

A Lightweight Dual-Mode Optimization for Generative Face Video Coding

Zihan Zhang, Shanzhi Yin, Bolin Chen, *Member, IEEE*, Ru-Ling Liao, Shiqi Wang, *Senior Member, IEEE*, Yan Ye, *Senior Member, IEEE*

Abstract—Generative Face Video Coding (GFVC) achieves superior rate-distortion performance by leveraging the strong inference capabilities of deep generative models. However, its practical deployment is hindered by large model parameters and high computational costs. To address this, we propose a lightweight GFVC framework that introduces dual-mode optimization—combining architectural redesign and operational refinement—to reduce complexity whilst preserving reconstruction quality. Architecturally, we replace traditional 3×3 convolutions with slimmer and more efficient layers, reducing complexity without compromising feature expressiveness. Operationally, we develop a two-stage adaptive channel pruning strategy: (1) soft pruning during training identifies redundant channels via learnable thresholds, and (2) hard pruning permanently eliminates these channels post-training using a derived mask. This dual-phase approach ensures both training stability and inference efficiency. Experimental results demonstrate that the proposed lightweight dual-mode optimization for GFVC can achieve 90.4% parameter reduction and 88.9% computation saving compared to the baseline, whilst achieving superior performance compared to state-of-the-art video coding standard Versatile Video Coding (VVC) in terms of perceptual-level quality metrics. As such, the proposed method is expected to enable efficient GFVC deployment in resource-constrained environments such as mobile edge devices.

Index Terms—Video compression, lightweight optimization, channel pruning, deep generative model

I. INTRODUCTION

THE increasing demand for video conferencing has driven the need for efficient transmission and compression of facial videos. In recent years, traditional video compression standards, such as H.265/High Efficiency Video Coding (HEVC) [1] and H.266/Versatile Video Coding (VVC) [2], have been developed to improve compression efficiency while maintaining high visual quality. Simultaneously, the emergence of generative models [3], especially Generative Adversarial Networks (GANs) [4], has significantly advanced facial video compression techniques. Generative Face Video Compression (GFVC) [5], [6], [7], [8] employs compact representations at the encoder to model facial motion and deep generative models to reconstruct high-quality video at the decoder, achieving

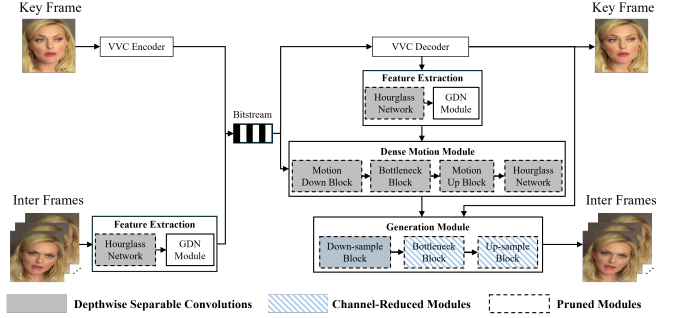


Fig. 1. Overview of the proposed lightweight GFVC framework. The lightweight GFVC system integrates architectural and operational optimization. Architectural optimization includes replacing standard convolutions with depthwise separable convolutions (gray) and reducing channels in the generation module (blue-striped). Operational optimization introduces pruning modules (dashed boxes) that identify and eliminate redundant channels, contributing to model efficiency.

ultra-low bit-rate facial video transmission with vivid reconstructions [9], [10]. Despite its impressive rate-distortion (RD) performance, GFVC suffers from high model complexity, posing challenges for deployment on resource-limited devices.

To address computational challenges of deep learning models, lightweight techniques have been explored to maintain performance with reduced complexity. Notable examples include MobileNet [11], [12], [13], EfficientViT [14] and LDNN-PFD [15], which use efficient convolution operations to reduce parameter amount and complexity. For instance, Light-weighted Temporal Evolution Inference (LW-TEI) [16] employs inverted residual structures to reduce parameters in GFVC models, followed by feature-level knowledge distillation to enhance performance. However, this approach achieves limited parameter reduction and degrades performance in complex GFVC scenarios. Another common approach is model pruning, which eliminates redundant weights, channels, or layers to reduce memory and computation cost. However, most traditional pruning techniques [17], [18], [19], [20] require manually specified configurations, such as fixed pruning ratios and thresholds, which may lack flexibility and generalizability.

