# Supplementary Materials

## FaceLiVT: Face Recognition Using Linear Vision Transformer with Knowledge Distillation

This supplementary material provides implementation specifics and extra quality comparisons to further illustrate our method's effectiveness and superiority over existing approaches. With these additional evaluations, we aim to offer a thorough analysis and underscore the benefits of our approach.

### A. Details of FaceLiVT Architecture

The macro-design approach of FaceLiVT is influenced by MetaFormer [1], utilizing two stacked residual blocks. The FaceLiVT model begins with a stem module, often featuring two $3 \times 3$ convolutions, each with a stride of 2. On the macro level, the design uses a token mixer block for spatial feature extraction, followed by a channel mixer block. Both blocks include a normalization layer and use either residual or skip connections to stabilize loss and improve training. Let $X_i, X_i'$, and $X_i'' \in \mathbb{R}^{H_i \times W_i \times C_i}$ denote the feature maps at stage $i$ with resolution $H_i \times W_i$ and $C_i$ channels, with additional operational details in Eq. (10).

$$\begin{aligned} X_i' &= X_i + TokenMixer(X_i), \\ X_i'' &= X_i' + ChannelMixer(X_i'), \end{aligned} \tag{1}$$

$TokenMixer(.)$ operator is set as either convolution mixer or self-attention (refer to Table I). $ChannelMixer(.)$ comprises an MLP network, implemented by two fully connected linear layers and one activation function, as described in Eq. (2):

$$MLP(X_i') = BN\Big(\sigma\big(BN(X_i' * W_e)\big) * W_r\Big), \tag{2}$$

where $W_e \in \mathbb{R}^{C_i \times rC_i}$ and $W_r \in \mathbb{R}^{rC_i \times C_i}$ represent the layer weights, $r$ denotes the fully connected layer's expansion ratio, usually set to 3. The activation function $\sigma$ is selected as $GELU(.)$.

**Reparameterize Batch Normalization.** In CNN facial recognition systems, convolutional layers are often paired with Batch Normalization (BN) layers [2], [3]. BN aids in speeding up convergence and minimizing overfitting during training. Nevertheless, it increases complexity and latency for inference. Therefore, BN is integrated into the preceding convolution layer, creating the FaceLiVT.

In a convolutional layer having a kernel size $K$, the weights $W$ are described by $W \in \mathbb{R}^{C_o \times C_i \times K \times K}$, and the bias $b$ by $b \in \mathbb{R}^D$. Here, $C_i$ and $C_o$ refer to the input and output channel dimensions, respectively. The convolution process on the feature $X \in \mathbb{R}^{N \times C_i \times H \times W}$ is succeeded by Batch Normalization (BN), which includes the accumulated mean $\mu$, accumulated standard deviation $\sigma$, feature scale $\gamma$, bias $\beta$, and the convolution operation $*$. This is further elaborated in Eq.(3).

$$BN(Conv(X)) = \gamma \frac{(W * X + b) - \mu}{\sigma} + \beta. \tag{3}$$

As convolutions followed by BN during inference are linear operations, these can be merged into a single convolution layer with integrated BN, represented by Eq. (4):

$$BNConv(x) = W' \cdot X + b', \tag{4}$$

where the transformed weight is $W' = W\frac{\gamma}{\sigma}$ and the adjusted bias is $b' = (b-\mu)\frac{\gamma}{\sigma} + \beta$. BN is merged into the preceding convolutional layer across all branches, leaving only convolution in the architecture.

**Reparameterized Depth-Wise Convolution Mixer (DWMix).** The concept of convolutional mixing was initially presented in ConvMixer [4]. For an input tensor $X_i$, the mixing block within the layer was formulated as

$$X_i' = X_i + BN(\sigma(DWC(X_i))), \tag{5}$$

where $\sigma$ denotes a non-linear activation function, and DWC is a DepthWise Convolutional layer. Although this configuration is proved quite effective, the authors of [5] modified the sequence of operations and omitted the non-linear activation. To improve the DWMix, we enhance it with a $1 \times 1$ DWC after $k \times k$ DWC, which can enhance learnability during training.

$$X_i' = X_i + \{BN(DWC_{k\times k}(X_i) + DWC_{1\times 1}(X_i))\}. \tag{6}$$

At inference time, reparameterizing the skip connection and the $1 \times 1$ DWC into a single depthwise convolutional layer reduces computational and memory demands, benefiting mobile devices.

