them with the high-level features extracted from the last layers of the backbone results in better segmentation. With larger kernels in lower stages, the low-level features are no longer confined to small local areas so the UPerNet benefits less from combining them with the high-level features. We verify this explanation by making the UPerNet only use the high-level features (*i.e.*, outputs of Stage 4) to evaluate the quality of the eventual features alone. Under this setting, K=15 delivers the best mIoU (42.7), the model with large kernels in Stage 1 performs as well as the baseline (42.4), and K=11 performs the worst (41.9). Such observations confirm that large kernels, even when they are used inappropriately, *do not damage the feature quality* of ConvNet but *merely make the low-level features less favorable for certain downstream models that require local low-level features*, suggesting we should decide the kernel size according to the specific downstream tasks and framework. In our specific use cases (*i.e.*, representative image recognition tasks with common downstream frameworks), we employ 13×13 kernels in the middle- and high-level stages by default.

Architectural Guideline 4 on the Scaling Rule: while scaling up the depth, the added blocks should use small kernels. The scaling rule of existing large-kernel ConvNets follows the traditional ConvNets, *i.e.*, stacking more large kernels to build up a deeper model, but we argue that a large-kernel ConvNet may not benefit from more large kernels. In this group of experiments (Table 4), we scale up N_3 from 9 to 27, following ConvNeXt-S [52]. Considering that nine 13×13 blocks may have already built up sufficient receptive field, we examine if the added blocks should also use large kernels. Specifically, we refer to the block with a Dilated Reparam Block as the *Large Kernel Block (LarK Block)* and name a block that uses a DW 3×3 conv as a *Small Kernel Block (SmaK Block)* so that there are 3 SmaK Blocks in Stage 1 and 3/9/3 LarK Blocks in Stage 2/3/4 of the shallow model. While scaling up the depth of Stage 3, we tried the following options. **A)** All of the 27 blocks are LarK Blocks. **B)** We interleave SmaK with LarK Blocks so that Stage 3 has 14 LarK Blocks and 13 SmaK Blocks. **C)** We place two SmaK Blocks after a LarK Block so that the resultant model will have the same 9 LarK Blocks as before but 18 extra SmaK Blocks. **D)** We remove the DW 3×3 layers in SmaK Blocks. Table 4 shows that scaling up the depth brings significant improvements, which is expected, and 9 LarK Blocks are sufficient. Though 27 LarK Blocks perform slightly better in the ADE20K mIoU, the inference speed is observably slowed down. Besides, the model without 3×3 conv in SmaK Blocks shows significantly lower mIoU with only minor improvements in the throughput, suggesting such small kernels in SmaK Blocks are useful while scaling up the depth of large-kernel ConvNet as they increase the abstract hierarchy of spatial patterns, though they may not effectively enlarge the ERF [19, 55]. This observation supports our motivation to decouple the effects of conv layers in enlarging the ERF and extracting more complicated spatial patterns, as discussed in Sec. 1.

Table 3. Models with different kernel sizes in the four stages denoted by S1 - S4. Numbers in parentheses are obtained with the UPerNet only taking the outputs of S4.

S1	S2	S3	S4	Params	FLOPs	Img/s	Acc	mIoU
3	13	13	13	32.9M	4.92G	1863	81.6	**46.5** (42.4)
3	11	11	11	32.6M	4.86G	1876	81.6	45.5 (41.9)
3	3	13	13	32.8M	4.85G	2006	81.7	46.1
3	13	3	13	32.4M	4.81G	2015	81.6	45.9
3	13	13	3	32.5M	4.90G	1884	81.4	45.8
3	15	15	15	33.3M	4.99G	1851	81.7	45.9 (**42.7**)
13	13	13	13	33.0M	5.06G	1547	81.6	44.9 (42.4)

Table 4. Different numbers of LarK and SmaK Blocks in Stage 3.

N_3	LarK	SmaK	Params	FLOPs	Img/s	Acc	mIoU
9	9	0	32.9M	4.92G	1863	81.6	**46.5**
27	27	0	56.7M	9.31G	1145	82.3	49.0
27	14	13, 3×3	55.9M	9.15G	1229	82.3	48.8
27	9	18, 3×3	55.6M	9.10G	1264	82.3	48.8
27	9	18, w/o 3×3	55.5M	9.08G	1289	82.2	47.8

3.3. Architectural Specifications

Following our proposed guidelines, we instantiate a series of models (Table 5). For a fair comparison with ConvNeXt V2 [85], UniRepLKNet-A/F/P/N follows its configurations.

[3] While describing the architecture in this paper, using a $K\times K$ ($K\geq 9$) conv means a $K\times K$ Dilated Reparam Block, unless otherwise noted.

