International Journal of Neural Systems, (2025) 2550070 (13 pages)
© World Scientific Publishing Company
DOI: 10.1142/S0129065725500704

World Scientific
www.worldscientific.com

Evolutionary Channel Pruning for Style-Based Generative Adversarial Networks

Yixia Zhang
School of Computer Science and Electronic Engineering, University of Surrey, Guildford, United Kingdom

Ferrante Neri*
School of Computer Science and Electronic Engineering, University of Surrey, Guildford, United Kingdom
School of Software, Nanjing University of Information Science and Technology, Nanjing, P. R. China
f.neri@surrey.ac.uk

Xilu Wang
School of Computer Science and Electronic Engineering, University of Surrey, Guildford, United Kingdom

Pengcheng Jiang and Yu Xue
School of Software, Nanjing University of Information Science and Technology, Nanjing, P. R. China

Received 18 July 2025
Accepted 2 September 2025
Published Online 27 September 2025

Generative Adversarial Networks (GANs) have demonstrated remarkable success in high-quality image synthesis, with StyleGAN and its successor, StyleGAN2, achieving state-of-the-art performance in terms of realism and control over generated features. However, the large number of parameters and high floating-point operations per second (FLOPs) hinder real-time applications and scalability, posing challenges for deploying these models in resource-constrained environments such as edge devices and mobile platforms. To address this issue, we propose Evolutionary Channel Pruning for StyleGANs (ECP-StyleGANs), a novel algorithm that leverages evolutionary algorithms to compress StyleGAN and StyleGAN2 while maintaining competitive image quality. Our approach encodes pruning configurations as binary masks on the model's convolutional channels and iteratively refines them through selection, crossover, and mutation. By integrating carefully designed fitness functions that balance model complexity and generation quality, ECP-StyleGANs identifies optimally pruned architectures that reduce computational demands without compromising visual fidelity, achieving approximately a 4 × reduction in FLOPs and parameters, while maintaining visual fidelity with only a slight increase in FID (Fréchet Inception Distance) compared to the original un-pruned model. This study should be interpreted as a preliminary step towards the formulation and management of the generative AI pruning problem as a multi-objective optimisation task, aimed at enhancing the trade-off between model efficiency and image quality, thereby making large deep models more accessible for real-world applications such as edge devices and resource-constrained environments. **Source codes will be available.**

Keywords: StyleGANs; channel pruning; evolutionary algorithm; generative AI.

*Corresponding author.

1. Introduction

Generative Adversarial Networks (GANs)[1] have significantly advanced the broader field of generative modeling, making it possible to produce realistic data across diverse domains such as the medical field,[2] industrial applications,[3–5] and beyond. Among these, GANs have been particularly impactful in image synthesis, where they generate high-fidelity images through an adversarial training process.[6,7] However, traditional GANs often face challenges such as mode collapse, limited control over generated features, and unstable training dynamics.[8] To address these limitations, StyleGAN[9,10] introduced a novel architecture that separates the latent space from the generated image space, allowing for more precise control over image attributes through style modulation at multiple layers. This innovation not only improves image quality and diversity but also facilitates fine-grained manipulation of visual features, making StyleGAN particularly well-suited for applications requiring high-resolution, photorealistic image synthesis.[11,12] Despite these advantages, StyleGANs ability to produce highly realistic images comes with a high computational cost, which restricts its deployment in resource-constrained environments such as edge devices and mobile platforms, where both computational power and memory are limited.

Various network compression techniques have been explored to reduce the computational demands of StyleGAN architectures. Among these, pruning has proven to be an effective approach for eliminating redundant parameters.[13,14] Two common pruning strategies, weight pruning and channel pruning, are often applied to the convolutional layers of StyleGANs, which are the most computationally intensive components.[15] Weight pruning removes individual weights (connections) within a layer based on some importance criterion, resulting in a sparse weight matrix. Channel pruning, in contrast, removes entire feature channels (and their associated kernels) from a layer, yielding a smaller dense network. While StyleGAN[9] introduced a groundbreaking architecture that enabled high-quality image synthesis with fine control over generated features, StyleGAN2[10] refined this approach by addressing some of the limitations of its predecessor. StyleGAN2 improved perceptual quality by removing artefacts, enhancing the generators normalisation, and introducing path length regularisation, which stabilised training and increased image fidelity. Although early pruning methods primarily focused on conditional GANs (e.g. pix2pix[16] and CycleGAN[17]), recent research has extended these techniques to unconditional GANs, such as StyleGAN and StyleGAN2.

Jiwoo Chung *et al.*[18] introduced a channel pruning method that leverages the varying sensitivities of channels to latent vectors, enhancing sample diversity in the compressed model. Yuchen Liu *et al.*[19] proposed a specialized combination of channel pruning and knowledge distillation for unconditional GANs, focusing on preserving critical content areas such as human faces. These methods target specific objectives, such as maintaining diversity or retaining quality in specific regions, but may fail to capture globally significant yet less salient features. This limitation can lead to suboptimal performance in tasks that require high overall image quality and balanced feature representation.

Evolutionary algorithms (EAs) provide an effective solution to these challenges by simulating pruning processes to find optimal pruning configurations.[20,21] Unlike traditional approaches that rely on fixed heuristics, evolutionary methods can dynamically explore the solution space through iterative refinement, leveraging evolutionary operators, i.e. selection, crossover and mutation operators, to evaluate and evolve multiple pruning strategies simultaneously.[20] This population-based mechanism in EAs has demonstrated effectiveness in various optimisation tasks, ranging from image classification[22,23] to engineering design,[24–26] especially for automatically discovering good neural architectures.[27,28] However, leveraging EAs for pruning StyleGANs remains an underexplored domain.

