TINY AUDIO SPECTROGRAM TRANSFORMER: MOBILEVIT FOR LOW-COMPLEXITY ACOUSTIC SCENE CLASSIFICATION WITH DECOUPLED KNOWLEDGE DISTILLATION

Technical Report

Jinyang Yu[1], Zikai Song[2,3], Jiahao Ji[2,3], Lixian Zhu[2,3], Kele Xu[1], Kun Qian[2,3†], Yong Dou[1*], and Bin Hu[2,3†]*

[1] National University of Defense Technology, Computer Dept., Changsha, P.R. China,
yujinyang@nudt.edu.cn, xukelele@163.com, yongdou@nudt.edu.cn
[2] Key Laboratory of Brain Health Intelligent Evaluation and Intervention,
Ministry of Education (Beijing Institute of Technology), P. R. China,
[3] School of Medical Technology, Beijing Institute of Technology, P. R. China,
{songzk, jiahao.ji, zhulx17, qian, bh}@bit.edu.cn

ABSTRACT

This report presents BIT&NUDT submissions to DCASE2023 challenge Task1[1], which aims to acoustic scene classification (ASC) with low complexity. Several vision transformers adapted to audio classification tasks[2, 3]have been proved to be more robust than CNNs due to their global representations. However, considering the complexity of self-attention, they seem not fit for light-weight edge devices. In our submission, we transfer a light-weight vision transformer, MobileViT[4] from image tasks to ASC. By inserting the MobileViT block into CNN, our network can benefit from both attention global representations and CNN spatial representations. Under the parameter memory limitation of 128KB, we make quantization and convert a part of the parameters to INT8 for balance between complexity and accuracy. Further more, we use Decoupled Knowledge Distillation[5] to take advantage of PaSST[3] teacher models which outperformed in previous DCASE challenge.

Index Terms— Acoustic Scene Classification, Transformer, MobileViT, Decoupled Knowledge Distillation, Low-complexity

1. INTRODUCTION

Acoustic scene classification (ASC) is to classify where the given audio probably occurs. For example, giving an audio describing airplane engine and people talking, we estimate it occurs in an airport. In DCASE2023 Task1, the classification challenge requires device-robustness and low-complexity for edge devices like Cortex-M4. The dataset is TAU Urban Acoustic Scenes 2022 Mobile[6], which pre-defines acoustic scenes to 10 classes. However, it is unbalanced on audio recording device, with the difficulty of unseen devices in evaluation set. For the system complexity requirement, it differs from 2022's that the limitation of 128KB is imposed on parameter memory use but not parameter counts, which means less amount of available parameters in higher numerical representations.

In common methods to extract audio features, a sound wave is converted to a spectrogram which can be regarded as an image. This makes vision transformers can be used in audio classification and achieve excellent performance as well as in image classification. Owing to self-attention, transformer aggregates global representations and take maximum use of the input information. However, this is at the cost of parameter amount. MobileViT is a light-weight vision transformer that fuses features from transformers and CNNs, leading to lower complexity and the combination of both advantages. MobileViT blocks are inserted after convolution layers, fusing the local CNN representations and global transformer representations without losing CNN's inductive bias. Thus, we design our ASC models based on MobileViT. After reduction and quantization, they can be compressed under 128KB in total parameter memory use, and 30M in MACs (multiply-accumulate operations).

Inspired by previous DCASE participants, we adopt Knowledge Distillation to transfer a complex teacher's domain knowledge to our light-weight model for better accuracy. Decoupled Knowledge Distillation[5] offers more flexible control of training than the classical[7] by reconstructing the loss into two parts. We experiment on PaSST, which outperformed in last DCASE challenge, and make it as a teacher to optimize our model.

2. DATA PROCESSING

The dataset TAU Urban Acoustic Scene 2022 Mobile contains recordings from 12 European cities in 10 different acoustic scenes using 4 different devices in 44.1kHz. Additionally, synthetic data for 11 mobile devices was created based on the original recordings. All audio segments are 1s which cut from previous 10s segments. In development set, it contains 64 hours audio records from 9 devices, and only 6 devices in training segments. The evaluation set contains data from 12 cities, 10 acoustic scenes, 11 devices, where 5 new devices are unseen in the development set. The training set is unbalanced due to 70% of segments are from device A. To guarantee device-robustness, we make a two-stage data processing, for the waveforms and the extracted spectrograms.