Table 5. Architectural hyper-parameters of UniRepLKNet instances, including the number of blocks in the four stages N_1, N_2, N_3, N_4 and channels C of the first stage. Stage 1 uses SmaK Blocks, and Stages 2 and 4 use LarK Blocks only. For Stage 3, *e.g.*, "9 + 18" means 9 LarK Blocks and 18 SmaK Blocks.

	N_1	N_2	N_3	N_4	C	Params
UniRepLKNet-A	2	2	6 + 0	2	40	4.4M
UniRepLKNet-F	2	2	6 + 0	2	48	6.2M
UniRepLKNet-P	2	2	6 + 0	2	64	10.7M
UniRepLKNet-N	2	2	8 + 0	2	80	18.3M
UniRepLKNet-T	3	3	9 + 9	3	80	31.0M
UniRepLKNet-S	3	3	9 + 18	3	96	55.6M
UniRepLKNet-B	3	3	9 + 18	3	128	97.9M
UniRepLKNet-L	3	3	9 + 18	3	192	218.3M
UniRepLKNet-XL	3	3	9 + 18	3	256	386.4M

We scale up the depth to build UniRepLKNet-T/S and scale up the width to construct UniRepLKNet-S/B/L/XL.

3.4. Generalizing UniRepLKNet beyond Image

To utilize the universal perception ability of UniRepLKNet, we preprocess the data of different modalities into $B \times C' \times H \times W$ embedding maps, where B is the batch size and C' is determined by the modality, and configure the input channel of the first layer of UniRepLKNet to C'. For simplicity, the other parts of the models are the same as the UniRepLKNet initially designed for the image without any modality-specific customization. By doing so, we directly apply a ConvNet typically used for image tasks to deal with data of other modalities. In other words, the UniRepLKNet for image tasks can be seen as a general UniRepLKNet with C'=3 and no such preprocessing. We introduce how to transform the data into such embedding maps as follows.

Time-series. Let L and D be the length and dimensions of a time-series sequence $\boldsymbol{x}_T \in \mathbb{R}^{B\times L\times D}$, we adopt the embedding layer in Corrformer [86] to split it into n nodes then project it into a latent space $\mathbb{R}^{Bn\times L\times D'}$ (D' and n are configurable hyper-parameters of the embedding layer). Then we simply reshape it into a single-channel embedding map.

$$\begin{aligned}\boldsymbol{x}_T \in \mathbb{R}^{B\times L\times D} &\to \mathbb{R}^{Bn\times L\times \frac{D}{n}} \to \mathbb{R}^{Bn\times L\times D'} \\ &\to \mathbb{R}^{Bn\times 1\times H\times W} \text{ s.t.} HW = LD'.\end{aligned} \tag{2}$$

Audio. Let T and F be the numbers of time frames and frequency bins, we use $\boldsymbol{x}_A \in \mathbb{R}^{B\times T\times F}$ to represent audio data. A sample is seen as a $1 \times T \times F$ embedding map that resembles a single-channel image so C'=1, H=T, W=F.

$$\boldsymbol{x}_A \in \mathbb{R}^{B\times T\times F} \to \mathbb{R}^{B\times 1\times T\times F}. \tag{3}$$

Point cloud. Assume a sample comprises P points each represented by the X/Y/Z coordinates, we use a series of conv layers to generate three-view projections [93]. We configure the resolution of the generated projections to be 224 so that H=W=224, C'=3.

$$\boldsymbol{x}_P \in \mathbb{R}^{B\times P\times 3} \to \mathbb{R}^{B\times 3\times 224\times 224}. \tag{4}$$

Video. We represent a video as N_F frames and each frame is a $3 \times h \times w$ image. We reshape it by merging the frame dimension into the height and width dimensions so that we obtain a representation that can be viewed as a single image created by laying out (*i.e.*, concatenating) the N_F frames. For example, in our experiments, we have N_F=16 and h=w=224 so that H=W=896. Generally,

$$\boldsymbol{x}_V \in \mathbb{R}^{B\times N_F\times 3\times h\times w} \to \mathbb{R}^{B\times 3\times H\times W} \text{ s.t.} \frac{HW}{hw} = N_F\,. \tag{5}$$

