Progressive Training of Generative Diffusion Models with Fuzzy Growth for Higher Fidelity

1st Michał Wieczorek
Faculty of Applied Mathematics
Silesian University of Technology
Gliwice, Poland
michal.wieczorek@polsl.pl

2nd Alicja Polowczyk
Faculty of Applied Mathematics
Silesian University of Technology
Gliwice, Poland
alicjapolowczyk47@gmail.com

3rd Agnieszka Polowczyk
Faculty of Applied Mathematics
Silesian University of Technology
Gliwice, Poland
agnieszkapolowczyk11@gmail.com

4th Jakub Siłka
Faculty of Applied Mathematics
Silesian University of Technology
Gliwice, Poland
jakub.silka@polsl.pl

***Abstract*—Training diffusion models for high-resolution image generation remains computationally intensive and data-demanding due to the quadratic scaling of complexity with image dimensions. We present a progressive training framework that systematically increases spatial resolution during training with fuzzy based growth module, enabling efficient synthesis of high-quality images while significantly reducing computational overhead. Our approach employs a conditional U-Net architecture operating in a latent diffusion space, augmented with resolution-specific modules that are gradually activated through smooth fuzzy transition mechanisms. The model initiates training at low resolutions to establish structural priors, then progressively grows to higher resolutions while transferring learned representations through weight inheritance and controlled blending based on fuzzy logic gating. This progressive strategy reduces total training time by up to 60% compared to direct high-resolution training while maintaining comparable loss scores on the same dataset. Furthermore, the method demonstrates improved data efficiency by leveraging low-resolution pretraining as a regularizer for high-frequency detail learning. We provide comprehensive experimental validation on resolutions from 8x8 to 256x256 pixels, along with theoretical analysis of the progressive learning dynamics. Our work offers a practical pathway for scalable high-resolution image generation with diffusion models, balancing computational efficiency with sample quality.**

***Index Terms*—Machine Learning, Artificial Intelligence, Diffusion, Generative AI**

I. Introduction

Training diffusion models directly at the target image resolution is computationally expensive and often inefficient. Memory and computation grow approximately quadratically with spatial resolution, while early optimization still focuses mainly on coarse structure that can be learned at much lower cost. Progressive growing addresses this issue by allowing the model to learn low-frequency structure first and introduce high-frequency detail later. The main difficulty is then the transition between scales: when new resolution-specific modules are activated, an abrupt switch can destabilize optimization.

In this work, we treat the transition between scales as an uncertainty-aware control problem. During diffusion training, the information available at a growth step, such as stage progress and mini-batch gradient strength, is noisy because it depends on sampled timesteps, injected noise, and batch composition. Instead of relying on a fixed deterministic fade-in schedule, we define the blending gate with interval Type-2 fuzzy logic [1], [2]. Compared with Type-1 fuzzy sets, interval Type-2 sets explicitly model uncertainty in membership functions and therefore provide more conservative behavior when gradient evidence is ambiguous. When the uncertainty intervals collapse, the controller reduces to a deterministic fade-in schedule.

Diffusion probabilistic models are currently among the most effective generative approaches for image synthesis [3], [4]. They offer stable training and high-quality generation, but high-resolution training remains costly. Latent diffusion reduces this burden by operating in a compressed representation space [5], yet training at high resolutions is still resource intensive.

We therefore propose a progressive conditional latent diffusion framework in which resolution-specific heads and tails are activated stage by stage, inherited weights are reused across scales, and transitions are controlled by a fuzzy gate. The method combines three key ideas:

1) resolution-specific modular heads and tails activated progressively,
2) smooth fade-in transitions for stable scale changes,
3) weight inheritance across scales to reduce high-resolution training cost.

II. Dataset

We use the open-access Kaggle "Cats & Dogs" dataset. For training, we use 4000 cat images and 4000 dog images from the training split. Images depict animals in varied poses, lighting conditions, and backgrounds.