To bridge this gap, we propose **E**volutionary **C**hannel **P**runing for **StyleGANs** (ECP-StyleGANs), an EA framework that achieves channel-wise pruning for StyleGANs by trade-offing between model efficiency and image fidelity. The proposed method iteratively prunes model channels through evolutionary operators — selection, crossover, and mutation, guided by fitness criteria that jointly evaluate parameter count reduction, image diversity, and perceptual quality. Specifically, through crossover and mutation operators,[29] ECP-StyleGANs

explores a diverse set of pruning configurations, enabling it to avoid suboptimal solutions. The selection criterion strategically guides pruning decisions by balancing different critical aspects of model performance. By continuously refining the model through an evolutionary process based on performance metrics, our method ensures a balance between computational cost and high-quality image generation. Extensive experiments demonstrate that pruned models achieved by ECP-StyleGANs effectively reduce the number of parameters and FLOPs (Floating Point Operations), achieving a $3 \times 4 \times$ reduction on StyleGAN and StyleGAN2, respectively, while maintaining visually negligible image quality loss compared to the full-size model.

The proposed evolutionary channel pruning StyleGANs (ECP-StyleGANs) offers the following contributions:

- We apply evolutionary algorithms to compress StyleGAN and StyleGAN2. By encoding binary masks on model channels and iteratively refining them through selection, crossover, and mutation, our method effectively reduces the number of parameters and FLOPs while maintaining high image quality.
- We design and implement selection criteria within the evolutionary process to guide pruning decisions. These criteria enable a thorough exploration of trade-offs between model complexity, generation quality, and diversity, ultimately leading to an optimally pruned model.
- We provide a comprehensive evaluation framework that uses Pareto front graphs[30] to systematically analyze the trade-offs between model efficiency and generated image quality. This framework facilitates a nuanced comparison between ECP-StyleGANs and state-of-the-art (SOTA) methods.

The rest of the paper is organized as follows. Section 2 details the proposed method, while Sec. 3 presents the experimental results. Finally, Sec. 4 concludes the paper with key findings and remarks.

2. Proposed Method

As illustrated in Fig. 1, the framework of ECP-StyleGANs consists of two main phases: the evolutionary search phase and the training phase. The process begins with a full-size StyleGAN generator serving as the base model for pruning. During the evolutionary search phase, an evolutionary algorithm is adopted to iteratively explore different candidate pruning solutions, generating K distinct pruned models, each representing a unique channel reduction strategy. Given the proposed selection criterion that considers different objectives, all candidate pruned models are evaluated and the best-performing pruned generator is identified. Following this, the training phase allows the optimal pruned generator to be trained with noise inputs to generate images, which are subsequently evaluated by a discriminator that differentiates between real and generated images.

The evolutionary search phase is further detailed in Fig. 2. It begins with a randomly initialized population, where each individual represents a candidate pruned generator. Note that each individual is encoded as a binary mask vector, where bits represent the activation state (retained/pruned) of corresponding channels in the StyleGAN generator. This binary encoding not only simplifies the pruning

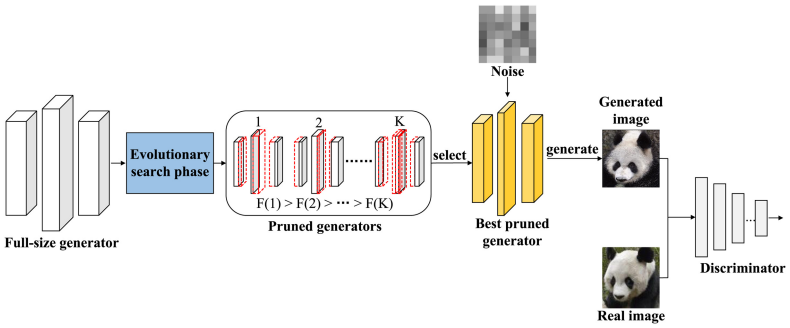

Fig. 1. The framework of ECP-StyleGANs. During the evolutionary search phase, K pruned generators are obtained from the full-size generator, and the best-performing generator is selected for StyleGAN training according to their fitness values F(K).

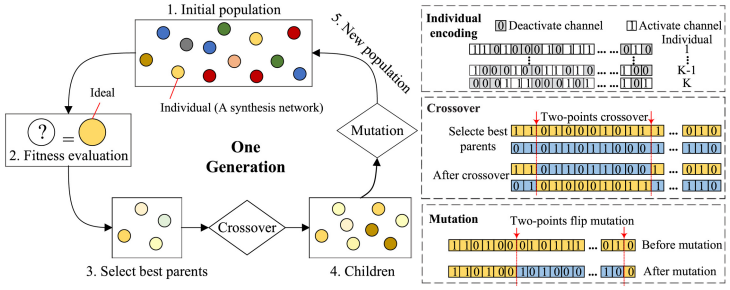

Fig. 2. Evolutionary search phase for channel pruning for StyeleGANs. It begins with the random initialization of a population. The population is then evaluated using fitness functions, among which elit individuals are selected. Following this, crossover and mutation operators are used to generate new individuals for the next generation of population. This process repeats until the maximum number of generations is reached.

process but also facilitates efficient manipulation of model structures during evolutionary search.

Each individual then undergoes fitness evaluation based on predefined criteria considering different objectives. Elite individuals are then selected as parents for the next generation. A two-point crossover operator is applied to these selected parents, where segments of their binary encoding are exchanged to generate offspring with mixed traits. Additionally, a mutation operator is used to stochastically flip bits in the offsprings binary encoding. This controlled randomness prevents premature convergence to local optima and enhances population diversity. This cycle of selection, crossover, mutation, and evaluation repeats until the maximum number of generations is reached, gradually refining the population towards an optimal pruned generator.

2.1. *Encoding and initialisation*

Effective evolutionary search for StyleGAN pruning requires a compact and interpretable encoding of potential solutions. Given a StyleGAN network, we encode the convolution layers in the generator using binary masks, where each bit corresponds to a channel within a layer. A value of 1 in the mask retains the channel, while 0 deactivates it by setting its weights to zero. This encoding method is illustrated in the right panel of Fig. 2, where the grey blocks with 0 represent deactivated channels, and the white blocks with 1 indicate active ones. We randomly initialize a population of K candidate individuals, each representing a pruning solution with a unique configuration of binary masks.