*This work was partially supported by National Key Laboratory of Parallel and Distributed Computing (PDL) of National University of Defense Technology, Changsha, China. *(Corresponding author: K. Xu and Y. Dou)*

†This work was partially supported by the Ministry of Science and Technology of the People's Republic of China with the STI2030-Major Projects (No. 2021ZD0201900), the National Natural Science Foundation of China (No. 62227807 and 62272044), the Teli Young Fellow Program from the Beijing Institute of Technology, China, and the BIT Research and Innovation Promoting Project (Grant No. 2022YCXZ012), China. (*Corresponding authors: K. Qian and B. Hu.)*

2.1. Waveform Processing

We reassemble the original 1s segments to 10s, and randomly cut slices of 1s for each training epoch, as supposed in [8]. Each 1s audio is down-sampled to 32kHz, pre-emphasized, and time-shifted with range $\pm$0.5s. We experiment on adopting pitch shifting and adding Gauss noise to the audio wave. In training step, we also apply mixup[9] to the waveforms.

2.2. Spectrogram Processing

The preprocessed audio waveforms are transformed to spectrograms. We use a STFT window size 2000, hop size 500 to extract linear spectrogram, and apply Mel-scaled filter bank of 256 frequency bins on it. After log operation and normalization, the output spectrogram is in shape 256$\times$64. We make this shape for alignment with our network input. To make augmentation we experiment with time-masking and frequency-masking. In order to enhance device generalization capability, we also experiment with spectrogram cutmix[10] and mixstyle[11], and choose frequency-wise mixstyle to process the unbalanced device information.

3. MODEL ARCHITECTURE

Self-attention spectrogram networks originated from vision transformers have been proved to be robust in different tasks recently[2, 3, 12, 13, 14]. Unlike image tasks, an audio CNN has to be designed with a suitable receptive filed (RF)[15], that is fixed by the model structure. Benefit from self-attention, a transformer offers learnable receptive filed[16], which we intuit it brings robustness on unseen devices. However, it seems unconventional to use a transformer in resource-constrained circumstances due to its high complexity. In this section, we present our model based on MobileViT[4], a light-weight vision transformer, which makes it possible to introduce self-attention into small networks.

3.1. MobileViT

MobileViT is designed for edge devices which achieves considerable accuracy and generalization capability under strict parameter limitation. It consists of both convolution layers and transformer layers. In a MobileViT block, convolution layers offer spatial-aware local representations with inductive bias for images. These local representations are split into multiple patches, and unfolded for a transformer to encode global representations like classical ViT[17] does. There is no need for positional encoding since the global representations are folded back to a feature map later, which will make an concatenation with the previous CNN local representations. The concatenated features are fused via another following convolution layer. To reduce the parameters and MACs, MobileViT blocks are inserted after light-weight CNN blocks for down-sampling, e.g. Inverted Residual blocks used in MobileNetv2[18]. In MobileViTv3 block[19] it applies depthwise separable convolutions and 1×1 convolutions for further reduction of the parameters, and adds a shortcut from input to the fused representation.

3.2. Transfer from Image to Audio

To adapt MobileViT block and MobileViTv3 block to audio task, we use Log Mel spectrogram as the input. The shape should be fit for unfolding and folding. Original image input of MobileViT is $3 \times 256 \times 256$, with height and width both a power of 2, that means the input can be always 2×2 patched when it is repeatedly down-sampled by half. Considering the short input in our task, we set time dimension to 64. And the channel is set to 1 for mono recordings. Thus, the input shape is $1 \times 256 \times 64$ for channel, frequency and time. In the blocks, we use default configuration of convolution kernel size $(n, n) = (3, 3)$, and patch size $(h, w) = (2, 2)$ as they satisfy $h, w \leq n$ which means each pixel can make use of all other pixel information. We named the transferred model as MobileAST.

