



REPVSR: Efficient Video Super-Resolution via Structural Re-Parameterization

KunLei Hu¹ ^a and Dahai Yu² ^b

¹Tetras.Ai, China

²TCL Corporate Research (HK) Co., Ltd, China

Keywords: Video Super-Resolution, Re-Parameterization Method, Efficiency Network.

Abstract: Recent advances in video super-resolution (VSR) explored the power of deep learning to achieve a better reconstruction performance. However, the high computational cost still hinders it from practical usage that demands real-time performance (24 fps). In this paper, we propose a re-parameterization video super-resolution (REPVSR) to accelerate the reconstruction speed with efficient and generic network. Specifically, we propose re-parameterizable building blocks, namely **Super-Resolution Multi-Branch** block (SRMB) for efficient SR part design and **FlowNet Multi-Branch** block (FNMB) for optical flow estimation part. The blocks extract features in multiple paths in the training stage, and merge the multiple operations into one single 3×3 convolution in the inference stage. We then propose an extremely efficient VSR network based on SRMB and FNMB, namely REPVSR. Extensive experiments demonstrate the effectiveness and efficiency of REPVSR.

1 INTRODUCTION

Video super-resolution (VSR) is developed from single image super-resolution, it aims to generate a high-resolution (HR) video from its corresponding low-resolution (LR) observation by filling in missing details, trying to restore the definition of video and improve the subjective visual quality. Thanks to deep learning, VSR based on neural networks experienced significant improvements over the last few years. However, the main research directions (Wang et al., 2019; Chan et al., 2021; Liu et al., 2021) lie in the pursuit of high fidelity scores by employing a very deep and complicated network structure, ignoring computational efficiency and memory constraints.


In order to deploy VSR models on resource-limited devices, latest research demonstrated meaningful advances in terms of lightweight model structure design (Xia et al., 2023; Fuoli et al., 2023), models with fewer FLOPs may have even larger latency because of the deployment of hardware-unfriendly operators (Wang et al., 2019), some tiny VSR models such as VESPCN (Caballero et al., 2017) can reach nearly real-time speed, in the meantime, their VSR performance measured by PSNR is quite limited.


Thus, model parameters reduction and hardware-friendly operators design have attracted more and more attention. It is always challenging to design both light-weight and inference efficient VSR model due to the very limited hardware resources, but along with growing commercial and industrial demand, it is also very necessary to design a lightweight VSR model with fewer parameters and efficient structures.

In this paper, inspired by Ding *et al.* (Ding et al., 2021b; Ding et al., 2022; Zhou et al., 2023), We propose a rigorous and effective framework SRMB and FNMB that is theoretically verified and experimentally validated. Based on SRMB and FNMB structure, we further propose recurrent VSR network (REPVSR) using super-light model design and re-parameterization technique to accelerate the inference speed and enhance reconstructive quality. The contributions of this study are as listed:

(1) The SRMB and FNMB blocks proposed in this paper can be used to improve the super resolution performance and optical flow estimation results respectively, without introducing any extra burden on inference or deployment.

(2) We propose a super efficient and lightweight VSR model termed REPVSR by embedding the SRMB and FNMB blocks into recurrent end-to-end trainable VSR framework. Extensive experiments and comparisons validate the computational efficiency