2.2. *Fitness evaluation*

The design of the fitness function plays a critical role in evaluating the quality of each individual in the population, as it directly influences the selection process and guides evolutionary optimisation.[31] The proposed fitness function is designed to balance two competing objectives: preserving visual quality and reducing computational complexity. The fitness function F, defined in Eq. (1), consists of two key factors: the Fréchet Inception Distance (FID) F_{fid},[32] which measures the quality and diversity of the generated images, and the number of activated channels F_{ch}, which measures the compression ratio. These metrics are combined to assess the trade-off between image generation quality and model efficiency, ensuring that the evolutionary process converges to pruning solutions that achieve an optimal balance. Notably, unlike multi-objective optimization,[33,34] the fitness function is a single composite metric guiding the evolutionary pruning process. Hence, we define the fitness $F(k)$ for the kth individual as

$$F(k) = F_{\text{fid}}(k) + F_{\text{ch}}(k), \tag{1}$$

where $F_{\text{fid}}(k)$ is used to measure the distance between the images generated by the kth compressed generator and real images, the lower the better.

Specifically, the FID is calculated by

$$\text{FID} = \|\mu_r - \mu_g\|_2^2 + \text{Tr}\left(\sum_r + \sum_g - 2\left(\sum_r \sum_g\right)^{\frac{1}{2}}\right), \tag{2}$$

where μ_r, μ_g, $\sum_r$ and $\sum_g$ are the means and the covariance matrices of the real (subscripted as r) and

generated data (subscripted as g) distributions, respectively, and $Tr()$ represents the trace operation. The definition of F_{fid} is

$$F_{\text{fid}}(k) = \frac{1}{1 + e^{-\text{FID}(k)}}, \tag{3}$$

where the value range of F_{fid} is [0,1].

To measure the compression ratio of the generator, F_{ch} is defined as

$$F_{\text{ch}}(k) = \frac{\text{sum}(\text{mask}(k))}{\text{len}(\text{mask}(k))}, \tag{4}$$

where $\text{sum}(\text{mask}(k))$ cumulates the number of 1 in the k-th binary mask, and $\text{len}(\text{mask}(k))$ is the length of the k-th mask. The value range of F_{ch} is also [0,1]. The lower the F_{ch}, the smaller the number of parameters in the compressed generator.

2.3. *Selection, crossover and mutation*

After evaluating the initial population, the evolutionary algorithm proceeds with standard genetic operations. Roulette selection[35] is employed to choose individuals for reproduction via crossover and mutation operators, allowing those with better fitness values to be selected with higher probabilities. This approach ensures that better-performing solutions are favored while preserving genetic diversity. Since our optimisation objectives are to minimize F_{fid} and F_{ch}, the selection probability of each individual is defined as

$$P^k = \frac{(F(k))^{-1}}{\sum_{k=1}^{K} F(k)}, \tag{5}$$

where P^k is the k-th individual in the population and K is the population size.

The selected individuals then undergo a two-point crossover, as shown on the right side of Fig. 2, in which two parent individuals exchange segments of their binary masks at two randomly chosen points. This operation creates offspring that inherit characteristics from both parents, promoting genetic diversity while preserving beneficial traits. To further enhance the exploration ability of the offspring population, we apply flip mutation, as illustrated on the right side of Fig. 2. Flip mutation randomly selects bits within an individual's binary mask, flipping their values (i.e. 0s become 1s and *vice versa*) to explore the search space further and prevent premature convergence.

Algorithm 1. Evolutionary Channel Pruning for StyleGANs (ECP-StyleGANs)

Input: Maximum number of iterations T, population size K;
Initialise a population with K individuals;
Select the local-best individual $\hat{P}^{(0)}$;
for $t = 1$ **to** T **do**
  Calculate the fitness of each individual based on Eq. (1);
  Obtain selection probabilities based on Eq. (5);
  Preserve the local-best individual $\hat{P}^{(t-1)}$;
  for $k = 2$ **to** K **do**
    Generate a random value $r \sim [0, 1]$;
    Perform selection, crossover, and mutation to generate offspring based on probability r;
  end for
  Replace the population with the offspring population for the subsequent generation;
end for
Output: The optimal pruned generator $\hat{P}^{(T)}$.

Additionally, we incorporate elitism[36] to ensure that the best individuals are preserved across generations. The top-performing individual from each generation is carried over to the next without modification, preventing the loss of high-quality solutions due to stochastic genetic operations.

This evolutionary process is repeated for T generations, ultimately yielding the best-performing individual, which is selected for further training. Algorithm 1 outlines the detailed steps for searching for the optimal pruned generator.

3. Experimental Results

The experiments on both StyleGAN and StyleGAN2 are conducted in two stages: the evolutionary search stage and the training stage. In the evolutionary search stage, we do not train the models from scratch. Instead, we use pre-trained StyleGAN and StyleGAN2 architectures, initially trained on the FFHQ[9] dataset. The evolutionary search process is then applied to obtain various pruned generator configurations based on these pre-trained models. Each pruned individual in the population is trained for 500 iterations, and the optimal pruned generator is selected based on its balance between model compression and image generation quality. In the training stage, the selected model is fine-tuned according to the original training scheme for additional iterations. All experiments are implemented

using the PyTorch framework and executed on an NVIDIA RTX 3090 GPU.

3.1. *Experiments on StyleGAN*

We employ two animal face datasets for StyleGAN experiments: the Animal Face dataset,[37] comprising 21 classes and 2,420 images with a resolution of 256×256 pixels, and the Animal FacesHQ (AFHQ) dataset,[38] consisting of 3 classes and 15,000 images with a resolution of 512×512 pixels.