3.3. Model Architecture

We compare MobileViT block with MobileViTv3 block in MobileAST, and apply MobileViTv3 block which provides less parameters as shown in Table 1. The convolution feature is concatenated with attention feature and a residual connection from input to output is applied. We also cut the channels and transformer encoding dimensions to make a light version in Table 2. The input spectrogram is down-sampled with a regular convolution layer first, and the MobileNetv2 blocks are used, expanding channels to get ready for MobileViTv3 block feature fusion. We adjust the transformer depth D, attention layer dim d_l, MLP head dim d_m to control the contribution of MViTv3 block to the whole network. Also, we set channel width c_1, c_2, c_{out} and expansion factor e_1, e_2, e_3 to control the CNN part. At the end, the average-pooling connects the final convolution layer and the fully connected layer instead of flatten. This leads to fewer neurons in the fully connected layer. Additionally, we compare the activation SiLU[20] and ReLU, one is used in original MobileViT to avoid low-dimensional information loss and the other brings simplicity since ReLU operator can be fused during quantization, which means larger c_{out} can be used for final classification at a constrained model size.

Block	K	C	S	E	O
Input	-	1	-	-	$1 \times 256 \times 64$
In_Conv	3×3	32	2	-	$32 \times 128 \times 32$
MV2	3×3	c_1	1	e_1	$c_1 \times 128 \times 32$
MV2	3×3	c_2	2	e_2	$c_2 \times 64 \times 16$
MaxPool	2×1	-	(2,1)	-	$c_2 \times 32 \times 16$
MViTv3	3×3	c_2	-	-	$c_2 \times 32 \times 16$
MV2	3×3	64	2	e_3	$c_2 \times 32 \times 16$
Out_Conv	1×1	c_{out}	1	-	$c_{out} \times 16 \times 8$
AvgPool	16×8	-	-	-	$c_{out} \times 1 \times 1$
Linear	-	-	-	-	10

Table 1: MobileAST_Light Architecture, where K = kernel size of convolution layers, C = channels, S = strides, O = output shape of the layer, MV2 = MobileNetv2 Blocks, E = expansion factors of MV2, and MViTv3 = MobileViTv3 Blocks.

4. DECOUPLED KNOWLEDGE DISTILLATION

We adopt the Decoupled Knowledge Distillation to significantly reduce the complexity and computation of the model under the premise of ensuring the classification accuracy. In the traditional knowledge distillation method, as shown in Figure 1, logits are the output of the fully connected layer of the neural network. The logits from the teacher network and the student network are not distinguished, and the KL divergence of the two networks is directly calculated. The formula of the KL divergence is as follows:

Layer	K/D_L	C/H	G/D_M	O
Input	-	-	-	$c_2 \times 32 \times 16$ (1)
Conv	3×3	c_2	c_2	$c_2 \times 32 \times 16$
Conv	1×1	d_l	1	$d_l \times 32 \times 16$ (2)
Unfold	-	-	-	$4 \times 128 \times c_2$
Transformer	d_l	4	d_m	$4 \times 128 \times c_2$
Fold	-	-	-	$c_2 \times 32 \times 16$
Conv	1×1	c_2	1	$c_2 \times 32 \times 16$
Cat with (2)	-	d_l+c_2	-	$c_2 \times 32 \times 16$
Conv	1×1	c_2	1	$c_2 \times 32 \times 16$
Add with (1)	-	-	-	$c_2 \times 32 \times 16$

Table 2: MobileViTv3 Block in our acoustic model, where K = kernel size of convolution layers, C = channels, G = groups, O = output shape of the operation. In transformer layer, D_L represents attention layer dim, H the heads of attention which we fix it to 4, and D_M the MLP feed-forward layer dim. In Unfold, the feature map is reshaped into $(h * w) \times (\lceil F/h \rceil * \lceil T/w \rceil) \times d_l$ where we use patch size $h = w = 2$.

$$D_{KL}(p\|q) = \sum_i p_i \log \frac{p_i}{q_i} \tag{1}$$

Where, i represents the discrete value of the probability distribution, and p_i and q_i represent the probability value of class i predicted by the teacher model and the student model respectively.