^a  <https://orcid.org/0009-0005-1309-0951>

^b  <https://orcid.org/0000-0003-1427-8807>

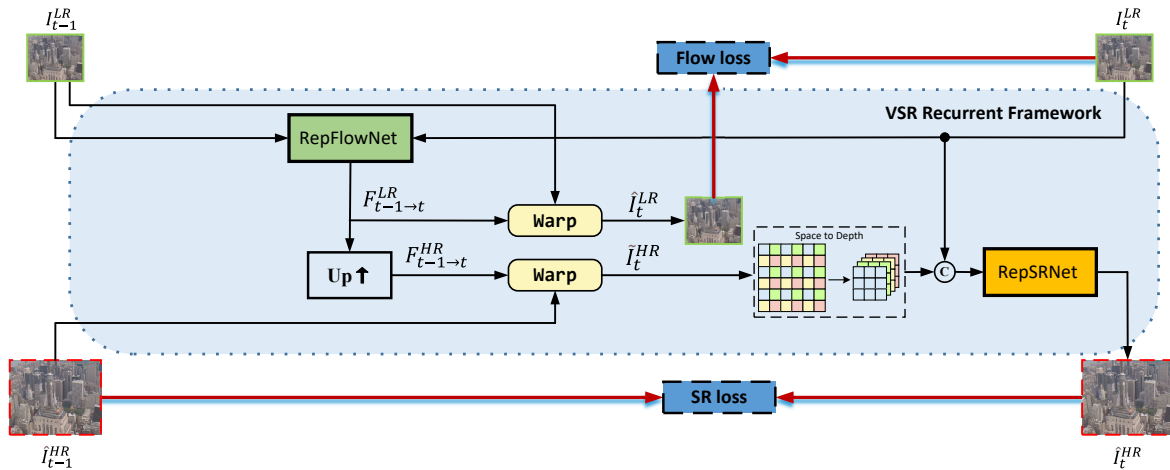


Figure 1: Overall of the RepVSR model. In the figure, green rectangles and red dash-lined rectangles represent the LR input frames and HR predicted frames, respectively.

and effectiveness of our proposed REPVSR network, which surpass recent re-parameterization schemes and lightweight VSR models, close to the large parameter model.

2 RELATED WORK

2.1 Deep-Learning Based Video Super-Resolution

Recently, deep-learning based VSR algorithms have risen rapidly. Existing VSR approaches can be mainly divided into sliding-window methods and recurrent methods. Sliding-window framework compute optical flows between multi-frames to aggregate information and perform spatial warping for alignment (Haris et al., 2019; Xue et al., 2019). Deformable convolution networks have been developed to address feature misalignment (Wang et al., 2019; Tian et al., 2020). Recurrent VSR structures can pass the previous HR estimate directly to the next step, recreating fine details and producing temporally consistent videos. FRVSR (Sajjadi et al., 2018) stores the HR estimate of the previous frame and uses it to generate the subsequent frame. Some bidirectional recurrences such as BasicVSR (Chan et al., 2021; Chan et al., 2022) can enforce the forward and backward consistency of the LR warped inputs and HR-predicted frames.

2.2 Structural Re-Parameterization Techniques

There are several studies on re-parameterization have shown their effectiveness on high-level vision tasks

such as image classification, object detection and semantic segmentation. DiracNet (Zagoruyko and Komodakis, 2017) builds deep plain models by encoding the kernel of convolution layers, getting comparable performance of ResNet. Related to DiracNet, RepVGG (Ding et al., 2021b) firstly proposed a structural re-parameterization technique. ACNet (Ding et al., 2019) and ExpandNet (Marnierides et al., 2018) can also be viewed as structural re-parameterization. Previous re-parameterization methods are mainly employed on high-level vision tasks and super-resolution tasks. In this paper, we embed re-parameterization mechanism into recurrent video super-resolution framework, proposing light-weight VSR model without introducing additional cost in the inference stage.

3 PROPOSED METHOD

Our REPVSR is based on recurrent framework as Figure 1 illustrates. Specifically, we employ re-parameterization mechanism to design optical flow estimation network (RepFlowNet) and super-resolution network (RepSRNet).