During the evolutionary search phase, we configure the following parameters: a maximum of $T = 10$ iterations and a population size of $K = 15$. It should be noted that T and K determine the total number of candidate evaluations and can be adjusted according to the computational budget and task complexity. We adopt widely used settings from prior evolutionary computation literature, setting the crossover probability to 0.7 and the mutation probability to 0.1. After completing the evolutionary search phase and obtaining the optimal pruned generator, we proceed to fine-tune the StyleGAN model by training it for an additional 50,000 iterations. Notably, training was successfully conducted on both datasets using the source models. Given the scarcity of research on StyleGAN compression, we compare our approach with two established pruning techniques: $L1$ pruning ($L1$) and random pruning (Random). $L1$ pruning retains channels with higher cumulative weight values, deactivating those with smaller weights, while random pruning selects the pruning ratio for each layer randomly. The StyleGAN implementation is based on the repository at https://github.com/sangwoomo/FreezeD.

3.1.1. *Results on animal face dataset*

To validate the effectiveness of the evolutionary process, we test a randomly selected pruned generator, with a comparison to the optimal pruned generator presented in Fig. 3. Compared to a random individual from the population, the optimal pruned model can generate images with more details. For example, there are no clear eyes on the eagle images using the randomly selected individual, the faces of panda and tiger are not completed, and the fur texture on the dog, deer, and wolf is less clear. In addition, cat and elephant images are entangled with the backgrounds.

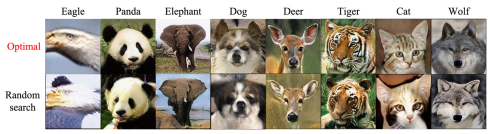

Fig. 3. Comparison of generated images between the optimal pruned generator (top) and a randomly selected individual (bottom) on the Animal Face dataset.

To further demonstrate the effectiveness of our proposed method, we compare our method with the full-size model, $L1$ pruning, and random pruning. The comparison of their generated images is shown in Fig. 4. From the results, we can see the images generated by our optimal pruned model maintain visually negligible image quality loss compared to the full-size model, with some slight blurring in the texture of the animal fur on bird and panda heads. Images generated by $L1$ pruning and random pruning are of lower quality, showing lower clarity and head incompleteness, see the last two rows in Fig. 4.

To assess the quality and fidelity of the generated images, we use the FID to measure the difference between real and generated images, where lower values indicate better quality. Additionally, we use FLOPs to evaluate model efficiency, lower FLOPs indicate more deactivated channels and a lighter model. The comparative results with the full-size model, $L1$ pruning and random pruning are presented in Table 1. From the table, it is clear that our method reduces FLOPs by $3 \times$ compared to the full-size model, nearly half of the channels are deactivated. For a fair comparison, we also compress the $L1$ pruning and random pruning models to the same FLOPs value of 13.2 M, their FID scores are higher than our method. These results prove the effectiveness of the proposed method on the Animal Face

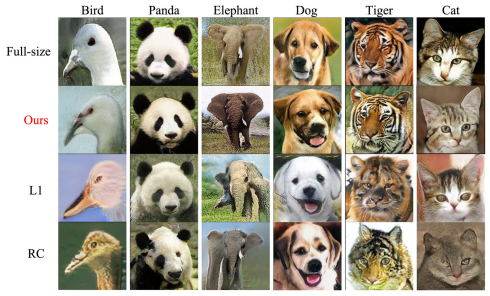

Fig. 4. Comparison of generated images with the full-size model, $L1$ pruning, and random pruning on the Animal Face dataset ($L1$: $L1$ pruning, RC: random pruning).

Table 1. Comparisons of FID and FLOPs on Animal Face dataset.

Method	FLOPs	FID	CH_num
*	40.9 M	58.84	2544
Ours	13.2 M	**59.45**	1231
L1[39]	13.2 M	62.23	1231
Random[40]	13.2 M	65.62	1231

Note: * the original full-size model. **CH_num**: Number of active channels.

dataset, maintaining the quality of generated samples while significantly improving the model efficiency.

3.1.2. *Results on animal FaceHQ (AFHQ) dataset*

We further validate the effectiveness of our method on the Animal FaceHQ dataset. The generated images by using the optimal pruned generator and a randomly selected individual are presented in Fig. 5. From the bottom row, we can see that there are obvious flaws on the heads of lions, tigers, and dogs. The texture of cats is a bit blurred compared to that of the optimal individual. It demonstrates the evolutionary process can avoid sub-optimal individuals.

Similarly, we compare the optimal pruned model with the full-size model, $L1$ pruning, and random pruning techniques. The comparison of generated images is illustrated in Fig. 6, and the comparison of FLOPs, FID, and number of activated channels is presented in Table 2. From the results, we can see that there is a minor quality loss in cat lion images between the full-size model and ours, while the fur of the tiger head is less attractive. Images generated by $L1$ pruning and random pruning are visually worse on lions, tigers, dogs. Moreover, $L1$ generates cats with less favorable texture, and random pruning techniques generate cats with flaws on eyes. In general, the quality of dogs and wild animals among

Fig. 5. Comparison of generated images between the optimal pruned generator (top) and a randomly selected individual (bottom) on the AFHQ dataset.

Fig. 6. Comparison of generated images among the full-size model, $L1$ pruning, and random pruning on the AFHQ dataset ($L1$: $L1$ pruning, RC: random pruning).

Table 2. Comparisons of FID and FLOPs on AFHQ dataset.

Method	FLOPs	FID	CH_num
*	40.9 M	56.35	2544
Ours	13.91 M	**56.69**	1323
L1[39]	13.91 M	58.52	1323
Random[40]	13.91 M	59.49	1323

Note: * the original full-size model. **CH_num:** Number of active channels.

these four methods is not as good as the cats, due to the fact that there are less dog and wild images in the AFHQ dataset.

3.2. *Experiments on StyleGAN2*

We use StyleGAN2 on Flickr-Faces-HQ Dataset (FFHQ-256) dataset, which consists of 52,000 high-quality PNG images at 256×256 resolution. Note that the FFHQ-256 dataset we use does not contain class labels, as it consists of human faces with varying ages and genders. In the evolutionary search phase, we set the same parameters as that used in the StyleGAN model. After obtaining the optimal pruned generator, we continue to fine-tune the StyleGAN2 model by training it for an additional 450,000 iterations. We compare our method with state-of-the-art methods in many aspects, including the generated images and two quantified metrics, i.e. FLOPs, and the number of parameters.