4. UniRepLKNet for Image Recognition

ImageNet classification. Following ConvNeXt [52], we use the widely adopted 300-epoch receipt to train UniRepLKNet-A/F/P/N/T/S on ImageNet-1K; we pretrain UniRepLKNet-S/B/L/XL on ImageNet-22K using the 90-epoch receipt and fine-tune with ImageNet-1K for 30 epochs (see the Appendix for details). As our goal is to develop models that *run with high actual speed*, we evaluate the actual throughput on the same A100 GPU using a batch size of 128. Table 6 shows the top-1 accuracy on the ImageNet-1K validation set where the results are sorted by the throughput. We split the results into seven segments for better readability. **1)** UniRepLKNet-A/F outperforms ConvNeXt-V2-A/F by 0.8/0.6 in the accuracy and runs 19%/17% faster, respectively. **2)** UniRepLKNet-P/N outperforms FastViT-T12/S12 and ConvNeXt V2-P/N by clear margins. **3)** UniRepLKNet-T outperforms multiple small-level competitors. **4)** UniRepLKNet-S outperforms a series of small-level and even base-level models in both speed and accuracy and runs almost as fast as InternImage-T. **5)** With ImageNet-22K pretraining, UniRepLKNet-S even approaches the accuracy of RepLKNet-31L and runs 3× as fast as the latter. UniRepLKNet-B outperforms CoAtNet-2 and DeiT III-B by clear margins. UniRepLKNet-L outperforms InternImage-L in both accuracy and throughput. **6)** On the XL-level, UniRepLKNet-XL outperforms in both accuracy and throughput, running more than 2× as fast as CoAtNet-3 and 3× as DeiT III-L.

COCO object detection and instance segmentation. We transfer the pretrained UniRepLKNets as the backbones of Cascade Mask R-CNN [4, 29] and adopt the standard 3x (36-epoch) training configuration with MMDetection [8]. Table 7 shows UniRepLKNet outperforms Swin, ConvNeXt, RepLKNet, and SLaK, which are representatives of ViTs, modern medium-kernel ConvNets, and existing large-kernel ConvNets, respectively, and shows comparable performance to InternImage [82], which is a latest powerful architecture with deformable convolution.

ADE20K semantic segmentation. We use the pretrained UniRepLKNets as the backbones of UPerNet [89] on ADE20K [98] and adopt the standard 160k-iteration training receipt with MMSegmentation [10]. Table 8 reports the mIoU on the validation set. Impressively, UniRepLKNet outperforms InternImage and the other models.

Table 6. **ImageNet classification**. Throughput is tested with an A100 GPU and batch size of 128. "T/C" denote transformer/ConvNet. "‡" indicates ImageNet-22K [12] pretraining.

Method	Type	Input size	Params (M)	FLOPs (G)	Throughput (img/s)	Acc (%)
UniRepLKNet-A	C	224^2	4.4	0.6	**5942**	**77.0**
UniRepLKNet-F	C	224^2	6.2	0.9	**5173**	**78.6**
ConvNeXt V2-A [85]	C	224^2	3.7	0.5	5054	76.2
FastViT-T8 [77]	T	256^2	3.6	0.7	5025	75.6
ConvNeXt V2-F [85]	C	224^2	5.2	0.8	4329	78.0
UniRepLKNet-P	C	224^2	10.7	1.6	**3949**	**80.2**
FastViT-T12 [77]	T	256^2	6.8	1.4	3407	79.1
ConvNeXt V2-P [85]	C	224^2	9.1	1.4	3339	79.7
FastViT-S12 [77]	T	256^2	8.8	1.8	3162	79.8
UniRepLKNet-N	C	224^2	18.3	2.8	**2807**	**81.6**
ConvNeXt V2-N [85]	C	224^2	15.6	2.4	2405	81.2
UniRepLKNet-T	C	224^2	31	4.9	**1804**	**83.2**
FastViT-SA24 [77]	T	256^2	21	3.8	1670	82.6
PVTv2-B2 [81]	T	224^2	25	4.0	1620	82.0
CoAtNet-0 [11]	T	224^2	25	4.2	1613	81.6
DeiT III-S [75]	T	224^2	22	4.6	1485	81.4
SwinV2-T/8 [51]	T	256^2	28	6	1406	81.8
SLaK-T [48]	C	224^2	30	5.0	1312	82.5
InternImage-T [82]	C	224^2	30	5	1292	83.5
UniRepLKNet-S	C	224^2	56	9.1	**1265**	**83.9**
ConvNeXt-S [52]	C	224^2	50	8.7	1182	83.1
HorNet-T [64]	C	224^2	23	3.9	1162	83.0
FastViT-SA36 [77]	T	256^2	30	5.6	1151	83.6
CoAtNet-1 [11]	T	224^2	42	8.4	969	83.3
SLaK-S [48]	C	224^2	55	9.8	967	83.8
FastViT-MA36 [77]	T	256^2	43	7.9	914	83.9
SwinV2-S/8 [51]	T	256^2	50	12	871	83.7
RepLKNet-31B [19]	C	224^2	79	15.3	859	83.5
PVTv2-B5 [81]	T	224^2	82	11.8	802	83.8
UniRepLKNet-S‡	C	384^2	56	26.7	**435**	**86.4**
ConvNeXt-S‡ [52]	C	384^2	50	25.5	415	85.8
UniRepLKNet-B‡	C	384^2	98	47.2	**314**	**87.4**
ConvNeXt-B‡ [52]	C	384^2	89	45.1	304	86.8
UniRepLKNet-L‡	C	384^2	218	105.4	**190**	**87.9**
ConvNeXt-L‡ [52]	C	384^2	198	101	185	87.5
CoAtNet-2‡ [11]	T	384^2	75	49.8	163	87.1
RepLKNet-31L‡ [19]	C	384^2	172	96.0	158	86.6
InternImage-L‡ [82]	C	384^2	223	108	143	87.7
DeiT III-B‡ [75]	T	384^2	87	55.5	138	86.7
UniRepLKNet-XL‡	C	384^2	386	187	**131**	**88.0**
ConvNeXt-XL‡ [52]	C	384^2	350	179	129	87.8
HorNet-L‡ [64]	C	384^2	202	102	127	87.7
InternImage-XL‡ [82]	C	384^2	335	163	114	88.0
CoAtNet-3‡ [11]	T	384^2	168	107	103	87.6
SwinV2-L/24‡ [51]	T	384^2	197	115	88	87.6
CoAtNet-4‡ [11]	T	384^2	275	190	58	87.9
DeiT III-L‡ [75]	T	384^2	305	191	42	87.7