Although some pruning methods avoid manual configurations by learning pruning ratios or thresholds automatically [21], [22], [23], their application in the highly complex domain of facial video compression remains limited. This is largely due to the multifaceted nature of GFVC models, which contain various modules and complete diverse tasks in an end-to-end pipeline, such as feature extraction, motion estimation and image generation, which span both low-level vision and high-level semantics. Furthermore, GFVC models

Zihan Zhang, Shanzhi Yin and Shiqi Wang are with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: zhzhzhang38-c@my.cityu.edu.hk, shanzhiyin3-c@my.cityu.edu.hk and shiqiwan@cityu.edu.hk).

Bolin Chen is with DAMO Academy, Alibaba Group, Hangzhou, China; Hupan Lab, Hangzhou, China; and Fudan University, Shanghai, China (e-mail: chenbolin.chenboli@alibaba-inc.com).

Ru-Ling Liao and Yan Ye are with DAMO Academy, Alibaba Group, Sunnyvale, CA 94085 USA (e-mail: ruling.lrl@alibaba-inc.com and yan.ye@alibaba-inc.com).

TABLE I
DISTRIBUTION OF PARAMETERS(M) ACROSS CFTE MODULES

Module	Feature Extraction		Dense Motion Module			Generation Module	
Network Structure	Hourglass Network	GDN Module	Motion Down/Up Block	Bottleneck Block	Hourglass Network	Down/Up-sample Block	Bottleneck Block
Original	13.21M	0.92M	7.82M	18.89M	9.31M	0.36M	6.50M
Proposed w/o pruning	0.70M	0.92M	0.90M	1.32M	1.07M	0.10M	1.63M
Proposed	0.60M	0.92M	0.68M	1.32M	0.61M	0.10M	0.54M

often involve intricate designs with residual connections, skip layers, and cross-modules dependencies, which can cause dimensional mismatches if pruning is applied indiscriminately.

In this paper, we propose a lightweight generative face video compression framework that reduces model complexity while maintaining reconstruction quality. The framework is built on Compact Feature Temporal Evolution (CFTE) [24], one of the representative baselines in GFVC. Specifically, CFTE models the temporal evolution between frames using a compact 4×4 feature matrix for each inter frame, which is extracted and compressed at the encoder side and used to guide frame synthesis via dense motion and occlusion maps at the decoder side. This design enables ultra-low bitrate compression while preserving high perceptual quality. For architectural optimization, we replace standard 3×3 convolutions in CFTE’s parameter-heavy structures with slimmer depthwise separable convolutions, significantly reducing the number of parameters and computational cost. For operational optimization, we introduce a two-stage adaptive channel pruning strategy. In the first stage, soft pruning is applied during training using learnable thresholds to identify redundant channels. In the second stage, hard pruning removes these channels based on the final learned masks. The major contribution of this work can be summarized as follows:

- We propose a lightweight generative face video coding framework that effectively reduces model parameters and computational complexity through dual-mode optimization combining architectural redesign and operational refinement.
- We analyze the parameter and computation distribution across CFTE modules, and introduce targeted architectural optimizations using depthwise separable convolutions and channel slimming to reduce redundancy in modules with high computational cost.
- We develop a structured pruning approach that embeds a differentiable binary mask mechanism into BatchNorm layers. As such, it enables adaptive and automated channel pruning during training, followed by hard pruning and post-training, yielding efficient models without compromising reconstruction quality.

II. PROPOSED METHOD

In this section, we provide an overview of the proposed lightweight optimization strategy for generative face video coding. As illustrated in Fig. 1, our approach consists of two key components: architectural optimization using depthwise separable convolutions and channel reduction, and operational optimization via adaptive channel pruning.