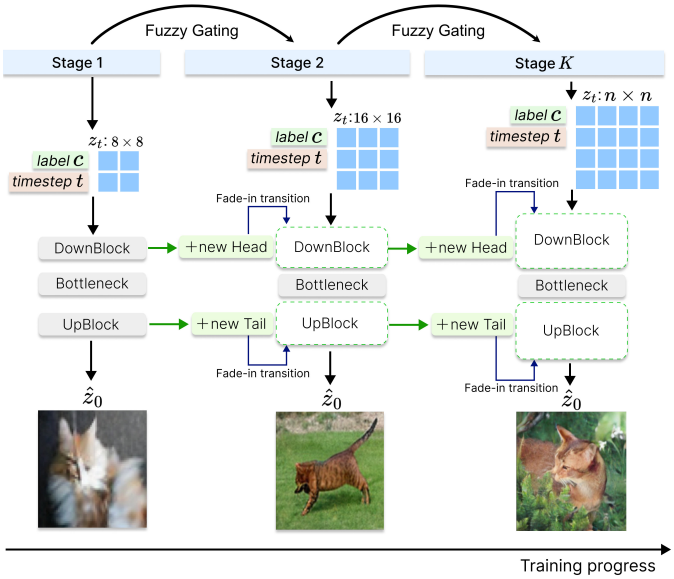

Fig. 1: **Overview of the training process of our progressive conditional latent diffusion model.** Training starts at a low latent resolution. At each subsequent stage, new head and tail layers are added to increase depth, and a fade-in transition gradually increases α_u and α_d, enabling a smooth shift to higher resolutions.

A. Data Pre-processing

All images are resized to the required resolution for the current stage and normalized from the $[0, 255]$ range to $[-1, 1]$. No additional data augmentation is used.

III. Proposed Solution

We propose a conditional latent diffusion model trained with progressive growing. The model starts from low-resolution latents and gradually activates resolution-specific modules, allowing it to learn global structure first and refine fine details later. This reduces computational cost and improves training stability.

A. Model Architecture

The generator is a time-conditional U-Net operating in VAE latent space. The network takes a noisy latent $\mathbf{x}_t$, timestep t, and class label $\mathbf{c}$, and predicts either the clean latent $\mathbf{x}_0$ or the noise term ϵ.

The architecture contains:

- **Embedding layers:** sinusoidal time embeddings and learned label embeddings injected into residual blocks,
- **Residual blocks:** group normalization, Swish activation, and two convolutions,
- **Attention blocks:** self-attention at lower resolutions,
- **Progressive heads and tails:** resolution-specific input and output layers activated according to the current growth stage.

A frozen VAE encodes images into a 4-channel latent space, which lowers computational cost while preserving perceptual structure.nd the diffusion process is applied in this compressed space. This reduces computational cost while preserving perceptual quality.

B. Diffusion Process

We adopt the standard Gaussian diffusion framework. Given a clean latent $\mathbf{x}_0$, the forward process produces noisy latents $\mathbf{x}_t$ for $t = 1, \ldots, T$:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \tag{1}$$

where $\epsilon \sim \mathcal{N}(0, I)$ and $\bar{\alpha}_t$ is the cumulative product of noise schedules $\alpha_t = 1 - \beta_t$. The model is trained to predict either ϵ (noise) or $\mathbf{x}_0$ (clean latent) via a mean-squared error loss. Reverse sampling is performed using either ancestral DDPM sampling or deterministic DDIM sampling, with optional classifier-free guidance to enhance conditional generation.

C. Progressive Growing Strategy

The proposed training schedule increases resolution incrementally. The model is first optimized on low-resolution latents, after which an additional head and tail pair is activated for the next scale. The inherited path and the newly activated path are blended during a short transition window instead of switching abruptly. For a fade-in stage with local step index $s \in \{1, \ldots, S_i\}$, the downsampling and upsampling blends are

$$\mathbf{h}_s^{\downarrow} = (1 - \alpha_{d,s})\mathbf{h}_{\text{old}}^{\downarrow} + \alpha_{d,s}\mathbf{h}_{\text{new}}^{\downarrow}, \tag{2}$$

$$\mathbf{h}_s^{\uparrow} = (1 - \alpha_{u,s})\mathbf{h}_{\text{old}}^{\uparrow} + \alpha_{u,s}\mathbf{h}_{\text{new}}^{\uparrow}, \tag{3}$$

where $\alpha_{d,s}, \alpha_{u,s} \in [0, 1]$ are gating coefficients. Equations (2)–(3) make the fade-in mechanism explicit before introducing the fuzzy controller that determines the coefficients.