5. Universal Perception on other Modalities

Time-series. Following Corrformer [86], we conduct experiments on the Global Temperature and Wind Speed Forecasting challenge [4] using the dataset collected from the National Centers for Environmental Information (NCEI). This huge-scale dataset contains hourly averaged wind speed and temperature data from 3,850 stations with different geographical scales and densities, spanning from 2019 to 2021. For a fair comparison with Corrformer, which was the previous state-of-the-art method, we use its embedding layer (as introduced in Sec. 3.4) and decoder and only replace its encoder transformer with UniRepLKNet-S. We also compare UniRepLKNet-S against a wide range of methods, including statistical and numerical approaches. Table 9 shows UniRepLKNet delivers a new state-of-the-art forecasting precision, achieving the lowest errors of 7.602, 1.832, 3.865, and 1.301 for MSE and MAE in forecasting global temperature and wind speed, respectively, with fewer parameters than existing deep learning methods. It is partic-

[4] https://codeocean.com/capsule/0341365/tree/v1

Table 7. **Object detection on COCO validation set**. FLOPs are measured with 1280×800 inputs. "‡" ImageNet-22K pretraining.

Method	Params (M)	FLOPs (G)	AP^{box}	AP^{mask}
UniRepLKNet-T	89	749	**51.8**	**44.9**
Swin-T [49]	86	745	50.4	43.7
ConvNeXt-T [52]	86	741	50.4	43.7
SLaK-T [48]	-	-	51.3	44.3
UniRepLKNet-S	113	835	**53.0**	**45.9**
Swin-S [49]	107	838	51.9	45.0
ConvNeXt-S [52]	108	827	51.9	45.0
UniRepLKNet-S‡	113	835	**54.3**	**47.1**
UniRepLKNet-B‡	155	978	**54.8**	**47.4**
Swin-B‡ [49]	145	982	53.0	45.8
ConvNeXt-B‡ [52]	146	964	54.0	46.9
RepLKNet-31B‡ [19]	137	965	52.2	45.2
UniRepLKNet-L‡	276	1385	55.8	48.4
Swin-L‡ [49]	253	1382	53.9	46.7
ConvNeXt-L‡ [52]	255	1354	54.8	47.6
RepLKNet-31L‡ [19]	229	1321	53.9	46.5
InternImage-L‡ [82]	277	1399	**56.1**	**48.5**
UniRepLKNet-XL‡	443	1952	**56.4**	**49.0**
InternImage-XL‡ [82]	387	1782	56.2	48.8
ConvNeXt-XL‡ [52]	407	1898	55.2	47.7

Table 8. **Semantic segmentation on ADE20K validation set**. The FLOPs are measured with 512×2048 or 640×2560 inputs according to the crop size. "SS" and "MS" mean single- and multi-scale testing, respectively. "‡" ImageNet-22K [12] pretraining.