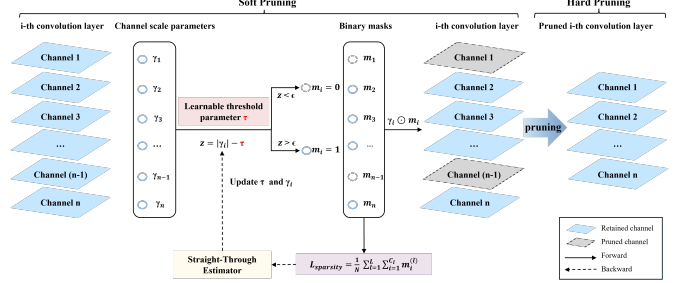


Fig. 2. Adaptive channel pruning for operational optimization. The soft pruning strategy is adopted by learning channel scaling factors and binary masks with a learnable threshold τ . A sparsity loss L_{sparsity} guides the pruning mask learning via a straight-through estimator, followed by hard pruning to remove unimportant channels from the convolutional layers, thereby improving efficiency without significant accuracy loss.

A. Architectural Optimization with Efficient Convolutions and Channel Reduction

To better understand where the computational burden lies, we analyze the parameter and computation distribution across CFTE modules. As shown in Table I, most of the complexity is concentrated in the feature extraction and dense motion modules, particularly in the hourglass networks and bottleneck blocks. Motivated by these observations, we explore lightweight architectural modifications to alleviate redundant computation while preserving performance. Following previous work [25], which introduced lightweight strategies for CFTE, we adopt architectural optimization including depthwise structures and channel reduction. Specifically, as illustrated in Fig. 1, we apply depthwise separable convolutions to selected 3×3 convolutions within the hourglass networks of both the feature extraction and dense motion modules. This yields a computational reduction by a factor of approximately $1/N + 1/D_K^2$, where D_K is the kernel size, and N is the number of output channels. The same replacement is also applied to the motion down/up and bottleneck blocks in the dense motion module. Additionally, we reduce the number of channels in the generation module to further alleviate computational costs.

Compared to the original CFTE model, our architectural optimization achieves substantial parameter reductions in high-complexity components, as summarized in Table I. Specifically, the hourglass networks in the feature extraction and dense motion modules achieve parameter reductions of 94.6% and 88.5%, respectively, through the application of depthwise separable convolutions. Moreover, the number of parameters in the bottleneck blocks is significantly compressed—from 18.89M to 1.32M in the dense motion module via depthwise separable convolutions. And through channel reduction, the number of parameters in the bottleneck block in generation module is compressed from 6.50M to 1.63M. These results demonstrate that most of the complexity lies in the hourglass

structures and bottleneck blocks, and our architectural optimization effectively targets these high-cost components.

B. Operational Optimization with Adaptive Channel Pruning

Traditional network pruning methods typically rely on manually predefined pruning ratios or thresholds, which require extensive empirical tuning and may result in suboptimal pruning configurations. To address this limitation, we propose an adaptive channel pruning framework based on learnable thresholds, as illustrated in Fig. 2. This approach introduces a differentiable binary mask in BatchNorm layers to achieve structured pruning optimization.

Specifically, we design a modified BatchNorm layer, termed MaskedBatchNorm2d, which integrates a learnable threshold parameter to adaptively modulate the BatchNorm scale parameter, thereby controlling the sparsity of each channel. Inspired by Network Slimming [26]—which treats the BatchNorm scale parameter as an indicator of channel importance—our method improves upon it by eliminating the need for manually setting global pruning ratios or thresholds. The threshold is jointly optimized with model parameters, allowing channel selection to be learned in a data-driven manner. Therefore, we construct a channel-wise binary mask $m_i \in \{0, 1\}$ based on the scale parameters γ_i and the learned threshold τ :

$$z_i = |\gamma_i| - \tau, \quad m_i = \mathbb{I}(z_i \geq 0) \quad (1)$$

where $\mathbb{I}(\cdot)$ denotes the indicator function, and τ is initialized to 0.1 in our experiments. The mask effectively zeros out channels where the scale parameter magnitude is below the threshold, enabling channel-wise pruning.