The growing procedure has three components:

1) **Weight inheritance:** parameters learned at resolution r_{i-1} are reused when growing to r_i.
2) **Smooth transition:** old and new paths are blended instead of replaced in a single step.
3) **Gradual depth increase:** new downsampling and upsampling modules are activated only when entering the next stage.

The overall training scheme of our progressive conditional latent diffusion model is illustrated in Fig. 1.

D. Interval Type-2 Fuzzy Gating

We model the fade-in coefficient as a function of two normalized inputs:

$$p_s = \frac{s}{S_i}, \tag{4}$$

$$G_s = \|\nabla_{\theta_{\text{old}}}\mathcal{L}_s\|_2, \qquad g_s = \text{clip}\left(\frac{G_s}{\bar{G}_s + \varepsilon}, 0, 1\right), \tag{5}$$

where p_s is the growth progress and g_s is the normalized gradient activity of the inherited branch. The running baseline $\bar{G}_s$ can be estimated with an exponential moving average,

$$\bar{G}_s = \lambda\bar{G}_{s-1} + (1 - \lambda)G_s, \tag{6}$$

Input	Term	$(\overline{\mu}(x), \underline{\mu}(x))$
p_s	Early	$\mathrm{trap}(x; 0, 0, 0.20, 0.45), \mathrm{trap}(x; 0, 0, 0.10, 0.35)$
p_s	Mid	$\mathrm{tri}(x; 0.20, 0.50, 0.80), \mathrm{tri}(x; 0.30, 0.50, 0.70)$
p_s	Late	$\mathrm{trap}(x; 0.55, 0.80, 1, 1), \mathrm{trap}(x; 0.65, 0.90, 1, 1)$
g_s	Low	$\mathrm{trap}(x; 0, 0, 0.15, 0.40), \mathrm{trap}(x; 0, 0, 0.10, 0.30)$
g_s	Medium	$\mathrm{tri}(x; 0.20, 0.50, 0.80), \mathrm{tri}(x; 0.30, 0.50, 0.70)$
g_s	High	$\mathrm{trap}(x; 0.60, 0.80, 1, 1), \mathrm{trap}(x; 0.70, 0.90, 1, 1)$

TABLE I: Upper and lower membership functions for the interval Type-2 fuzzy gate.

Progress \ Gradient	Low	Medium	High
Early	Medium	Low	VeryLow
Mid	High	Medium	Low
Late	VeryHigh	High	Medium

TABLE II: Rule base for the interval Type-2 gate.

with $\lambda \in [0, 1)$. Intuitively, high g_s indicates unstable or rapidly changing gradients, in which case the controller should be conservative.

For readability we define triangular and trapezoidal membership functions as

$$\mathrm{tri}(x; a, b, c) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right), \quad (7)$$

$$\mathrm{trap}(x; a, b, c, d) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right). \quad (8)$$

Each linguistic term is represented by an interval Type-2 set with upper and lower membership functions. The footprint of uncertainty encodes ambiguity in the exact boundaries of "early", "late", "low gradient", or "high gradient" regimes. Table I lists the membership functions used in the controller.

The output variable is the desired blend weight $\hat{\alpha}_s \in [0, 1]$. We use five interval consequents:

$$\begin{aligned} \text{VeryLow} &= [0.00, 0.15], \quad \text{Low} = [0.10, 0.35], \\ \text{Medium} &= [0.30, 0.60], \quad \text{High} = [0.55, 0.85], \\ \text{VeryHigh} &= [0.80, 1.00]. \end{aligned}$$

The full rule base is given in Table II. It follows a simple principle: early transitions and high gradients should keep the gate small, whereas late transitions and low gradients should favor a larger weight for newly activated modules.