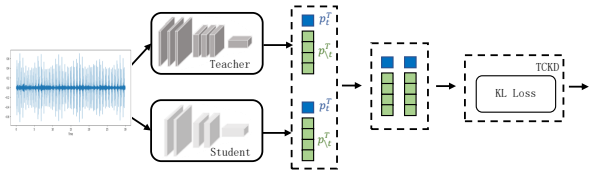

Figure 1: Traditional knowledge distillation methods.

This method limits the student's performance because it is inhibited by the teacher's confidence. The Decoupled Knowledge Distillation (DKD) method adopted in this work splits the classification output of the network prediction into target class and non-target class, and calculates the KL divergence of the two parts respectively, as shown in Figure 2.

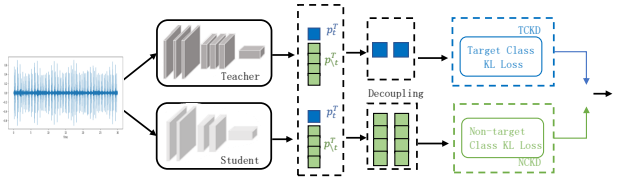

Figure 2: Decoupled knowledge distillation methods.

To split the logits of the network output, new probability distributions are defined: the binary probabilities of the target class and all the other non-target classes, p_t and $p_{\setminus t}$, and independently model probabilities among non-target classes $\hat{p}_i$. T and S denote the teacher model and the student model, respectively. The knowledge distillation loss function (KD_Loss) is defined as follows.

α	β	T	Validation Acc%
1	8	1	56.59
1	8	2	57.27
1	8	4	56.80
5	8	10	56.48
5	**10**	**4**	**57.79**
0.5	10	4	55.17

Table 3: DKD experiments on a MobileAST_Light baseline model with validation accuracy on TAU Urban Acoustic Scenes 2022 Mobile Development Set evaluation fold. The baseline model uses $c_1, c_2, c_3 = (64, 64, 160)$, $e_1, e_2, e_3 = (1, 1, 1)$ with SiLU activations.

$$\begin{aligned} KD\_Loss &= p_t^T log(\frac{p_t^T}{p_t^S}) + p_{\setminus t}^T log(\frac{p_{\setminus t}^T}{p_{\setminus t}^S}) + p_{\setminus t}^T \Sigma_{i=1,i\neq t}^C \hat{p}_i^T log(\frac{\hat{p}_i^T}{\hat{p}_i^S}) \\ &= KL(b^T||b^S) + (1 - p_t^T) KL(p^T||p^S) \\ &= TCKD + \left(1 - p_t^T\right) NCKD \end{aligned} \tag{2}$$

The KD_loss is reformulated into a weighted sum of two terms, TCKD (Target Class Knowledge Distillation, TCKD) and NCKD (Non-Target Class Knowledge Distillation, NCKD). Obviously, the weight of NCKD is coupled with p_t^T.

DKD loss function is composed of TCKD and NCKD, and the weights of both are set separately,

$$DKD\_Loss = \alpha TCKD + \beta NCKD \tag{3}$$

α and β in Equation 3, and the temperature T as in the common knowledge distillation are hyperparameters to be optimized. For teacher, we use PaSST as the teacher model, which is pretrained on audioset and retrained on TAU Urban Acoustic Scenes 2022 Mobile Development Set. We use a baseline version of MobileAST_Light as the student model to search for best distillation strategy as shown in Table 3.

5. QUANTIZATION

We choose post training static quantization (PTSQ) based on Pytorch FX Graph Mode to partly quantize the parameters into INT8 for balance between accuracy and model size. We fuse all convolution parts of the model, and also fuse linear layer with ReLU in transformer part. We use the fbgemm back-end to insert per channel MinMax observer for weights and histogram observer for activations. And a subset of the Development Set training fold is used for calibration. After quantization, the total parameter memory use can be compressed into under 128KB without significant performance loss.