The StyleGAN2 implementation is built upon the code from the repository at https://github.com/rosinality/stylegan2-pytorch.

3.2.1. *Results on FFHQ-256 Dataset*

Figure 8 presents images generated using the proposed ECP-StyleGANs. It should be noted that we only implemented our method on StyleGAN2. From the first two rows, we can observe that the generated images are of high quality, with almost no noticeable flaws. However, some less favorable results are shown in the third row, where issues such as incomplete faces (e.g. the little girl) and artifacts (e.g., the second-to-last man) persist. Additionally, there is a large black hole on the right side of the fifth mans face. The optimized pruned model successfully generates an image with two faces but fails to produce a natural and coherent face in the second person (see the first and last images), likely caused by aggressive pruning affecting deeper layers responsible for fine detail. The training curve for our optimal pruned model is shown in Fig. 7.

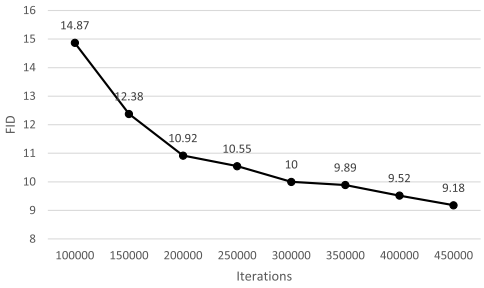

Fig. 7. A training curve of the optimally pruned model on StyleGAN2 over the course of iterations, with FID recorded every 100K iterations. The curve shows a steady improvement in the models performance over time, with the FID consistently decreasing as training progresses.

3.2.2. *Comparisons with SOTA*

To compare the performance of the optimal pruned model with other state-of-the-art approaches, we use FLOPs, the number of parameters, and FID as indicators. The results are shown in Table 3. The results demonstrate that model efficiency (measured in FLOPs and parameters) and image quality (FID) are inherently conflicting metrics, where reducing computational complexity often leads to a deterioration in image quality. CAGAN-heter has the lowest FLOPs and the number of parameters, but its FID is higher. Conversely, DGL-GAN has the lowest FID, but it increases the number of model parameters. Comparably, our method achieves compromising results between FID and model efficiency, with 4 × reduced on FLOPs and 3 × reduced on the model parameters. These results demonstrate that our method is a trade-off solution.

To further assess this trade-off, we visualize the performance of each model in a two-objective space considering different objectives, as shown in Fig. 9. Since some studies do not report FLOPs in their original papers, we separately visualize dominance relations based on (i) FLOPs *vs.* FID, and (ii) Params *vs.* FID. Specifically, we compute the hypervolume (HV)[44] to quantify the volume of the objective space dominated by the obtained pruning

Fig. 8. Images generated by the optimally pruned model on the FFHQ-256 dataset. The top two rows showcase high-quality samples with minimal visual artifacts. The bottom row presents less favorable results, with visible imperfections such as incomplete faces or slight distortions.

Table 3. Comparisons of FID, FLOPs(B), and the number of parameters(M) with the state-of-the-art methods on the FFHQ-256 dataset.

Method	FLOPs	FID	Params
*	45.1	4.5	30
CAGAN-heter[41]	**2.7**	13.75	**3.4**
Fast-GAN[42]	N/A	12.38	15.18
DGL-GAN[43]	N/A	**2.65**	30.37
Ours	10.5	9.18	10.2

Note: * the original full-size model.

solution with respect to a nadir/reference point. The higher the HV values the better the performance. Note that the reference point is set to (Params = 31, FID = 14, FLOPs = 46), which corresponds to the maximum observed value for each objective reported in Table 3, ensuring that the entire Pareto front is captured in the HV calculation.

As shown in Fig. 9, our method always yields higher the HV values compared with its peers, indicating a better trade-off between different objectives. Specifically, our method achieves the highest HV (100.26) in the Params-FID objective space and dominates Fast-GAN, demonstrating superior optimisation between model efficiency and image quality. While the full-size model (HV = 9.50) and DGL-GAN (HV = 7.15) achieve lower FIDs, they require significantly more computational resources caused by more parameters, making them impractical for edge devices and resource-constrained scenarios. Conversely, CAGAN-heter (HV = 6.90) with the lowest number of parameters can reduce the computational cost but at the expense of image quality, as indicated by its notably higher FID value.

Similarly, in the FLOPs-FID objective space, our method achieves an HV of 171.11, surpassing CAGAN-heter (10.82) and the full-size model (8.55). The full-size model delivers better image quality but requires excessive computational resources, while CAGAN-heter reduces computational demands but significantly compromises generation quality. Our approach offers a compromise, making it a compelling choice for efficient and high-quality image generation.

3.3. *Discussion*

In Secs. 2.1 and 3.1, we present the experimental results on StyleGAN and StyleGAN2, respectively. For StyleGAN, we transfer the pre-trained weights from a human face dataset to animal face datasets. Generally, direct transfer of a human-facepretrained model to animal face datasets results in poor FID scores due to the significant domain gap. In our experiments, the transfer is successful, generating visually plausible animal faces that highlight the strong feature representation capability of StyleGAN, as shown in Figs. 4 and 6. We attribute the improved performance primarily to an additional 50,000 post-search fine-tuning iterations, which enable effective adaptation of the model to animal-specific features. However, the FID score of the generated animal face images remains relatively high (ranging from 56 to 66, see Tables 1 and 2), in contrast to the much lower FID of 4.5 reported for human face images in the original paper.[9]