Using product inference, the firing strength of rule k is an interval

$$\underline{w}_k = \prod_{m \in \{p,g\}} \underline{\mu}_{A_{k,m}}(x_m), \qquad \overline{w}_k = \prod_{m \in \{p,g\}} \overline{\mu}_{A_{k,m}}(x_m), \quad (9)$$

where $x_p = p_s$ and $x_g = g_s$. Let the K rule consequents be sorted by their centroids $c_k = (c_k^- + c_k^+)/2$. Karnik-Mendel type-reduction produces an interval output $[y_l, y_r]$:

$$y_l = \frac{\sum_{k=1}^{L} \overline{w}_k c_k^- + \sum_{k=L+1}^{K} \underline{w}_k c_k^-}{\sum_{k=1}^{L} \overline{w}_k + \sum_{k=L+1}^{K} \underline{w}_k}, \quad (10)$$

$$y_r = \frac{\sum_{k=1}^{R} \underline{w}_k c_k^+ + \sum_{k=R+1}^{K} \overline{w}_k c_k^+}{\sum_{k=1}^{R} \underline{w}_k + \sum_{k=R+1}^{K} \overline{w}_k}, \quad (11)$$

where L and R are the switch points found by the iterative KM procedure. The crisp gate is then

$$\hat{\alpha}_s = \frac{y_l + y_r}{2}, \qquad \alpha_s = \max(\alpha_{s-1}, \hat{\alpha}_s). \quad (12)$$

In practice we use the same monotone gate for both paths, i.e. $\alpha_{d,s} = \alpha_{u,s} = \alpha_s$, although separate gates can also be used. If the upper and lower membership functions coincide and the consequent intervals collapse to singletons, the formulation becomes Type-1 fuzzy logic. If the rule outputs additionally follow a fixed deterministic ramp, it reduces to ordinary linear fade-in.

E. Advantages of Progressive Training

Progressive growing improves efficiency in three ways:

- **Efficient learning:** low-resolution stages learn global structure quickly and transfer it to higher resolutions,
- **Stable optimization:** early stages are cheaper and easier to optimize,
- **Data efficiency:** high-resolution stages focus on details instead of relearning coarse structure.

F. Training Details

Training follows the standard diffusion objective with the addition of progressive growing transitions. We use the AdamW optimizer with cosine annealing and warm-up scheduling. Gradient norms are monitored and optionally clipped or normalized to prevent instability. Classifier-free guidance is employed during sampling to improve conditional fidelity. The entire pipeline is implemented in PyTorch and can be adapted to various datasets by adjusting the resolution progression and channel multipliers.

This progressive diffusion model effectively combines the generative capabilities of diffusion processes with the training efficiency of progressive growing, enabling scalable high-resolution image synthesis with reduced computational cost.

IV. Proposed Network Training

A. Algorithm Description

The progressive training algorithm for conditional diffusion models is formalized in Algorithm 1. The approach systematically increases image resolution while preserving learned representations through weight inheritance and smooth transitions.

B. Algorithm Components

Progressive Resolution Schedule: The training iterates through increasing resolutions $r_1 < r_2 < \cdots < r_n$, where each r_i is double the previous. At each stage, the model learns features appropriate for the current scale.

Network Growing Mechanism: When transitioning from resolution r_{i-1} to r_i:

- Additional convolutional layers are activated in the U-Net's head and tail modules
- A smooth fade-in transition blends outputs from previous and current scales using α = fade_iters as number of

Algorithm 1 Progressive Training of Conditional Diffusion Model

Require: Dataset $\mathcal{D}$, resolution schedule $\mathcal{R} = [r_1, r_2, \ldots, r_n]$
Require: Channel multipliers $\mathcal{C}$, batch sizes $\mathcal{B}$, epochs $\mathcal{E}$
Require: Diffusion parameters (β_1, β_T, T), guidance weight w
Ensure: Trained model M_{final} capable of generating images at resolution r_n