3.1 Multi-Branch Training Block

As Figure 2 (a) and (b) show, the design of RepSRNet follows residual architectures (Sajjadi et al., 2017) and RepFlowNet uses encoder-decoder style architecture. Inspired by Diverse Branch Block (DBB) (Ding et al., 2021a) which enhances the representational capacity of a single convolution by combining diverse branches of different scales and complexities, we in-

introduce SRMB and FNMB in this paper. Figure 2 (c) illustrates the architecture of SRMB and FNMB block which are summarized as follows:

Component I: Common 3×3 Convolution. A common 3×3 convolution $W_0 \in \mathbb{R}^{D \times C \times 3 \times 3}$ is employed to C-channel input $I \in \mathbb{R}^{C \times H \times W}$ to ensure the base performance. The bias B_0 is added onto the results of convolution. The convolution operation is formulated as:

$$O = W_0 * I + B_0 \quad (1)$$

Component II: A conv for Sequential Convolutions. We merge a sequence of 1×1 conv - 3×3 conv, 3×3 conv - 1×1 conv and 1×1 conv - 3×3 conv - 1×1 conv into one 3×3 conv as wider features can improve the expressions. Take the first sequence as example, $W^{(1)} \in \mathbb{R}^{C \times D \times 3 \times 3}$ and $W^{(2)} \in \mathbb{R}^{D \times C \times 1 \times 1}$ represent 1×1 and 3×3 convolution kernel respectively to expand and squeeze features. The feature is extracted as:

$$O' = W^{(2)} * (W^{(1)} * I + B^{(1)}) + B^{(2)} \quad (2)$$

The other two sequence of can be merged following the same mechanism detailed above.

Component III: A conv for Convolution with Laplacian. Since the Laplacian filter is useful for finding the fine details of a video frame (Jian et al., 2008), we first employ 1×1 conv (the weights and bias are W_l and B_l) and then use the Laplacian filter (denoted as D_{lap}) to extract spatial derivative (Zhang et al., 2021). The edge information feature is formulated as follows.

$$O_{lap} = (S_{lap} \cdot D_{lap}) \otimes (W_l * I + B_l) + B_{lap} \quad (3)$$

where S_{lap} and B_{lap} respectively represent scaling factors and bias of depth-wise convolution, and \otimes means depth-wise convolution (DWConv).

In general, the output of FNMB is the combination of the first two components and the output of SRMB is the combination of all three components.

3.2 Re-Parameterization for VSR Inference

We re-parameterize FNMB and SRMB into a single 3×3 convolution for efficient inference. The sequence of 1×1 conv - 3×3 conv in component II can be merged into one single normal convolution with parameters W_1, B_1 .

$$\begin{aligned} W_1 &= perm(W^{(1)}) * W^{(2)} \\ B_1 &= W^{(2)} * rep(B^{(1)}) + B^{(2)}, \end{aligned} \quad (4)$$

where $perm$ represents the permute operation and rep means using spatial transmission to replicate the bias to specified dimension. Similarly, the sequence

3×3 conv - 1×1 conv and 1×1 conv - 3×3 conv - 1×1 conv can be merged as W_2, B_2 and W_3, B_3 . As for component III that employ 1×1 conv and 3×3 DWConv, we have:

$$\begin{aligned} W_{lap}[i, i, :, :] &= (S_{lap} \cdot D_{lap})[i, 1, :, :] \\ W_{lap}[i, j, :, :] &= 0, i \neq j, \end{aligned} \quad (5)$$

where W_{lap} denotes the weight of convolution which is equal to DWConv and i, j represent the number of channel. Thus, the weights of FNMB after re-parameterization is:

$$W_{FNMB} = \sum_{i=0}^3 \{W_i\}, B_{FNMB} = \sum_{i=0}^3 \{B_i\} \quad (6)$$

and the weights of SRMB after re-parameterization is:

$$\begin{aligned} W_{SRMB} &= \sum_{i=0}^3 \{W_i\} + perm(W_l) * W_{lap}, \\ B_{SRMB} &= \sum_{i=0}^3 \{B_i\} + perm(B_l) * B_{lap} \end{aligned} \quad (7)$$

The output feature of the multibranch architecture can be obtained by using single normal convolution in inference time by re-parameterization technique.