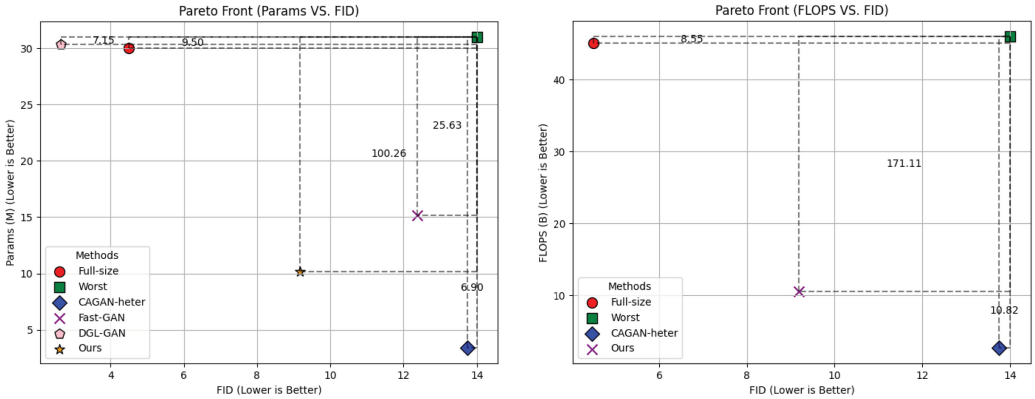

Fig. 9. Left: The Pareto Front showing the trade-off between the number of model parameters. Right: The Pareto Front illustrating the trade-off between FLOPS and FID. The number inside each rectangle represents its HV value, with higher the HV values indicating better performance.

Notably, few compression experiments have been conducted on the StyleGAN model. Therefore, we compare our method with $L1$ pruning and random pruning, both widely used as benchmarks in model compression studies.[45] Compared to the original full-size model, our method slightly increases the FID score but achieves a 3 reduction in FLOPs. We attribute this to the fact that the FFHQ-256 pre-trained model contains many redundant parameters for animal face datasets, which are considerably smaller in scale. When compared to $L1$ and random pruning, our method achieves the lowest FID scores, as shown in Tables 1 and 2. We believe this is because $L1$ pruning preserves only the weights with larger magnitudes, potentially ignoring smaller weights that may encode important fine-grained features such as texture details. This is evident in the generated images (Figs. 4 and 6), where $L1$ pruning leads to less visually appealing fur. In contrast, random pruning deactivates channels arbitrarily, which risks removing critical features. As shown in the same figures, random pruning results in missing details such as animal eyes and ears.

For StyleGAN2, we compare our method with traceable SOTA methods. As shown in Table 3, while DGL-GAN achieves the best performance in terms of FLOPs reduction and CAGAN-heter leads in FID score, our method yields a superior trade-off between the two objectives. This balanced performance is further confirmed by the HV metric that evaluates the trade-off performance in multi-objective optimisation, as illustrated in Fig. 9. This suggests that while some methods may yield optimal performance on a single objective, such as slightly smaller model size or better model performance, they generally sacrifice performance on the other objectives. In contrast, our approach maintains competitive image quality with significant model compression with the help of the proposed fitness function.

Computational cost and analysis. The search process requires 2.9 h on the AFHQ dataset, 2.1 h on the Animal Face dataset, and 3.8 h on the FFHQ-256 dataset per generation, primarily due to the evaluation of candidate pruned StyleGANs. Training the optimal pruned StyleGAN takes 12.5 h on AFHQ, 6.8 h on Animal Face, and 9 GPU days on FFHQ-256. The architecture search is conducted with 10 generations and a population size of 15, resulting in 150 candidate evaluations per experiment. Our search space sizes correspond to the number of whole channels: 2,544 (see Table 1) for StyleGAN and 5,889 (see Table 4) for StyleGAN2. This configuration explores roughly 6% of the StyleGAN search space and 2.5% of the StyleGAN2 search space. Given the per-generation costs (24 h depending on the dataset), the total search times are approximately 29 h, 21 h, and 38 h for AFHQ, Animal Face, and FFHQ-256, respectively. Most of this cost arises from partially training each of the 15 candidate pruned StyleGANs per generation to obtain stable performance estimates. Given our limited computational budget, this setup strikes a practical balance between exploration and runtime. After the search, the optimal pruned StyleGAN is trained from scratch, requiring 12.5 h on AFHQ, 6.8 h on Animal Face, and 9 GPU days on FFHQ-256.

3.4. *Ablation studies*

Finally, the ablation studies of the proposed method are presented. The aim is to validate the rationale behind the designed fitness function (Eq. (1)) in the evolutionary process. To this end, the models are optimized using only the FID score (FID) and only the number of activated channels (CH) separately, as shown in Table 4. The results show that, for both StyleGAN and StyleGAN2, using only the number of active channels as the fitness function leads to models with fewer active channels but higher FID scores. Conversely, when the FID score is used as the

Table 4. Ablation Studies on StyleGAN and StyleGAN2. **FC:** Fitness Function. **FID:** Uses only the FID value as the fitness function. **CH:** Uses only the number of active channels as the fitness function. **CH_num:** Number of active channels.

	StyleGAN				StyleGAN2	
	Animal Face		AHFQ		FFHQ	
FC	CH_num	FID	CH_num	FID	CH_num	FID
*	2544	58.84	2544	56.35	5889	4.5
Ours	1231	59.45	1323	56.69	2952	9.18
FID	1278	**59.40**	1236	**56.65**	3002	**9.02**
CH	**1185**	60.85	**1198**	57.86	**2854**	10.41

Note: * the original full-size model.

sole fitness function, the models tend to have a greater number of active channels, resulting in a smaller reduction. These findings indicate that balancing the number of active channels and the FID score provides a favourable trade-off.

4. Conclusion

This study introduces a novel approach to reducing the computational cost of StyleGAN and StyleGAN2 through evolutionary search, while preserving the high quality of the generative models. The method achieves this by identifying efficiently pruned models that strike an optimal balance between computational efficiency and image generation quality. Its effectiveness is validated using two animal face datasets (Animal Face and Animal FacesHQ) for StyleGAN and the FFHQ dataset for StyleGAN2, demonstrating a significant reduction in FLOPs and the number of parameters while maintaining competitive image generation performance. Experimental results indicate that the proposed method achieves robust performance across various scenarios, encompassing different base models and datasets.