Method	Crop size	Params (M)	FLOPs (G)	mIoU (SS)	mIoU (MS)
UniRepLKNet-T	512^2	61	946	**48.6**	**49.1**
Swin-T [49]	512^2	60	945	44.5	45.8
ConvNeXt-T [52]	512^2	60	939	46.0	46.7
SLaK-T [48]	512^2	65	936	47.6	-
InternImage-T [82]	512^2	59	944	47.9	48.1
UniRepLKNet-S	512^2	86	1036	**50.5**	**51.0**
Swin-S [49]	512^2	81	1038	47.6	49.5
ConvNeXt-S [52]	512^2	82	1027	48.7	49.6
SLaK-S [48]	512^2	91	1028	49.4	-
InternImage-S [82]	512^2	80	1017	50.1	50.9
UniRepLKNet-S‡	640^2	86	1618	**51.9**	**52.7**
UniRepLKNet-B‡	640^2	130	1850	**53.5**	**53.9**
Swin-B‡ [49]	640^2	121	1841	50.0	51.7
ConvNeXt-B‡ [52]	640^2	122	1828	52.6	53.1
RepLKNet-31B‡ [19]	640^2	112	1829	51.5	52.3
UniRepLKNet-L‡	640^2	254	2507	**54.5**	**55.1**
Swin-L‡ [49]	640^2	234	2468	52.1	53.5
RepLKNet-31L‡ [19]	640^2	207	2404	52.4	52.7
ConvNeXt-L‡ [52]	640^2	235	2458	53.2	53.7
InternImage-L‡ [82]	640^2	256	2526	53.9	54.1
UniRepLKNet-XL‡	640^2	425	3420	**55.2**	**55.6**
ConvNeXt-XL‡ [52]	640^2	391	3335	53.6	54.0
InternImage-XL‡ [82]	640^2	368	3142	55.0	55.3

Table 9. **Time-series forecasting** performance on Global Temperature and Wind Speed Forecasting challenge. UniRepLKNet delivers a new state-of-the-art performance in Mean Squared Error (MSE) and Mean Absolute Error (MAE). GFS (https://www.ncei.noaa.gov/) stands for the Global Forecasting System.

Method	Type	Params	Temperature		Wind speed	
			MSE ↓	MAE ↓	MSE ↓	MAE ↓
Statistics-based						
Holt–Winters [36]	-	-	13.241	2.262	5.912	1.664
Prophet [72]	-	-	11.626	2.946	9.691	2.382
GDBT [NeurIPS'17] [40]	-	-	9.706	2.214	4.101	1.417
Numerical Simulation						
GFS (reanalysis)	-	-	14.933	2.287	9.993	2.340
ERA5 (reanalysis) [30]	-	-	13.448	1.908	4.999	1.587
DeepAR [65]	-	-	32.249	4.262	5.248	1.602
N-BEATS [56]	-	-	9.203	2.117	4.124	1.390
Deep Learning Specialist						
StemGNN [NeurIPS'20] [6]	GNN	180M	13.926	2.746	4.066	1.389
Pyraformer [ICLR'21] [47]	Transformer	158M	23.326	3.669	4.614	1.514
Corrformer [Nat. Mach. Intell.'23] [86]	Transformer	155M	7.709	1.888	3.889	1.304
Generalist						
UniRepLKNet-S	ConvNet	132M	**7.602**	**1.832**	**3.865**	**1.301**

Table 10. **Audio recognition** on Speech Commands V2 dataset.

Method	Pretrain	Type	Acc. (%)	Params
PANNS [41]	-	ConvNet	61.8	-
PSLA [25]	IN-1K	ConvNet	96.3	-
AST [24]	AS-2M	Transformer	96.2	86.9M
SSAST [26]	AS-2M	Transformer	97.8	89.3M
Audio-MAE [35]	AS-2M	Transformer	98.3	86.2M
Meta-Transformer [93]	LAION-2B	Transformer	97.0	86.6M
UniRepLKNet-S	-	ConvNet	**98.5**	**55.5M**

ularly noteworthy that UniRepLKNet, a generalist model, outperforms time-series specialists such as Pyraformer [47] and Corrformer [86] in both precision and efficiency. The significant advantages of UniRepLKNet open up new avenues for architectural discussions in time-series forecasting, presenting a viable alternative to transformer models.

Audio. We use Speech Commands V2 [84], which contains 105,829 one-second recordings of 35 common speech commands. Table 10 shows UniRepLKNet seamlessly adapts to audio and delivers an impressive accuracy of 98.5%, even without pretraining. Compared to transformers such as AST [24] and Audio-MAE [35], UniRepLKNet stands out with fewer parameters. Compared to previous ConvNets designed for audio, UniRepLKNet achieves better performance without customizations to the structure, highlighting the untapped potential of ConvNets in the realm of audio.