3.3 Loss Function

As Figure 1 illustrates, there are two streams during training stage: the HR and LR frames. The loss on HR frames \mathcal{L}_{SR} is compute between the output of RepSR-Net and the HR frames. I_t^{HR} denotes the ground truth frame and \hat{I}_t^{HR} denotes the generated frame at time t . Since optical flow of our video dataset do not have ground truth, we utilize the warped LR frames from $t-1$ to t as the loss function of RepFlowNet \mathcal{L}_{Flow} . For each recurrent step, the SR loss and Flow loss are calculated as:

$$\mathcal{L}_{SR} = \|\hat{I}_t^{HR} - I_t^{HR}\|_2^2 \quad (8)$$

and also:

$$\mathcal{L}_{Flow} = \|\text{warp}(I_{t-1}^{LR}, F_{t-1 \rightarrow t}^{LR}) - I_t^{LR}\|_2^2. \quad (9)$$

Where $\text{warp}(\cdot)$ represents warp operation. In all, the overall loss function for training are combined as:

$$\mathcal{L}_{total} = \mathcal{L}_{SR} + \mathcal{L}_{Flow} \quad (10)$$

4 EXPERIMENTS

4.1 Experiment Settings

4.1.1 Baseline Methods

The most popular dataset for testing is Vid4, more high-frequency details included than other datasets.

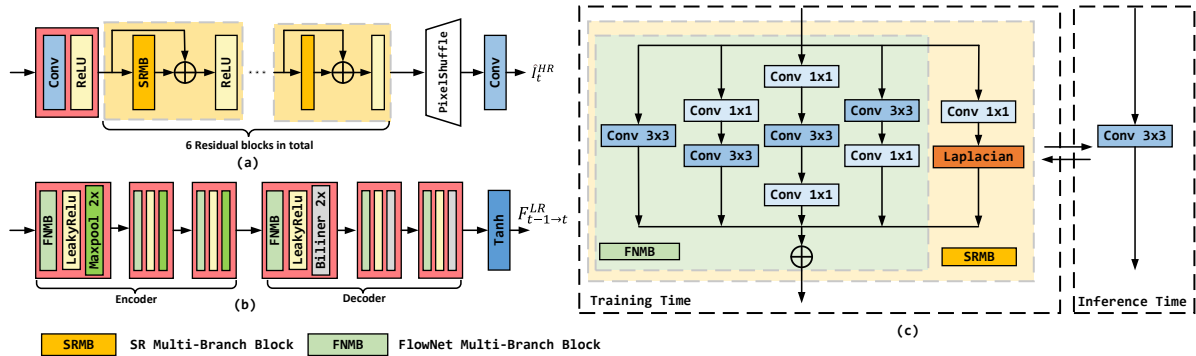


Figure 2: Network architectures for RepSRNet and RepFlowNet. The sub-figure (c) detail the re-parameterization block embedded in RepSRNet and RepFlowNet (SRMB and FNMB, respectively).

Table 1: Quantitative comparisons of several benchmark on PSNR|SSIM values.

Scale	Dataset	PSNR/SSIM						
		Bicubic	VESPCN	SOFVSR	FRVSR	TecoGAN	BasicVSR	REPVSR
x4	Vid4	23.53/0.628	25.35/0.756	26.01/0.772	26.69/0.822	25.89/0.737	27.24/0.825	26.85/0.817
	Vimeo-90k	31.32/0.868	33.55/0.907	34.89/0.923	35.64/0.932	34.27/0.925	37.18/0.945	35.62/0.928
x2	Set14	31.85/0.802	32.99/0.872	33.23/0.916	32.18/0.917	32.22/0.922	33.63/0.949	33.02/0.926
	Vimeo-90k	36.52/0.871	37.76/0.899	37.53/0.938	37.71/0.941	38.01/0.945	38.27/0.960	37.65/0.953

Thus, Vid4 is frequently used for evaluating the performance of VSR methods. Vimeo-90K and set15 includes videos with hard and real scenes, which is challenging for VSR methods. So we choose these three datasets as testing data in the following section.

Several DL-based methods are selected for comparison, including VESPCN(Caballero et al., 2017), SOFVSR(Wang et al., 2020), FRVSR(Sajjadi et al., 2018), TecoGAN(Chu et al., 2020). The reason for this selection is that we take the number of model parameters into consideration, the parameters of the selected models is similar to or larger than the model proposed in this paper. the BasicVSR(Chan et al., 2021) we chosen here is to verify the numerical metrics gap between our proposed method and leading large parameter model like BasicVSR.