While these results are promising, this study should be regarded as a preliminary step towards addressing generative model pruning and enhancing suitability for portable devices by formulating the task as a multi-objective optimisation problem and addressing it through methods that provide a reliable approximation of a Pareto set. In the experiments with StyleGAN2, artifact frequency across models was not formally quantified, with assessment relying instead on FID scores and qualitative inspection. Incorporating quantitative artifact metrics could provide additional valuable insights. Future work may explore human evaluations or structural similarity measures to better capture artifact prevalence and perceptual quality.

Future work will focus on extending the proposed method to address additional challenges in generative model pruning,[46] particularly by exploring advanced multi-objective optimisation[47] techniques to further balance computational efficiency and generation quality. Moreover, we aim to investigate the applicability of our approach to a wider range of generative models beyond StyleGAN and StyleGAN2, adapting the pruning strategy to architectures with different structures and objectives. Another important direction is to enhance the adaptability of pruned models for deployment on resource-constrained devices, such as mobile platforms and edge devices, ensuring efficient inference while maintaining high-quality image generation.[48] Finally, integrating knowledge distillation techniques[49,50] with evolutionary pruning could be explored to improve both model compression and performance.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 62376127, 61876089, 61876185, 61902281, and 61403206); the Natural Science Foundation of Jiangsu Province (No. BK20141005); and the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (No. 14KJB520025), Jiangsu Distinguished Professor Program.

ORCID

Yixia Zhang https://orcid.org/0009-0001-0064-7002
Ferrante Neri https://orcid.org/0000-0002-6100-6532
Xilu Wang https://orcid.org/0000-0002-0926-4454
Pengcheng Jiang https://orcid.org/0000-0002-9625-697X
Yu Xue https://orcid.org/0000-0002-9069-7547

References

1. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, Generative adversarial nets, *Adv Neural Inf Process Syst.* **27** (2014) 2672–2680.
2. J. Fu, H. Peng, B. Li, Z. Liu, R. Lugu, J. Wang and A. Ram'irez-de Arellano, Multitask adversarial networks based on extensive nonlinear spiking neuron models, *Int. J. Neural Syst.* **34**(6) (2024) 2450032.
3. L. Zhao, D. Song, W. Chen and Q. Kang, Coloring and fusing architectural sketches by combining a y-shaped generative adversarial network and a denoising diffusion implicit model, *Comput.-Aided Civ. Infrastruct. Eng.* **39**(7) (2024) 1003–1018.
4. S. Shim, Self-training approach for crack detection using synthesized crack images based on conditional

generative adversarial network, *Comput.-Aided Civ. Infrastruct. Eng.* **39**(7) (2024) 1019–1041.
5. Y. Zhang and L. Zhang, A generative adversarial network approach for removing motion blur in the automatic detection of pavement cracks, *Comput.-Aided Civ. Infrastruct. Eng.* **39**(22) (2024) 3412–3434.
6. T. Chakraborty, U. R. KS, S. M. Naik, M. Panja and B. Manvitha, Ten years of generative adversarial nets (GANs): a survey of the state-of-the-art, *Mach. Learn.: Sci. Technol.* **5**(1) (2024) 011001.
7. M. Durgadevi *et al.*, Generative adversarial network (GAN): A general review on different variants of GAN and applications, in *Proc. 6th Int. Conf. Communication and Electronics Systems* (IEEE, 2021), pp. 1–8.
8. W. Gao and M. Li, Evolution of discriminator and generator gradients in GAN training: From fitting to collapse, *Trans. Mach. Learn. Res.* (2024), https://openreview.net/forum?id=58gPkcVbFL.
9. T. Karras, A style-based generator architecture for generative adversarial networks, preprint (2019), arXiv:1812.04948.
10. T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen and T. Aila, Analyzing and improving the image quality of StyleGAN, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, (IEEE, 2020), pp. 8110–8119.
11. A. Melnik, M. Miasayedzenkau, D. Makaravets, D. Pirshtuk, E. Akbulut, D. Holzmann, T. Renusch, G. Reichert and H. Ritter, Face generation and editing with StyleGAN: A survey, *IEEE Trans. Pattern Anal. Mach. Intell.* **46**(5) (2024) 3557–3576.
12. I. Choi, S. Park and J. Park, Generating and modifying high resolution fashion model image using StyleGAN, in *Proc. 13th Int. Conf. Information and Communication Technology Convergence*, (IEEE, 2022), pp. 1536–1538.
13. H. Cheng, M. Zhang and J. Q. Shi, A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations, *IEEE Trans. Pattern Anal. Mach. Intell.* **46** (2024) 10558–10578.
14. K. M. R. Alam, N. Siddique and H. Adeli, A dynamic ensemble learning algorithm for neural networks, *Neural Comput. Appl.* **32**(12) (2020) 8675–8690.
15. Y. He and L. Xiao, Structured pruning for deep convolutional neural networks: A survey, *IEEE Trans. Pattern Anal. Mach. Intell.* **46**(5) (2023) 2900–2919.
16. P. Isola, J.-Y. Zhu, T. Zhou and A. A. Efros, Image-to-image translation with conditional adversarial networks, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (IEEE, 2017), pp. 1125–1134.
17. J.-Y. Zhu, T. Park, P. Isola and A. A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in *Proc. IEEE Int. Conf. Computer Vision* (IEEE, 2017), pp. 2223–2232.
18. J. Chung, S. Hyun, S.-H. Shim and J.-P. Heo, Diversity-aware channel pruning for StyleGAN compression, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition* (IEEE, 2024), pp. 7902–7911.
19. Y. Liu, Z. Shu, Y. Li, Z. Lin, F. Perazzi and S.-Y. Kung, Content-aware GAN compression, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition* (IEEE, 2021), pp. 12156–12166.
20. S. Wu, H. Du, Y. Xiong, S. Chen, T.-W. Kuo, N. Guan and C. J. Xue, EvoP: Robust LLM inference via evolutionary pruning, preprint (2025), arXiv:2502.14910.
21. P. Jiang, Y. Xue and F. Neri, Convolutional neural network pruning based on multi-objective feature map selection for image classification, *Appl. Soft Comput.* **139** (2023) 110229.
22. J. Zhang, L. Zhang, Y. Wang, J. Wang, X. Wei and W. Liu, An efficient multi-objective evolutionary zero-shot neural architecture search framework for image classification, *Int. J. Neural Syst.* **33**(5) (2023) 2350016.
23. S. J. Simske, D. Li and J. S. Aronoff, A statistical method for binary classification of images, in *Proc. ACM Symp. Document Engineering* (ACM, 2005), pp. 127–129.
24. H. Adeli and S. Kumar, Concurrent structural optimization on massively parallel supercomputer, *J. Struct. Eng.* **121**(11) (1995) 1588–1597.
25. H. Kim and H. Adeli, Discrete cost optimization of composite floors using a floating-point genetic algorithm, *Eng. Optim.* **33**(4) (2001) 485–501.
26. M. Kociecki and H. Adeli, Two-phase genetic algorithm for topology optimization of free-form steel space-frame roof structures with complex curvatures, *Eng. Appl. Artif. Intell.* **32** (2014) 218–227.
27. P. Liao, X. Wang, Y. Jin and W. Du, MO-EMT-NAS: Multi-objective continuous transfer of architectural knowledge between tasks from different datasets, in *Computer Vision — ECCV 2024* (Springer Nature Switzerland, Cham, 2025), pp. 18–35.
28. Y. Xue, W. Tong, F. Neri, P. Chen, T. Luo, L. Zhen and X. Wang, Evolutionary architecture search for generative adversarial networks based on weight sharing, *IEEE Trans. Evol. Comput.* **28**(3) (2024) 653–667.
29. S. M. Lim, A. B. M. Sultan, M. N. Sulaiman, A. Mustapha and K. Y. Leong, Crossover and mutation operators of genetic algorithms, *Int. J. Mach. Learn. Comput.* **7**(1) (2017) 9–12.
30. A. V. Lotov and K. Miettinen, Visualizing the Pareto frontier, in *Multiobjective Optimization: Interactive and Evolutionary Approaches* (Springer, 2008), pp. 213–243.
31. Y. Jin, A comprehensive survey of fitness approximation in evolutionary computation, *Soft Comput.* **9**(1) (2005) 3–12.
32. M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler and S. Hochreiter, GANs trained by a two time-scale update rule converge to a local Nash equilibrium, *Adv. Neural Inf. Process Syst.* **30** (2017) 6626–6637.