Video. Kinetics-400 [39] contains 240k training videos and 20k validation videos, spanning 400 classes for action recognition. Though the top-1 accuracy of 54.8% is somewhat behind state-of-the-art architectures like MViT [43], we note that UniRepLKNet is a generalist model without pretraining. Compared to the latest generalist methods, ImageBind [23] and Meta-Transformer [93], UniRepLKNet shows higher accuracy and requires no pretraining.

Point cloud. We explore the versatility of UniRepLKNet by assessing its proficiency in learning 3D patterns, extending beyond the conventional 2D signals of images and audio. We use the ModelNet-40 [88] 3D shape classification task with 9,843/2,468 training/validation samples of CAD models from 40 classes. Table 12 shows UniRepLKNet achieves

Table 11. **Video recognition** accuracy on Kinetics-400.

Method	Pretrain	Type	Acc (%)	Params
Specialist				
SlowFast-101 [21]	IN-1K	ConvNet+RNN	79.8	62.8M
MViTv2-B [43]	IN-1K	Transformer	81.2	51.2M
TimeSFormer [2]	K400	Transformer	80.7	122M
Generalist				
Meta-Transformer [93]	LAINON-2B	Transformer	47.3	86.9M
ImageBind [23]	CLIP Data	Transformer	50.0	632M
UniRepLKNet-S	-	ConvNet	54.8	55.5M

Table 12. **Point cloud analysis** on ModelNet-40 dataset.

Method	Type	ModelNet-40	
		mAcc (%)	OA (%)
PointNet [59]	MLP	86.0	89.2
PointNet++ [60]	MLP	-	91.9
PointConv [87]	ConvNet	-	92.5
KPConv [73]	ConvNet	-	92.9
DGCNN [83]	ConvNet	90.2	92.9
OpenShape [46]	Transformer	83.4	-
UniRepLKNet-S	ConvNet	**90.3**	**93.2**

Table 13. Universal perception performance with other ConvNets or UniRepLKNet with a smaller kernel size.

Modality	Time-Series	Point Cloud	Audio	Video
	MAE↓	OA (%)	Acc (%)	Acc (%)
ResNet-101 [28] (K=3)	7.846	92.6	73.6	41.3
ConvNeXt-S [52] (K=7)	7.641	92.7	94.3	48.5
UniRepLKNet-S (K=11)	7.751	92.9	94.7	51.7
UniRepLKNet-S (K=13)	**7.602**	**93.2**	**98.5**	**54.8**

an Overall Accuracy (OA) of 93.2% and a mean Accuracy (mAcc) of 90.3%. Such outcomes highlight the potential of further developing ConvNets in this domain.

Impact of kernel size on the performance. To investigate the influence of different kernel sizes on performance, we compare UniRepLKNet with models of smaller kernels. We adopted the same modality-specific preprocessing approaches and training configurations for a fair comparison. We take ResNet-101 as a representative small-kernel ConvNet because it has comparable parameters to UniRepLKNet-S. Table 13 shows large kernels are crucial for universal perception, at least in our specific cases.

6. Conclusion

UniRepLKNet shows a leading performance in image recognition and achieves remarkable results on audio and time-series data, outperforming multiple specialist models on those modalities. Such results signify a *"**comeback**"* for ConvNet in its original domain and showcase large-kernel ConvNet's potential to *"**conquer**"* new territories. The limitations are noticeable, *e.g.*, the dilated branches require more training resources, which may be upgraded with simpler [5] or gradient [18] re-parameterization; the applications to large vision-language models [38, 45, 79], cross-attention-based scenarios [7, 96], and generation tasks [58, 95] remain under-explored.

Acknowledgements. This work is partially supported by the National Natural Science Foundation of China (Grant No. 8326014) and CUHK Direct Grants (Grant No. 4055190).

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

[2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, page 4, 2021. 8

[3] bethgelab. Toolbox of model-vs-human. https://github.com/bethgelab/model-vs-human, 2022. 2

[4] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: High quality object detection and instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 6

[5] Zhicheng Cai, Xiaohan Ding, Qiu Shen, and Xun Cao. Refconv: Re-parameterized refocusing convolution for powerful convnets. *arXiv preprint arXiv:2310.10563*, 2023. 8

[6] Defu Cao, Yujing Wang, Juanyong Duan, Ce Zhang, Xia Zhu, Congrui Huang, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, et al. Spectral temporal graph neural network for multivariate time-series forecasting. *Advances in neural information processing systems*, 33:17766–17778, 2020. 8

[7] Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 357–366, 2021. 8

[8] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 6

[9] Yukang Chen, Jianhui Liu, Xiangyu Zhang, Xiaojuan Qi, and Jiaya Jia. Largekernel3d: Scaling up kernels in 3d sparse cnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13488–13498, 2023. 1, 3

[10] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. 4, 6

[11] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *Advances in neural information processing systems*, 34:3965–3977, 2021. 7

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. 2, 7

[13] Xiaohan Ding, Yuchen Guo, Guiguang Ding, and Jungong Han. Acnet: Strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1911–1920, 2019. 3

[14] Xiaohan Ding, Honghao Chen, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Repmlpnet: Hierarchical vision mlp with re-parameterized locality. *arXiv preprint arXiv:2112.11081*, 2021.