Since the binary mask m_i involves a non-differentiable indicator function, direct gradient computation is not feasible. To address this, we utilize the Straight-Through Estimator (STE) [27] to approximate gradients during back-propagation. Inspired by the theoretical analysis in [28], which shows that coarse gradients in STE correlate more positively with the true gradient near decision boundaries, we adopt a clipped variant of STE. Specifically, we allow gradients to propagate only within a small ϵ -neighborhood of the threshold boundary to improve convergence stability and reduce estimation bias. Within this band, we set $\partial m_i / \partial z_i \approx 1$ and compute the following gradients, the gradient approximations for the mask with respect to γ_i and τ are given by:

$$\frac{\partial m_i}{\partial \gamma_i} \approx \begin{cases} \text{sign}(\gamma_i), & \text{if } |z_i| \leq \epsilon \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

$$\frac{\partial m_i}{\partial \tau} \approx \begin{cases} -1, & \text{if } |z_i| \leq \epsilon \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where ϵ is a small constant that controls the width of the gradient propagation window. In our experiments, we set $\epsilon = 1.0$ to balance gradient stability and learning efficiency. The forward operation for the masked BatchNorm scale parameter $\hat{\gamma}_i$ is defined as:

$$\hat{\gamma}_i = \gamma_i \odot m_i \quad (4)$$

where $\odot$ denotes element-wise multiplication. This formulation ensures that channels with $m_i = 0$ are effectively

TABLE II
COMPARISON OF MODEL COMPLEXITY AND COMPRESSION EFFICIENCY
(VS. VVC ANCHOR) FOR CFTE VARIANTS USING DIFFERENT
LIGHTWEIGHT STRATEGIES

Method	Params (M)	KMACs/Pixel	Rate-DISTS	Rate-LPIPS
Original	58.03	852.00	-56.68%	-52.41%
LW-TEI	36.85	381.93	-57.19%	-50.08%
JVET-AH0109	28.33	209.31	-55.54%	-47.89%
Proposed (CFTE-Lite) w/o pruning	7.65	174.24	-56.76%	-48.45%
Proposed (CFTE-Lite)	5.58	94.57	-58.04%	-51.87%

deactivated by setting their corresponding scale to zero during forward propagation.

To encourage the model to learn a sparse channel configuration, we define the sparsity loss as:

$$\mathcal{L}_{\text{sparsity}} = \frac{1}{N} \sum_{l=1}^L \sum_{i=1}^{C_l} m_i^{(l)} \quad (5)$$

where N denotes the total number of channels across all layers, L is the number of layers subject to pruning, C_l is the number of channels in the l -th layer. The overall training objective is defined as :

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda_{\text{sparse}} \mathcal{L}_{\text{sparsity}} \quad (6)$$

where $\mathcal{L}_{\text{task}}$ denotes the primary loss function used for training the CFTE model, and λ_{sparse} is a regularization coefficient that balances the sparsity constraint. To prevent premature pruning and ensure stable convergence, we adopt a progressive training strategy, where $\lambda_{\text{sparse}} = 0$ in the initial phase, with the sparsity constraint gradually introduced in subsequent stages.

Moreover, our pruning process is designed as two stages. In the first stage, *soft pruning* is performed during training: the network architecture remains unchanged, but redundant channels are dynamically suppressed through binary masks with STE-based gradient approximation in the proposed MaskedBatchNorm2d. During this phase, both the scale parameter γ and threshold τ are jointly optimized— γ is influenced by task-specific and sparsity losses, while τ is updated via the STE to gradient flow through the non-differentiable masking operation. In the second stage, *hard pruning* is performed after training by removing channels whose final mask value is zero. However, to preserve architectural integrity, channels that are structurally required by subsequent layers are retained, even if their mask values are zero. This results in a compact and efficient network architecture, followed by a fine-tuning step to refine the remaining channels and recover potential performance degradation introduced by pruning. This approach removes the need for manual pruning ratio specification and allows the network to autonomously learn an optimal sparse structure based on data-driven criteria, while maintaining training stability through differentiability.