33. M. Gutierrez Soto and H. Adeli, Many-objective control optimization of high-rise building structures using replicator dynamics and neural dynamics model, *Struct. Multidisc. Optim.* **56**(6) (2017) 1521–1537.
34. D. Rodrigues, J. P. Papa and H. Adeli, Meta-heuristic multi-and many-objective optimization techniques for solution of machine learning problems, *Expert Syst.* **34**(6) (2017) e12255.
35. A. Lipowski and D. Lipowska, Roulette-wheel selection via stochastic acceptance, *Physica A* **391**(6) (2012) 2193–2196.
36. C. W. Ahn and R. S. Ramakrishna, Elitism-based compact genetic algorithms, *IEEE Trans. Evol. Comput.* **7**(4) (2003) 367–385.
37. Z. Si and S.-C. Zhu, Learning hybrid image templates (HIT) by information projection, *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(7) (2011) 1354–1367.
38. Y. Choi, Y. Uh, J. Yoo and J.-W. Ha, StarGAN v2: Diverse image synthesis for multiple domains, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition* (IEEE, 2020), pp. 8188–8197.
39. C. Yang, Z. Yang, A. M. Khattak, L. Yang, W. Zhang, W. Gao and M. Wang, Structured pruning of convolutional neural networks via L1 regularization, *IEEE Access* **7** (2019) 106385–106394.
40. Y. Li, K. Adamczewski, W. Li, S. Gu, R. Timofte and L. Van Gool, Revisiting random channel pruning for neural network compression, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition* (IEEE, 2022), pp. 191–201.
41. G. Xu, Y. Hou, Z. Liu and C. C. Loy, Mind the gap in distilling StyleGANs, in *European Conf. Computer Vision* (Springer, 2022), pp. 423–439.
42. B. Liu, Y. Zhu, K. Song and A. Elgammal, Towards faster and stabilized GAN training for high-fidelity few-shot image synthesis, preprint (2021), arXiv:2101.04775.
43. Y. Tian, L. Shen, X. Tian, D. Tao, Z. Li, W. Liu and Y. Chen, DGL-GAN: discriminator-guided GAN compression, *Vis. Comput.* **41**(2024) 1–22.
44. A. P. Guerreiro, C. M. Fonseca and L. Paquete, The hypervolume indicator: Computational problems and algorithms, *ACM Comput. Surv.* **54**(6) (2021) 1–42.
45. T. Choudhary, V. Mishra, A. Goswami and J. Sarangapani, A comprehensive survey on model compression and acceleration, *Artif. Intell. Rev.* **53** (2020) 5113–5155.
46. W. Wang, W. Chen, Y. Luo, Y. Long, Z. Lin, L. Zhang, B. Lin, D. Cai and X. He, Model compression and efficient inference for large language models: A survey, preprint (2024), arXiv:2402.09748.
47. S. Jiang, J. Zou, S. Yang and X. Yao, Evolutionary dynamic multi-objective optimisation: A survey, *ACM Comput. Surv.* **55**(4) (2022) 1–47.
48. H.-I. Liu, M. Galindo, H. Xie, L.-K. Wong, H.-H. Shuai, Y.-H. Li and W.-H. Cheng, Lightweight deep learning for resource-constrained environments: A survey, *ACM Comput. Surv.* **56**(10) (2024) 1–42.
49. J. Gou, B. Yu, S. J. Maybank and D. Tao, Knowledge distillation: A survey, *Int. J. Comput. Vision* **129**(6) (2021) 1789–1819.
50. Y. Xue, Y. Lin and F. Neri, Architecture knowledge distillation for evolutionary generative adversarial network, *Int. J. Neural Syst.* **35**(4) (2025) 2550013.