[15] Xiaohan Ding, Tianxiang Hao, Jianchao Tan, Ji Liu, Jungong Han, Yuchen Guo, and Guiguang Ding. Resrep: Lossless cnn pruning via decoupling remembering and forgetting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4510–4520, 2021.

[16] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Diverse branch block: Building a convolution as an inception-like unit. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10886–10895, 2021.

[17] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13733–13742, 2021.

[18] Xiaohan Ding, Honghao Chen, Xiangyu Zhang, Kaiqi Huang, Jungong Han, and Guiguang Ding. Re-parameterizing your optimizers rather than architectures. *arXiv preprint arXiv:2205.15242*, 2022. 3, 8

[19] 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, 2, 3, 4, 5, 7

[20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 1

[21] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019. 8

[22] Chongjian Ge, Xiaohan Ding, Zhan Tong, Li Yuan, Jiangliu Wang, Yibing Song, and Ping Luo. Advancing vision transformers with group-mix attention. *arXiv preprint arXiv:2311.15157*, 2023. 1

[23] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15180–15190, 2023. 8

[24] Yuan Gong, Yu-An Chung, and James Glass. Ast: Audio spectrogram transformer. *arXiv preprint arXiv:2104.01778*, 2021. 8

[25] Yuan Gong, Yu-An Chung, and James Glass. Psla: Improving audio tagging with pretraining, sampling, labeling, and

aggregation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3292–3306, 2021. 8

[26] Yuan Gong, Cheng-I Lai, Yu-An Chung, and James Glass. Ssast: Self-supervised audio spectrogram transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10699–10709, 2022. 8

[27] Qi Han, Zejia Fan, Qi Dai, Lei Sun, Ming-Ming Cheng, Jiaying Liu, and Jingdong Wang. Demystifying local vision transformer: Sparse connectivity, weight sharing, and dynamic weight. *arXiv preprint arXiv:2106.04263*, 2021. 2

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 8

[29] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 6

[30] H Hersbach, B Bell, P Berrisford, Sh Hirahara, A Horányi, J Muñoz-Sabater, J Nicolas, C Peubey, R Radu, Di Schepers, et al. The era5 global reanalysis, qj roy. meteor. soc., 146, 1999–2049, 2020. 8

[31] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 1

[32] Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3464–3473, 2019. 2

[33] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 2, 4

[34] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2261–2269. IEEE Computer Society, 2017. 1

[35] Po-Yao Huang, Hu Xu, Juncheng Li, Alexei Baevski, Michael Auli, Wojciech Galuba, Florian Metze, and Christoph Feichtenhofer. Masked autoencoders that listen. *Advances in Neural Information Processing Systems*, 35: 28708–28720, 2022. 8

[36] Rob J Hyndman and George Athanasopoulos. Forecasting: Principles and practice. otexts; 2014. *Online at http://otexts. org/fpp*, 2017. 8

[37] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015. 2, 3

[38] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. *arXiv preprint arXiv:2102.05918*, 2021. 8

[39] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 8

[40] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017. 8

[41] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D Plumbley. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2880–2894, 2020. 8

[42] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 2

[43] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4804–4814, 2022. 8

[44] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2

[45] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 8

[46] Minghua Liu, Ruoxi Shi, Kaiming Kuang, Yinhao Zhu, Xuanlin Li, Shizhong Han, Hong Cai, Fatih Porikli, and Hao Su. Openshape: Scaling up 3d shape representation towards open-world understanding. *arXiv preprint arXiv:2305.10764*, 2023. 8

[47] Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International conference on learning representations*, 2021. 8

[48] Shiwei Liu, Tianlong Chen, Xiaohan Chen, Xuxi Chen, Qiao Xiao, Boqian Wu, Tommi Kärkkäinen, 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, 2, 3, 7

[49] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. *arXiv preprint arXiv:2111.09883*, 2021. 1, 7

[50] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 1, 2

[51] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In

Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 12009–12019, 2022. 2, 7

[52] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *arXiv preprint arXiv:2201.03545*, 2022. 5, 6, 7, 8, 1