4.1.2 Implementation Details

We conduct experiments on data captured from 40 high-resolution videos (720p, 1080p and 4K) downloaded from *vimeo.com*. We apply Gaussain blur with standard deviation $\sigma = 1.5$ to the HR frames and downsample them by $4\times$ to produce the input LR videos, also knows as Blur Down(BD). Our model is implemented with Pytorch framework on the PC with a single NVIDIA GeForce GTX 2080Ti GPU.

The Adam optimizer is used to train the network with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ with a basic learning rate of 0.0001 and it is decayed by 0.5 every 150000 iterations. We choose the size of the mini-batch as 4 and the total number of iterations as $4e5$.

4.2 Evaluation Results and Discussion

4.2.1 Quantitative Results and Qualitative Evaluations

As Table 1 shows, the quantitative metrics peak pixel-to-noise ratio (PSNR) and structural similarity (SSIM) are computed on RGB-channels for an objective assessment of VSR image quality in Vid4 datasets and Vimeo-90k test part in BD method.

(1) Compared with competitive lightweight VSR networks, our REPVSR obtains 0.46dB gain on Vid4 over FRVSR, and also has huge advantages over other compared models. Note that, different from original network FRVSR, we merely depoly SRMB and FNMB technique on the shortened backbone network, and we obtain superior performance while only consuming a fraction of FLOPs of the original FRVSR network.

(2) It is interesting that our small model, despite being much more efficient, gets very close re-



Figure 3: Qualitative comparison on Vid4.

Table 2: Comparison of running time (in seconds).

Method	Parameters(M)	Source	Target	FLOPs(G)	FPS (GPU)
VESPCN	0.879	320× 180	720p	96.56	48.48
		480× 270	1080p	221.08	24.76
		960× 540	4K	886.47	6.78
SOFVSR	1.640	320× 180	720p	226.12	13.31
		480× 270	1080p	508.78	5.993
		960× 540	4K	2035.11	1.73
FRVSR	2.589	320× 180	720p	190.81	31.16
		480× 270	1080p	429.30	15.10
		960× 540	4K	1718.65	3.76
TecoGAN	2.589	320× 180	720p	190.81	31.15
		480× 270	1080p	429.30	15.05
		960× 540	4K	1718.65	3.74
RepVSR	0.274	320× 180	720p	29.435	96.76
		480× 270	1080p	66.228	37.52
		960× 540	4K	264.926	14.36

sults compared to the much larger model, like BasicVSR on the validation datasets, demonstrating that our REPVSR method can make better use of the re-parameterized structure of the network and increases the efficiency of the learned network parameters.

From the objective results, the above quantitative evaluation is consistent with the qualitative evaluation show in Figure 3. We can see that our models are able to recover fine details and produce visually pleasing results. REPVSR achieves the most restoration ability while maintaining a slim framework.

4.2.2 Running Time Analysis

The running frame rates of different VSR models during inference stage will be presented in this part. The experimental results are shown in Table 2. The second column lists the parameters of each VSR model and column 5 counts the statistics of corresponding computation cost. The total computation cost required by our REPVSr during inference time is only 31.17% of VESPCN, 16.71% of SOFVSr, and 10.58% of FRVSr and TecoGAN. Not to mention REPVSr, which has a very large parameter of 338.5G FLOPS. The last columns illustrate the average FPS in different resolutions, When generating 1080p definition video, the proposed method can run in real time on NVIDIA Geforce GTX 1080 level graphics cards. Due to the implementation of structural re-parameterization, our REPVSr model runs two times and even much more faster on GPU platform compared with other deep models.

5 CONCLUSION

In this paper, we design a recurrent VSR network based on re-parameterization (REPVSr) to re-parameterize models with a multi-branch design. The positive results show favorable speed-accuracy trade-off compared to existing VSR models. In the future, we aim to embed re-parameterization mechanism to other efficient VSR architecture.