III. EXPERIMENTS

A. Experimental Settings

1) *Implementation Details*: The proposed lightweight CFTE model (CFTE-Lite) is trained on the VoxCeleb2 [29] and CelebVHQ [30] datasets, using images with a resolution of 256×256 . We implement the proposed model with the PyTorch framework and use NVIDIA TESLA A100 GPUs for

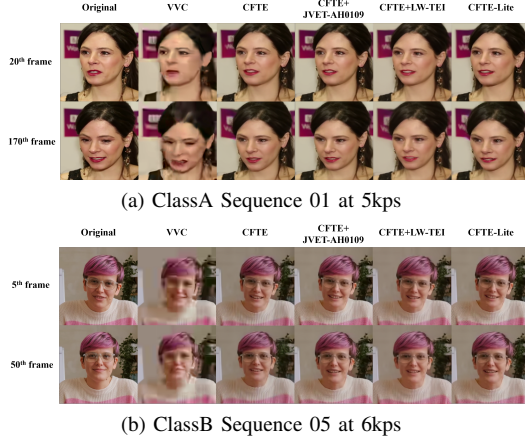


Fig. 3. Visual quality comparisons among VVC, CFTE and CFTE variants at similar bit rates for each sequence.

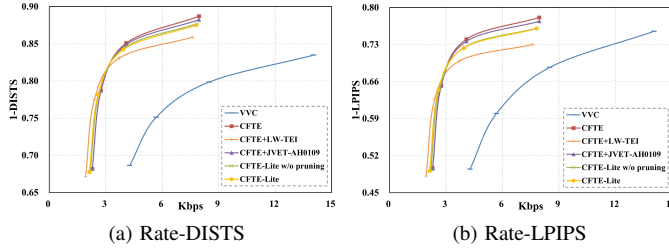


Fig. 4. Rate-Distortion performance comparisons of VVC, CFTE and CFTE variants in terms of DISTS and LPIPS.

training and fine-tuning. During training, the Adam optimizer is employed with parameters $\beta_1 = 0.5$ and $\beta_2 = 0.999$, and a learning rate of 0.0002 to ensure convergence. The sparsity coefficient λ_{sparse} is initially set to 0 for the first 40 epochs and then increased to 0.1. The model is trained for 100 epochs in total, followed by an additional 50 epochs of fine-tuning after hard pruning. For performance evaluation, we follow the GFVC standardization activities in Joint Video Experts Team (JVET) of ISO/IEC JTC 1/SC 29 and ITU-T SG21 and adopt the test sequences specified in JVET-AG2035 [31]. Specifically, the test set consists of 33 sequences at 256×256 resolution: Class A contains 15 face-centric sequences, each with 250 frames, and Class B includes 18 head-and-shoulder sequences, each with 125 frames.

2) *Compared Methods*: To demonstrate the effectiveness of the proposed CFTE-Lite, we compare its performance with the VVC, the original CFTE model, the CFTE model optimized with the lightweight approach proposed in LW-TEI [16] and JVET-AH0109 [25]. For VVC, we employ the VTM 22.2 reference software in Low-Delay Bidirectional mode with Quantization Parameters (QPs) set to 37, 42, 47, and 52. For CFTE and its variants, key-reference frames are encoded using VTM 22.2 with QPs of 22, 32, 42, and 52, and the subsequent inter frames are compressed using the GFVC solution.

3) *Evaluation Measures*: We employ two perceptual-level metrics to assess the quality of the generated videos: Deep Image Structure and Texture Similarity (DISTS) [32] and Learned Perceptual Image Patch Similarity (LPIPS) [33]. Bjøntegaard Delta Rate (BD-rate) [34] and RD curve are adopted to quantify the overall compression performances of different models. To present the RD curves in increasing order and calculate the savings of the BD-rate, we use “1-DISTS”, “1-LPIPS” as the y-axis of the graphs.

B. Comparisons

1) *Model Complexity*: Table II compares the parameters and KMACs/pixel between the original CFTE, the LW-TEI CFTE variant, the JVET-AH0109 CFTE variant, and CFTE-Lite. The proposed CFTE-Lite requires only 5.58M parameters and 94.57 KMACs/pixel. This corresponds to 90.4% parameters reduction and 88.9% computation reduction compared to the original CFTE model, demonstrating the effectiveness of our optimization.