6. EXPERIMENTAL SETUP

We trained our models all on the training fold of TAU Urban Acoustic Scenes 2022 Mobile Development Set. Each epoch the input audio segments are 1s uniformly sampled from the reassembled 10s. For teacher model PaSST we use the pretrained weights on audioset and fine-tune it on the development set. Since the number of classes changed from 527 to 10, we make average-pooling to the weights for corresponding layers. The fine-tuning setup is similar to [8],

Num.	c_1, c_2, c_{out}	e_1, e_2, e_3	d_l, d_m, D	Act.	Params, MACs, Memory	Validation Accuracy
1	(64, 64, 160)	(1, 1, 1)	(32, 64, 2)	SiLU	52.288 K, 23.804 M, 125.885 K	60.82, 58.95
2	(32, 32, 224)	(2, 1, 2)	(32, 64, 2)	ReLU	51.648 K, 28.400 M, 123.654 K	61.34, 60.60
3	(32, 32, 192)	(1, 1, 1)	(48, 64, 2)	ReLU	57.392 K, 13.403 M, 125.650 K	60.58, 59.65
4	(50, -, -)	(1, -, -)	(64, 64, 2)	ReLU	66.114 K, 11.879 M, 125.057 K	61.61, 61.30

Table 4: Submissions using different versions of MobileAST_Light, with complexity after quantization and validation accuracy before and after quantization.

namely patchout=6 on frequency, with stride=10 on both frequency and time. Specaugment masks on frequency and time are used with parameter 48 and 20. In the training process, we use Adam optimizer with 1e-3 weight decay. The learning rate warm-up to 1e-5 in the first 30 epochs and cosine-anneal to 1e-7 until epoch 250. The teacher is trained over NVIDIA Geforce RTX 2080Ti. For student models we use the same optimizer, and the training lasts for 600 epochs with learning rate increased to 1e-3 exponentially in first 30 epochs and decreased linearly to 1e-5 until epoch 400. The distillation of submitted models is executed over NVIDIA Geforce RTX 3090.

7. SUBMISSIONS

We make 4 submissions with different hypertuning of MobileAST_Light as shown in Table 4. In these configurations we apply different CNN or transformer proportion of the whole network, and all of them achieve over 60% accuracy on validation data (before quantization). We run them on TAU Urban Acoustic Scenes 2023 Mobile Evaluation dataset and generate submission results. The details for each configuration are following:

- **Submission 1** uses 64-channel convolution feature maps for global fusion with SiLU activation. The MV2 blocks do not use expansion. The transformer part has 16,896 parameters, 32.3% of the total. The parameter memory after quantization is 125.885 K. In distillation, we use $\alpha = 5, \beta = 10, T = 4$. We also apply mixup with $\alpha = 0.3$ on input waveforms and frequency-wise mixstyle with $\alpha = 0.3, p = 0.6$ after spectrum extracting.
- **Submission 2** uses 32-channel convolution feature maps for global fusion with ReLU activation. The first and the last MV2 blocks use expansion=2, this increases some MACs. The transformer part stay the same with submission 1, accounts for 32.7% of the total. After quantization the parameter memory is 123.654 K. In distillation, we use $\alpha = 5, \beta = 10, T = 4$. We also apply mixup with $\alpha = 0.3$ on input waveforms and frequency-wise mixstyle with $\alpha = 0.3, p = 0.5$ after spectrum extracting.
- **Submission 3** uses 32-channel convolution feature maps for global fusion with ReLU activation. We increase the transformer attention layer dim to 48, making the transformer part contributes 31,424 parameters, 54.8% of the total. The MV2 blocks also do not use expansion. After quantization, the memory is 125.650 K, similar to the previous submissions, but the MACs is down to 13.403 M. The mix and distillation configuration stay the same with Submission 1.
- **Submission 4** uses 50-channel convolution feature maps for global fusion with ReLU activation. The attention layer dim of transformer expands to 64, leading to 75.7% of the network is contributed by transformer. We have to cut more CNN parts under complexity limitation. The In_Conv layer is changed to 5×5 for larger receptive field since there remains only one MV2 block in the whole net work, which has $c_1 = 50$ following the In_Conv. After MViTv3 block the features are directly sent to AvgPool layer and the final classification layer. This submission has 66.114 K parameters as the most one, but is able to quantize to 125.057 K in memory, and MACs is only down to 11.897 M since convolutions are less. The mix and distillation configuration stay the same with Submission 2.