REFERENCES

- Caballero, J., Ledig, C., Aitken, A., Acosta, A., Totz, J., Wang, Z., and Shi, W. (2017). Real-time video super-resolution with spatio-temporal networks and motion compensation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4778–4787.
- Chan, K. C., Wang, X., Yu, K., Dong, C., and Loy, C. C. (2021). Basicvsr: The search for essential components in video super-resolution and beyond. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4947–4956.
- Chan, K. C., Zhou, S., Xu, X., and Loy, C. C. (2022). Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5972–5981.
- Chu, M., Xie, Y., Mayer, J., Leal-Taixé, L., and Thurey, N. (2020). Learning temporal coherence via self-supervision for gan-based video generation. *ACM Transactions on Graphics (TOG)*, 39(4):75–1.
- Ding, X., Chen, H., Zhang, X., Huang, K., Han, J., and Ding, G. (2022). Re-parameterizing your optimizers rather than architectures. *arXiv preprint arXiv:2205.15242*.
- Ding, X., Guo, Y., Ding, G., and Han, J. (2019). Acnet: Strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1911–1920.
- Ding, X., Zhang, X., Han, J., and Ding, G. (2021a). 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.
- Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., and Sun, J. (2021b). Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13733–13742.
- Fuoli, D., Danelljan, M., Timofte, R., and Van Gool, L. (2023). Fast online video super-resolution with deformable attention pyramid. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1735–1744.
- Haris, M., Shakhnarovich, G., and Ukita, N. (2019). Recurrent back-projection network for video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3897–3906.
- Jian, S., Xu, Z., and Shum, H. Y. (2008). Image super-resolution using gradient profile prior. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, 24-26 June 2008, Anchorage, Alaska, USA.
- Liu, H., Zhao, P., Ruan, Z., Shang, F., and Liu, Y. (2021). Large motion video super-resolution with dual subnet and multi-stage communicated upsampling. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 2127–2135.
- Marnerides, D., Bashford-Rogers, T., Hachett, J., and DeBattista, K. (2018). Expandnet: A deep convolutional neural network for high dynamic range expansion from low dynamic range content. In *Computer Graphics Forum*, volume 37, pages 37–49. Wiley Online Library.
- Sajjadi, M. S., Scholkopf, B., and Hirsch, M. (2017). Enhancenet: Single image super-resolution through automated texture synthesis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4491–4500.
- Sajjadi, M. S., Vemulapalli, R., and Brown, M. (2018). Frame-recurrent video super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6626–6634.
- Tian, Y., Zhang, Y., Fu, Y., and Xu, C. (2020). Tdan: Temporally-deformable alignment network for video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3360–3369.
- Wang, L., Guo, Y., Liu, L., Lin, Z., Deng, X., and An, W. (2020). Deep video super-resolution using hr optical

- flow estimation. *IEEE Transactions on Image Processing*, 29:4323–4336.
- Wang, X., Chan, K. C., Yu, K., Dong, C., and Change Loy, C. (2019). Edvr: Video restoration with enhanced deformable convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0.
- Xia, B., He, J., Zhang, Y., Wang, Y., Tian, Y., Yang, W., and Van Gool, L. (2023). Structured sparsity learning for efficient video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22638–22647.
- Xue, T., Chen, B., Wu, J., Wei, D., and Freeman, W. T. (2019). Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127:1106–1125.
- Zagoruyko, S. and Komodakis, N. (2017). Diracnets: Training very deep neural networks without skip-connections. *arXiv preprint arXiv:1706.00388*.
- Zhang, X., Zeng, H., and Zhang, L. (2021). Edge-oriented convolution block for real-time super resolution on mobile devices. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 4034–4043.
- Zhou, D., Gu, C., Xu, J., Liu, F., Wang, Q., Chen, G., and Heng, P.-A. (2023). Repmode: Learning to reparameterize diverse experts for subcellular structure prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3312–3322.