**Multi Head Self Attention.** In vision transformers, the Multi-Head Self-Attention (MHSA) mechanism enables the model to assess token importance in a sequence for context and prediction. For an input sequence $X$ with $N$ tokens, MHSA calculates key $K$, query $Q$, and value $V$ via linear transformations, where $K, Q, V \in \mathbb{R}^{B \times H_e \times N \times C}$; $B$ is the batch size, $H_e$ is the number of heads, $N$ is the token count, and $C$ is the channel dimension. MHSA details are in Eq. (7).

$$MHSA(Q, K, V) = Concat(SA_0, .., SA_{H_e})W_o. \tag{7}$$

where $SA$ refers to the self-attention operation in each head. It calculates a weighted average of the values based on a similarity score between pairs of input tokens as follows:

$$SA = Softmax\left(\frac{QK^T}{\sqrt{C}}\right)V. \tag{8}$$

Given that $QK^T$ represents a matrix multiplication of size $N \times N$, this operation entails a high computational complexity of $\mathcal{O}(N^2)$.

**Model Variant.** Table I shows the detailed configuration of the FaceLiVT variant with the proposed MHLA and conventional Multi-Head Self Attention (MHSA). *Downspl* denotes the down-sampling blocks to down-sample the feature map spatially and embed the desired channel dimension. *Dim* denotes the feature map channel dimension at stage-i. *#Blocks* denotes the number of blocks (MLP + Token Mixer) in stage-i.

### B. Detail of Experiment

**Knowledge Distillation (KD).** We utilize an efficient training method using a more complex network, specifically the FaceLiVT-L (MHSA) variant as the "Teacher" and the FaceLiVT MHLA variant as the "Student." The distillation loss is characterized by the mean squared error between the output features of the teacher, $X_1$, and the student, $X_2$, as illustrated

TABLE I
ALL VARIANT FACELIVT MODEL CONFIGURATIONS. #BLOCKS DENOTES THE NUMBER OF FACELIVT BLOCKS.

| Stage | Size | Layer | FaceLiVT | | FaceLiVT-MHSA | | |
|---|---|---|---|---|---|---|---|
| | | | S | M | S | M | M+ |
| Stem | $112^2$ | Conv | $[3 \times 3$, Stride 2$] \times 2$ | | | | |
| | | Dims($C_i$) | 40 | 64 | 40 | 64 | 64 |
| 1 | $28^2$ | Mixer | DWMix $3 \times 3$ | | | | |
| | | #Blocks | 2 | 2 | 2 | 2 | 3 |
| 2 | $14^2$ | Downspl | DWMix $3 \times 3$, Stride 2 | | | | |
| | | Dim($C_i$) | 80 | 128 | 80 | 128 | 128 |
| | | Mixer | DWMix $3 \times 3$ | | | | |
| | | #Blocks | 4 | 4 | 4 | 4 | 3 |
| 3 | $7^2$ | Downspl | DWMix $3 \times 3$, Stride 2 | | | | |
| | | Dim ($C_i$) | 160 | 256 | 160 | 256 | 256 |
| | | Mixer | MHLA | | MHSA | | |
| | | #Blocks | 6 | 6 | 6 | 6 | 9 |
| 4 | $4^2$ | Downspl | DWMix $3 \times 3$, Stride 2 | | | | |
| | | Dim ($C_i$) | 320 | 512 | 320 | 512 | 512 |
| | | Mixer | MHLA | | MHSA | | |
| | | #Blocks | 2 | 2 | 2 | 2 | 3 |
| Classifier Head | | | Avg Pool, FC (512) | | | | |

in the equation below.

$$\mathcal{L}_{dis} = \frac{1}{N} \sum_{i=0}^{N} \sqrt{(x_1 - x_2)^2} \tag{9}$$

$$\mathcal{L} = \mathcal{L}_{cls} + \alpha \mathcal{L}_{dis}, \tag{10}$$

The total loss is the combination of CosFace [6] as the classification loss function and the distillation loss with $\alpha = 0.5$.

**Training Setup.** We adopt a training approach similar to [3], [7], [8] using Glint360K [9]. The AdamW optimizer [10] is applied with a polynomial decay learning rate schedule. The learning rate begins at $6 \times 10^{-3}$ and decreases to a minimum of $1 \times 10^{-5}$. A total batch size of 384 is employed across $3\times$ NVIDIA RTX-A6000 GPUs, with a weight decay of $1 \times 10^{-4}$. The model undergoes training for 20 epochs with a resolution of $(112 \times 112)$.