2) *Rate-Distortion Performance*: Table II shows the BD-rate savings of the original CFTE, the LW-TEI CFTE variant, the JVET-AH0109 CFTE variant, and the proposed CFTE-Lite compared to the VVC anchor. CFTE-Lite achieves BD-rate savings of 58.04% and 51.87% for Rate-DISTS and Rate-LPIPS, respectively, outperforming both the LW-TEI CFTE variant and the JVET-AH0109 variant. Figure 4 illustrates the RD performance in terms of DISTS and LPIPS for VVC, CFTE, the LW-TEI CFTE variant, the JVET-AH0109 CFTE variant, and the proposed CFTE-Lite. It can be observed that CFTE-Lite significantly outperforms VVC and the LW-TEI CFTE variant while exhibiting only slight performance degradation compared to CFTE, within an acceptable range. This demonstrates that CFTE-Lite effectively balances RD performance and model complexity.

3) *Subjective Quality*: As shown in Figure 3, we visualize two sequences under ultra-low bitrates. VVC results suffer from strong blocking artifacts, while the proposed CFTE-Lite still reconstructs coherent and realistic facial structures with a significantly reduced model size. In particular, the proposed CFTE-Lite reconstructions show negligible additional distortion compared to the original CFTE reconstructions.

C. Ablation Studies

To further validate the effectiveness of the proposed adaptive channel pruning method, we conduct an ablation study comparing the proposed CFTE-Lite with and without the pruning operation. It can be seen from Table II that the pruning process reduces the parameter amount from 7.65M to 5.58M and the computational cost from 174.24 to 94.57 KMACs/pixel. Fig. 4 shows the RD performance of CFTE-Lite without pruning and CFTE-Lite in terms of DISTS and LPIPS. It can be observed that the pruned model achieves nearly identical RD performance compared to the unpruned variant. This result demonstrates that the proposed adaptive channel pruning effectively removes redundant channels while preserving critical features for high-quality facial video reconstruction.

IV. CONCLUSION

This paper proposes a lightweight generative face video compression framework that reduces model complexity through architectural and operational optimization. By integrating efficient depthwise convolutional layers and a two-stage adaptive pruning method, our approach achieves a 90.4% reduction in parameters and 88.9% savings in KMACs/pixel compared to the baseline CFTE model, while maintaining promising RD performance. These results demonstrate the potential of lightweight GFVC for future deployment on resource-constrained devices.