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

[54] Pinjun Luo, Guoqiang Xiao, Xinbo Gao, and Song Wu. Lkd-net: Large kernel convolution network for single image dehazing. In *2023 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1601–1606. IEEE, 2023. 1, 3

[55] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard S. Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4898–4906, 2016. 1, 5

[56] Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*, 2019. 8

[57] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. Large kernel matters–improve semantic segmentation by global convolutional network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4353–4361, 2017. 2

[58] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 8

[59] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 8

[60] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017. 8

[61] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10428–10436, 2020. 1

[62] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Standalone self-attention in vision models. *arXiv preprint arXiv:1906.05909*, 2019. 1

[63] Yongming Rao, Wenliang Zhao, Zheng Zhu, Jiwen Lu, and Jie Zhou. Global filter networks for image classification. *arXiv preprint arXiv:2107.00645*, 2021. 3

[64] Yongming Rao, Wenliang Zhao, Yansong Tang, Jie Zhou, Ser Nam Lim, and Jiwen Lu. Hornet: Efficient high-order spatial interactions with recursive gated convolutions. *Advances in Neural Information Processing Systems*, 35: 10353–10366, 2022. 7

[65] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020. 8

[66] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1, 2

[67] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16519–16529, 2021. 1

[68] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 2

[69] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[70] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017. 2

[71] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019. 1

[72] Sean J Taylor and Benjamin Letham. Forecasting at scale. *The American Statistician*, 72(1):37–45, 2018. 8

[73] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *ICCV*, 2019. 8

[74] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 1

[75] Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. In *European Conference on Computer Vision*, pages 516–533. Springer, 2022. 2, 7

[76] Shikhar Tuli, Ishita Dasgupta, Erin Grant, and Thomas L Griffiths. Are convolutional neural networks or transformers more like human vision? *arXiv preprint arXiv:2105.07197*, 2021. 2

[77] Pavan Kumar Anasosalu Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, and Anurag Ranjan. Fastvit: A fast hybrid vision transformer using structural reparameterization. *arXiv preprint arXiv:2303.14189*, 2023. 2, 7

[78] Ashish Vaswani, Prajit Ramachandran, Aravind Srinivas, Niki Parmar, Blake Hechtman, and Jonathon Shlens. Scaling local self-attention for parameter efficient visual backbones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12894–12904, 2021. 1

[79] Guangzhi Wang, Yixiao Ge, Xiaohan Ding, Mohan Kankanhalli, and Ying Shan. What makes for good visual tokenizers for large language models? *arXiv preprint arXiv:2305.12223*, 2023. 8

[80] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv preprint arXiv:2102.12122*, 2021. 1

[81] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022. 7

[82] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14408–14419, 2023. 6, 7

[83] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *TOG*, 2019. 8

[84] Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018. 8

[85] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16133–16142, 2023. 2, 4, 5, 7

[86] Haixu Wu, Hang Zhou, Mingsheng Long, and Jianmin Wang. Interpretable weather forecasting for worldwide stations with a unified deep model. *Nature Machine Intelligence*, pages 1–10, 2023. 6, 7, 8

[87] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *CVPR*, 2019. 8

[88] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, 2015. 8

[89] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 418–434, 2018. 4, 6

[90] Chengxing Xie, Xiaoming Zhang, Linze Li, Haiteng Meng, Tianlin Zhang, Tianrui Li, and Xiaole Zhao. Large kernel distillation network for efficient single image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1283–1292, 2023. 1, 3

[91] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. *arXiv preprint arXiv:2111.11418*, 2021. 2

[92] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018. 1

[93] Yiyuan Zhang, Kaixiong Gong, Kaipeng Zhang, Hongsheng Li, Yu Qiao, Wanli Ouyang, and Xiangyu Yue. Meta-transformer: A unified framework for multimodal learning. *arXiv preprint arXiv:2307.10802*, 2023. 2, 6, 8

[94] Yiyuan Zhang, Xiaohan Ding, Kaixiong Gong, Yixiao Ge, Ying Shan, and Xiangyu Yue. Multimodal pathway: Improve transformers with irrelevant data from other modalities. *arXiv preprint arXiv:2401.14405*, 2024. 2

[95] Yiyuan Zhang, Yuhao Kang, Zhixin Zhang, Xiaohan Ding, Sanyuan Zhao, and Xiangyu Yue. Interactivevideo: User-centric controllable video generation with synergistic multimodal instructions. *arXiv preprint arXiv:2402.03040*, 2024. 8

[96] Zhixin Zhang, Yiyuan Zhang, Xiaohan Ding, Fusheng Jin, and Xiangyu Yue. Online vectorized hd map construction using geometry. *arXiv preprint arXiv:2312.03341*, 2023. 8

[97] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 1

[98] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019. 2, 6