8. REFERENCES

[1] I. Martín-Morató, F. Paissan, A. Ancilotto, T. Heittola, A. Mesaros, E. Farella, A. Brutti, and T. Virtanen, "Low-complexity acoustic scene classification in dcase 2022 challenge," 2022. [Online]. Available: https://arxiv.org/abs/2206.03835

[2] Y. Gong, Y.-A. Chung, and J. Glass, "AST: Audio Spectrogram Transformer," July 2021, arXiv:2104.01778 [cs]. [Online]. Available: http://arxiv.org/abs/2104.01778

[3] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient Training of Audio Transformers with Patchout," in *Interspeech 2022*, Sept. 2022, pp. 2753–2757, arXiv:2110.05069 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2110.05069

[4] S. Mehta and M. Rastegari, "MobileViT: Light-weight, General-purpose, and Mobile-friendly Vision Transformer," Mar. 2022, arXiv:2110.02178 [cs]. [Online]. Available: http://arxiv.org/abs/2110.02178

[5] B. Zhao, Q. Cui, R. Song, Y. Qiu, and J. Liang, "Decoupled knowledge distillation," in *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 2022, pp. 11 953–11 962.

[6] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in dcase 2020 challenge: generalization across devices and low complexity solutions," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, 2020, pp. 56–60. [Online]. Available: https://arxiv.org/abs/2005.14623

[7] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," Mar. 2015. [Online]. Available: https://arxiv.org/abs/1503.02531v1

[8] F. Schmid, S. Masoudian, K. Koutini, and G. Widmer, "CP-JKU submission to dcase22: Distilling knowledge for low-complexity convolutional neural networks from a patchout audio transformer," DCASE2022 Challenge, Tech. Rep., June 2022.

[9] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond Empirical Risk Minimization," Apr. 2018, arXiv:1710.09412 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1710.09412

[10] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features," Aug. 2019, arXiv:1905.04899 [cs]. [Online]. Available: http://arxiv.org/abs/1905.04899

[11] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "DOMAIN GENERALIZATION WITH MIXSTYLE," 2021.

[12] Y. Gong, C.-I. J. Lai, Y.-A. Chung, and J. Glass, "SSAST: Self-Supervised Audio Spectrogram Transformer," Oct. 2021. [Online]. Available: https://arxiv.org/abs/2110.09784v2

[13] S. Atito, M. Awais, W. Wang, M. D. Plumbley, and J. Kittler, "ASiT: Audio Spectrogram vIsion Transformer for General Audio Representation," Nov. 2022. [Online]. Available: https://arxiv.org/abs/2211.13189v1

[14] A. Baade, P. Peng, and D. Harwath, "MAE-AST: Masked Autoencoding Audio Spectrogram Transformer," Mar. 2022. [Online]. Available: https://arxiv.org/abs/2203.16691v1

[15] Q. Kong and W. Wang, "PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition," vol. 28, 2020.

[16] Y. Gong, S. Khurana, A. Rouditchenko, and J. Glass, "CMKD: CNN/Transformer-Based Cross-Model Knowledge Distillation for Audio Classification," Mar. 2022, arXiv:2203.06760 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2203.06760

[17] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE," 2021.

[18] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," Mar. 2019, arXiv:1801.04381 [cs]. [Online]. Available: http://arxiv.org/abs/1801.04381

[19] S. N. Wadekar and A. Chaurasia, "MobileViTv3: Mobile-Friendly Vision Transformer with Simple and Effective Fusion of Local, Global and Input Features," Oct. 2022, arXiv:2209.15159 [cs]. [Online]. Available: http://arxiv.org/abs/2209.15159

[20] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," 2017.