REFERENCES

- [1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.
- [2] B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohm, "Overview of the versatile video coding (VVC) standard and its applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3736–3764, 2021.
- [3] B. Chen, S. Yin, P. Chen, S. Wang, and Y. Ye, "Generative visual compression: A review," in *2024 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2024, pp. 3709–3715.
- [4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.
- [5] B. Chen, J. Chen, S. Wang, and Y. Ye, "Generative face video coding techniques and standardization efforts: A review," in *Data Compression Conference*, 2024, pp. 103–112.
- [6] B. Chen, Z. Wang, B. Li, S. Wang, S. Wang, and Y. Ye, "Interactive face video coding: A generative compression framework," *IEEE Transactions on Image Processing*, vol. 34, pp. 2910–2925, 2025.
- [7] B. Chen, Z. Wang, B. Li, S. Wang, and Y. Ye, "Compact temporal trajectory representation for talking face video compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 11, pp. 7009–7023, 2023.
- [8] S. Yin, B. Chen, S. Wang, and Y. Ye, "Enabling translatability of generative face video coding: A unified face feature transcoding framework," in *Data Compression Conference*, 2024, pp. 113–122.
- [9] B. Chen, Y. Ye, J. Chen, R.-L. Liao, S. Yin, S. Wang, K. Yang, Y. Li, Y. Xu, Y.-K. Wang *et al.*, "Standardizing generative face video compression using supplemental enhancement information," *arXiv preprint arXiv:2410.15105*, 2024.
- [10] B. Chen, S. Yin, G. Konuko, G. Valenzise, Z. Zhang, S. Wang, and Y. Ye, "Generative models at the frontier of compression: A survey on generative face video coding," *arXiv preprint arXiv:2506.07369*, 2025.
- [11] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.
- [12] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.
- [13] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1314–1324.
- [14] H. Cai, J. Li, M. Hu, C. Gan, and S. Han, "Efficientvit: Lightweight multi-scale attention for high-resolution dense prediction," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 17302–17313.
- [15] D. Liu, X. Guo, N. Wang, and Q. Wu, "Lightweight deep neural network model with padding-free downsampling," *IEEE Signal Processing Letters*, vol. 31, pp. 865–869, 2024.
- [16] Z. Zhang, B. Chen, S. Yin, S. Wang, and Y. Ye, "Light-weighted temporal evolution inference for generative face video compression," in *2024 IEEE 26th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2024, pp. 1–6.
- [17] C. Liu and H. Wu, "Channel pruning based on mean gradient for accelerating convolutional neural networks," *Signal Processing*, vol. 156, pp. 84–91, 2019.
- [18] H. Li, C. Li, M. Xue, G. Fang, S. Zhou, Z. Feng, H. Wang, Y. Wang, L. Cheng, M. Song *et al.*, "Pruningbench: A comprehensive benchmark of structural pruning," *arXiv preprint arXiv:2406.12315*, 2024.
- [19] J.-Y. Wu, K. Gamlath, and A. Misra, "Pr-ge-ne: Efficient encoding of pervasive video sensing streams by pruned generative networks," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 21, no. 2, pp. 1–22, 2025.
- [20] W. Lu, X. Ding, Y. Jiang, and J. Chu, "A novel pruning algorithm based on data-driven and knowledge-driven global channel pruning," *IEEE Signal Processing Letters*, 2025.
- [21] D. Saxena, J. Cao, J. Xu, and T. Kulshrestha, "Rg-gan: Dynamic regenerative pruning for data-efficient generative adversarial networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 38, no. 5, 2024, pp. 4704–4712.
- [22] M. Ma, T. Zhang, Z. Wang, Y. Wang, T. Du, and W. Li, "Global channel pruning with self-supervised mask learning," *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.
- [23] J. Wu, Z. Li, Z. Hui, Y. Zhang, L. Kong, and X. Yang, "Quanticache: Adaptive importance-guided quantization with hierarchical latent and layer caching for video generation," *arXiv preprint arXiv:2503.06545*, 2025.
- [24] B. Chen, Z. Wang, B. Li, R. Lin, S. Wang, and Y. Ye, "Beyond keypoint coding: Temporal evolution inference with compact feature representation for talking face video compression," in *Data Compression Conference*, 2022, pp. 13–22.
- [25] R. Zou, R.-L. Liao, B. Chen, Y. Ye, and J. Chen, "AHG16: Removal of the analysis model on the decoder side and lightweight design," *The Joint Video Experts Team of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29, doc. no. JVET-AH0109*, April 2024.
- [26] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Learning efficient convolutional networks through network slimming," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2736–2744.
- [27] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv preprint arXiv:1308.3432*, 2013.
- [28] P. Yin, J. Lyu, S. Zhang, S. Osher, Y. Qi, and J. Xin, "Understanding straight-through estimator in training activation quantized neural nets," *arXiv preprint arXiv:1903.05662*, 2019.
- [29] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," *arXiv preprint arXiv:1806.05622*, 2018.
- [30] H. Zhu, W. Wu, W. Zhu, L. Jiang, S. Tang, L. Zhang, Z. Liu, and C. C. Loy, "CelebV-HQ: A large-scale video facial attributes dataset," in *ECCV*, 2022.
- [31] S. McCarthy and B. Chen, "Test conditions and evaluation procedures for generative face video coding," *The Joint Video Experts Team of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29, doc. no. JVET-AG2035*, January 2024.
- [32] K. Ding, K. Ma, S. Wang, and E. P. Simoncelli, "Image quality assessment: Unifying structure and texture similarity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 5, pp. 2567–2581, 2020.
- [33] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 586–595.
- [34] G. Bjontegaard, "Calculation of average psnr differences between rd-curves," *ITU SG16 Doc. VCEG-M33*, 2001.