Initialize U-Net model M with depth $= 0$, $\alpha_d = \alpha_u = 1$
Initialize VAE encoder/decoder $\mathcal{V}$ (frozen)
for $i \leftarrow 1$ **to** n **do**
  $r_i \leftarrow \mathcal{R}[i]$ {Current target resolution}
  $b_i \leftarrow \mathcal{B}[i]$ {Batch size for current resolution}
  if $i > 1$ **then**
    Grow Network:
    M.depth $\leftarrow M$.depth $+ 1$
    M.fade_iters $\leftarrow 1/(5 \times |\mathcal{D}|/b_i)$
    $M.\alpha_d \leftarrow FuzzyGate(M.weights, M.\text{fade\_iters})$
    Activate additional head/tail layers for resolution r_i
  end if
  Initialize dataloader $\mathcal{L}_i$ with resolution r_i, batch size b_i
  Initialize optimizer $\mathcal{O}$ with learning rate η_i
  for epoch $\leftarrow 1$ **to** $\mathcal{E}_i$ **do**
    for $(\mathbf{x}, \mathbf{c}) \in \mathcal{L}_i$ **do**
      Forward Diffusion: {Image $\mathbf{x}$, label $\mathbf{c}$}
      $\mathbf{z}_0 \leftarrow \mathcal{V}_{\text{encode}}(\mathbf{x})$ {Encode to latent space}
      $t \sim \mathcal{U}\{1, T\}$ {Sample timestep}
      $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
      $\mathbf{z}_t \leftarrow \sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$
      Conditional Prediction:
      $\hat{\mathbf{z}}_0 \leftarrow M(\mathbf{z}_t, t, \mathbf{c})$ {Model predicts clean latent}
      Loss Computation:
      $\mathcal{L} \leftarrow \|\hat{\mathbf{z}}_0 - \mathbf{z}_0\|_2^2$ {MSE in latent space}
      Backward Pass with Gradient Control:
      $\nabla\mathcal{L} \leftarrow \text{Backward}(\mathcal{L})$
      if grad_normalize **then**
        $g_{norm} \leftarrow \|\nabla\mathcal{L}\|_2$
        $\nabla\mathcal{L} \leftarrow \nabla\mathcal{L} \times \min(1, \tau/g_{norm})$ {Scale if $g_{norm} > \tau$}
      else if grad_clip **then**
        $\nabla\mathcal{L} \leftarrow \text{Clip}(\nabla\mathcal{L}, \lambda)$ {Clipping threshold λ}
      end if
      $\mathcal{O}$.step() {Parameter update}
      Fade-in Update:
      if $i > 1$ **and** epoch $\leq$ fade_epochs **then**
        $M.\alpha_d \leftarrow M.\alpha_d + M$.fade_iters
        $M.\alpha_u \leftarrow M.\alpha_u + M$.fade_iters
      end if
    end for
    $\mathcal{O}$.lr_schedule() {Learning rate adjustment}
  end for
  Save model checkpoint M_{r_i}
end for
return M_{r_n} {Final model at highest resolution}

substeps and weights of the previous stage, from which the gradient strength is taken for later dynamic fuzzy gating based on Type-2 Fuzzy Logic.

The gating mechanism is defined using a set of fuzzy rules:

$$R_i: \text{ IF } x_1 \text{ is } \tilde{A}_{i1} \text{ AND } \ldots \text{ AND } x_n \text{ is } \tilde{A}_{in} \text{ THEN } g \text{ is } \tilde{B}_i \tag{13}$$

where $\tilde{A}_{ij}$ and $\tilde{B}_i$ are interval Type-2 fuzzy sets and are based on growth substep iterator and previous model gradient strength for a given batch of images.

The firing strength of each rule is computed as an interval:

$$w_i = [\underline{w}_i, \overline{w}_i] \tag{14}$$

Type-reduction is performed using the Karnik-Mendel (KM) algorithm to obtain an interval output. The crisp gating output is obtained by averaging the type-reduced interval giving us the output weight α_d.

Training at Each Resolution: For each resolution r_i:

- Images are encoded to latent space using a frozen VAE: $\mathbf{z} = \mathcal{V}_{\text{encode}}(\mathbf{x})$
- The diffusion process adds noise to latents: $\mathbf{z}_t = \sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$
- The model learns to predict clean latents $\hat{\mathbf{z}}_0$ conditioned on class labels $\mathbf{c}$
- Gradient normalization/clipping stabilizes training across resolutions

Sampling Procedure: After training at resolution r_i, images can be generated via algorithm 2.