TABLE II
FACELIVT TRAINING HYPERPARAMETERS ON GLINT360K DATASETS

| Hyperparameters | Config |
|---|---|
| optimizer | AdamW |
| learning | rate 0.006 |
| batch | size 384 |
| LR schedule | polynomial |
| warmup epochs | 0 |
| training epochs | 20 |
| weight decay | 0.0001 |
| embedding size | 512 |

### C. Ablation Study

Three main factors affect FaceLiVT-S's performance: knowledge distillation, structural reparameterization, and the MHLA mechanism. This ablation study focuses on the latter two. Table III demonstrates that structural reparameterization significantly reduces latency; removing it from residual or skip connections increases latency from 0.47 ms to 0.50 ms, indicating its role in speeding up computation. Additionally, removing fused Batch Normalization (BN) raises latency to 0.60 ms, highlighting its importance for runtime efficiency. These enhancements do not change the model parameters (Param) or computational costs (FLOP), thereby improving the processing pipeline without adding complexity.

TABLE III
EFFECT OF STRUCTURAL REPARAMETERIZATION

| Ablation | Param (M) | FLOP (M) | Latency (ms) |
|---|---|---|---|
| FaceLiVT-S | 5.05 | 160 | 0.47 |
| w/o Residual Rep | 4.09 | 160 | 0.50 |
| w/o fused BN | 4.10 | 163 | 0.60 |

Table IV presents the effects of modifying $H_e$, the number of heads in MHLA. Decreasing the heads to 8 cuts the parameter count to 4.09 M and slightly reduces FLOPs to 157 M, improving latency to 0.41 ms, but results in a slight drop in accuracy, achieving 99.6% on LFW and 93.9% on CFP-FP. In contrast, increasing $H_e$ to 16 boosts accuracy to 94.6% on CFP-FP and 95.6% on AgeDB-30, with a minor increase in computational demand and latency (now 0.48 ms), suggesting that raising $H_e$ enhances the model's ability to capture complex features, albeit with reduced runtime efficiency.

TABLE IV
ABLATION OF NUMBER HEAD $H_e$

| $H_e$ | Par (M) | FLOP (M) | LFW | CFP-FP | Age DB-30 | Lat (ms) |
|---|---|---|---|---|---|---|
| 8 | 4.09 | 157 | 99.6 | 93.9 | 95.0 | 0.41 |
| 16 | 5.05 | 160 | 99.6 | 94.6 | 95.6 | 0.48 |

The ablation study reveals that structural reparameterization methods and selecting $H_e$ in MHLA are essential for enhancing accuracy and latency in FaceLiVT-S-(LA). Furthermore, knowledge distillation improves accuracy without impacting latency. The ideal configuration, which includes fused BN, residual reparameterization, and $H_e = 16$, successfully balances high accuracy and low latency, making it suitable for real-time applications.

## REFERENCES


[1] W. Yu, M. Luo, P. Zhou, C. Si, Y. Zhou, X. Wang, J. Feng, and S. Yan, "Metaformer is actually what you need for vision," in *Proc. of the IEEE/CVF Conf. Comp. Vis. Pattern Recog.*, pp. 10819–10829, 2022.

[2] S. Chen, Y. Liu, X. Gao, and Z. Han, "Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices," in *Biometric Recognition: 13th Chinese Conference, CCBR 2018, Urumqi, China, August 11-12, 2018, Proceedings 13*, pp. 428–438, Springer, 2018.

[3] Y. Martindez-Diaz, L. S. Luevano, H. Mendez-Vazquez, M. Nicolas-Diaz, L. Chang, and M. Gonzalez-Mendoza, "Shufflefacenet: A lightweight face architecture for efficient and highly-accurate face recognition," in *Proceedings of the IEEE/CVF international conference on computer vision workshops*, pp. 0–0, 2019.

[4] A. Trockman and J. Z. Kolter, "Patches are all you need?," *arXiv preprint arXiv:2201.09792*, 2022.

[5] P. K. A. Vasu, J. Gabriel, J. Zhu, O. Tuzel, and A. Ranjan, "Fastvit: A fast hybrid vision transformer using structural reparameterization," in *Proc. of the IEEE/CVF Int. Conf. on Comp. Vis.*, pp. 5785–5795, 2023.

[6] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5265–5274, 2018.

[7] A. George, C. Ecabert, H. O. Shahreza, K. Kotwal, and S. Marcel, "Edgeface: Efficient face recognition model for edge devices," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2024.

[8] Y. Martinez-Diaz, M. Nicolas-Diaz, H. Mendez-Vazquez, L. S. Luevano, L. Chang, M. Gonzalez-Mendoza, and L. E. Sucar, "Benchmarking lightweight face architectures on specific face recognition scenarios," *Artificial Intelligence Review*, vol. 54, no. 8, pp. 6201–6244, 2021.

[9] X. An, X. Zhu, Y. Gao, Y. Xiao, Y. Zhao, Z. Feng, L. Wu, B. Qin, M. Zhang, D. Zhang, *et al.*, "Partial fc: Training 10 million identities on a single machine," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1445–1449, 2021.

[10] I. Loshchilov, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.