Algorithm 2 Sampling Procedure

$\mathbf{z}_T \sim \mathcal{N}(0, \mathbf{I})$ {Sample noise in latent space}
for $t \leftarrow T$ to 1 **do**
  $\hat{\mathbf{z}}_0 \leftarrow M(\mathbf{z}_t, t, \mathbf{c})$ {Predict clean latent}
  $\mathbf{z}_{t-1} \leftarrow \text{DDIM}(\mathbf{z}_t, \hat{\mathbf{z}}_0, t)$ {Reverse diffusion step}
end for
$\mathbf{x}_{gen} \leftarrow \mathcal{V}_{\text{decode}}(\mathbf{z}_0)$ {Decode to image space}

C. Complexity Analysis

The progressive approach reduces computational complexity compared to direct high-resolution training:

- **Memory:** Training at lower resolutions requires $\mathcal{O}(r_i^2)$ memory per image, significantly less than $\mathcal{O}(r_n^2)$
- **Time:** Early stages converge rapidly as they learn coarse structures; later stages focus on fine details
- **Data Efficiency:** Transfer learning between resolutions reduces required high-resolution samples by leveraging low-resolution priors

The algorithm implements progressive knowledge transfer where low-resolution training establishes structural priors that accelerate learning of high-frequency details at higher resolutions.

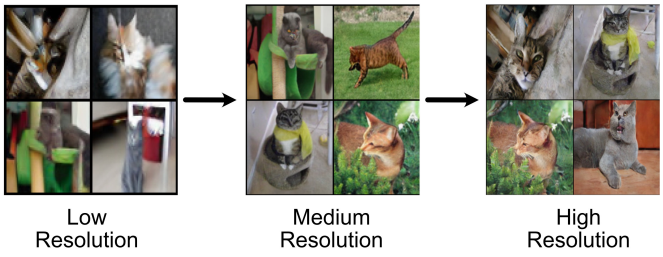

Fig. 2: Results of each stage training. As seen, on lower resolutions we can see the model generates overall correct shapes but with low quality of details, while in each following stage it learns more high frequency details and generates more coherent results.

Model	MSE Loss	FID	Total training time
Proposed	0.0034	2.56	12h
Standard	0.0465	7.21	30h

TABLE III: Comparison with standard training on the same dataset

V. Results

The training was performed on a single Mac Studio M3 Ultra with 512GB of Unified Memory. Final training took around 12h throughout all stages. Standard diffusion training has been performed for 30h on the same hardware. Both training setups used identical model and dataset with the only difference being the training method. Results of this training can be found in Fig. 3, Fig. 2 and Tab. III. As we can see, the results of the proposed method are of higher visual and numerical quality even considering much shorter training times. In the last subfigure we can see a test run on CelebA dataset, showing high quality output.

Experimental results demonstrate that the proposed progressive training strategy provides substantial advantages over classical single stage high resolution diffusion training. The most important outcome is reduction in training time. Proposed progressive model reached convergence in approximately 12 hours, compared to 30 hours required by the standard approach when trained on the same hardware, dataset and architecture. This 60% decrease in runtime is especially important since progressive model not only trains faster but also achieves superior reconstruction quality and lower loss (0.0034 vs. 0.0465). These results support our hypothesis that low resolution pretraining establishes strong structural background that accelerates subsequent learning at higher resolutions.

Generated samples reinforce this trend. At the final stage, progressive model produces images with sharper edges, cleaner textures and fewer high frequency artifacts compared to model trained directly in high resolution. This suggests that proposed learning process allows the network to first internalize coarse semantic information (shapes, contours, object placement) without being overwhelmed by fine grained

(a) Single high resolution training for 3x more epochs than the proposed solution.

(b) Final result of the progressive training.

(c) Final result of the progressive training on CelebA dataset.

Fig. 3: Comparison of classical training vs progressive training. As seen the classical approach provides mixed results with some images lacking significant amount of details, both in terms of the overall shape and high frequency details, while the proposed solution provides good looking results.

noise patterns that dominate high resolution inputs. Only once these global structures are stabilized the model introduce higher resolution layers, enabling it to refine local details more effectively.

Another important observation is that training stability significantly improved during progressive stages. Classical high resolution diffusion training frequently suffers from gradient spikes, slow convergence of early epochs and difficulties capturing global structure. In contrast, proposed novel progressive model benefits from consistently smooth loss curves and reduced gradient variance due to the lower resolution early stages requiring simpler patterns to be learned first.

Proposed progressive approach also show evidence of better data efficiency, despite using the same dataset. Since high resolution data often contains redundant low frequency information already learned in earlier stages. Our model can allocate more of its high resolution training capacity to genuinely new information such as textures, details and lighting variations. This helps explain why the model converges faster in later stages and produces cleaner high frequency detail than baseline even if both models are exposed to identical images at full resolution. Experimental results show that progressive training is not a computational improvement but a fundamental improvement in learning dynamics of diffusion models. It redefines how these models should be trained, particularly when resources are constrained. We have demonstrated that starting simple and increasing complexity over time gives both faster and better outcomes.

VI. Related Works

Generative Adversarial Networks (GANs) have been the dominant paradigm in image generation for many years, where a discriminator guides the training of the generator. They are used in class-conditional settings, such as BigGANs [6], where conditioning on class labels significantly improves the fidelity of generated images on challenging datasets, such as ImageNet [7]. To enable high-quality image generation, ProGAN [8] was proposed, which utilizes progressive learning strategies. However, GANs suffer from training stability issues, high sensitivity to hyperparameter selection, and limited data coverage. To address these limitations, diffusion models were proposed [3], which introduce the idea of iterative latent denoising starting from random Gaussian noise. Conditional guidance methods, including classifier guidance [9], have been introduced to steer the inference process. However, despite iterative denoising, the high computational cost of the training process remains a significant limitation. Accelerated sampling methods, such as DDIM [10] or cascaded approaches [11], have been proposed. In our approach, we introduce a progressive learning strategy for the diffusion model, which enables faster and more efficient learning.

VII. Conclusions

In this paper, we introduced a novel progressive training framework for generative diffusion models, designed to significantly reduce computational burden of high resolution training while preserving or even improving output quality. Unlike standard diffusion training pipelines that operate directly at the target resolution, our method systematically grows the model across increasing spatial scales, enabling it to first learn coarse structural priors and then refine them with high frequency details. This hierarchical approach mirrors human visual learning and builds upon curriculum learning principles, but uniquely adapts them to the multiscale generative nature of diffusion models.

Experimental results demonstrated that the proposed progressive diffusion strategy achieves higher sample quality, lower loss values, and up to 60% shorter training times compared to classical approaches, even on a modest dataset such as Cats vs Dogs. These results confirm that progressive resolution learning offers a powerful and data efficient alternative to standard diffusion training, especially for practitioners with limited computational resources.

VIII. Acknowledgment

The authors would like to acknowledge the contribution to this research from the Polish Ministry of Education and Science under grant No. PN/02/0084/2023.

References

[1] N. N. Karnik, J. M. Mendel, and Q. Liang, "Type-2 fuzzy logic systems," *IEEE transactions on Fuzzy Systems*, vol. 7, no. 6, pp. 643–658, 1999.

[2] J. M. Mendel and R. B. John, "Type-2 fuzzy sets made simple," *IEEE Transactions on fuzzy systems*, vol. 10, no. 2, pp. 117–127, 2002.

[3] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[4] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International conference on machine learning*. pmlr, 2015, pp. 2256–2265.

[5] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[6] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

[7] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 7354–7363.

[8] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.

[9] P. Dhariwal and A. Q. Nichol, "Diffusion models beat gans on image synthesis," in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021, pp. 8780–8794.

[10] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[11] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans, "Cascaded diffusion models for high fidelity image generation," *J. Mach. Learn. Res.*, vol. 23, pp. 47:1–47:33, 2021.