FlowDCN: Exploring DCN-like Architectures for Fast Image Generation with Arbitrary Resolution

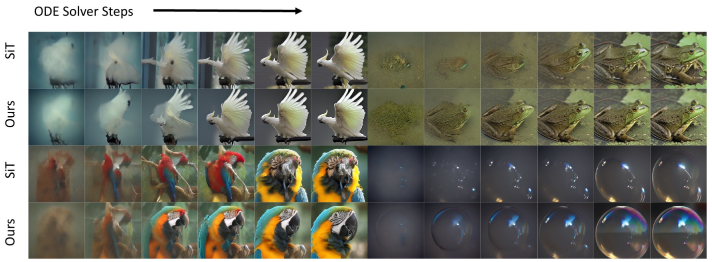
Shuai Wang, Zexian Li, Tianhui Song, Xubin Li, Tiezheng Ge, Bo Zheng, Limin Wang

Github

Arxiv

NEURAL INFORMATION PROCESSING SYSTEMS

Selected arbitrary-resolution samples (384x384, 224x448, 448x224, 256x256)



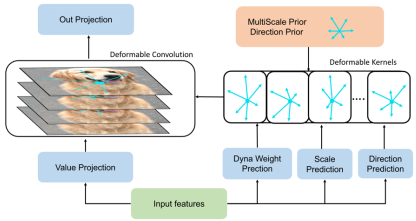
samples from our FlowDCN- XL/2 and SiT-XL/2 with Euler ODE solver under 2, 3, 4, 5, 8, 10 steps using the same noise

Motivations

Arbitrary-resolution image generation still remains a challenging task in AIGC, as it requires handling varying resolutions and aspect ratios while maintaining high visual quality. Existing transformer-based diffusion methods suffer from quadratic computation cost and limited resolution extrapolation capabilities, making them less effective for this task. In this paper, we propose FlowDCN, a purely convolution-based generative model with linear time and memory complexity, that can efficiently generate high-quality images at arbitrary resolutions. Equipped with a new design of learnable group-wise deformable convolution block, our FlowDCN yields higher flexibility and capability to handle different resolutions

- Explore Pure DCN-like arch for image generation
- DCN consumes Linear Complexity compared to Attention
- DCN can handle Arbitrary Resolution Generation

Method-GroupwiseMSDCN



- Different from DCNv3/DCNv4
- We Decouple deformable filed into Direction and Scales

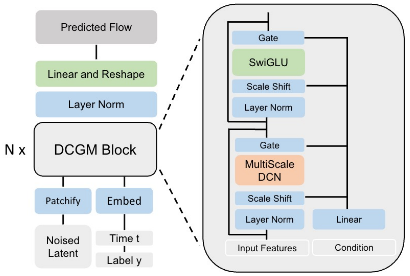
$$s^g(\mathbf{x}) = S_{\max} * \text{sigmoid}(\mathbf{W}_s^T \mathbf{x} + s_0^g),$$

$$p = p_0 + s^g(\mathbf{x}) * (p_k + \Delta p_k(\mathbf{x})).$$

- Initianlize s_0^g with following equation to achieve Groupswise MultiScale

$$s_0^{g+1} = \log\left(\frac{g}{G-g}\right).$$

Method-FlowDCN



The basic block:

$$\mathbf{x}_1 = \mathbf{x} + \text{AdaLN}(\mathbf{y}, \mathbf{t}, \text{MultiScale-DCN}(\mathbf{x})),$$
$$\mathbf{x}_2 = \mathbf{x}_1 + \text{AdaLN}(\mathbf{y}, \mathbf{t}, \ \text{SwiGLU}(\mathbf{x}_1)).$$

Training with rectified flow:

$$x_t = tx + (1-t)\epsilon.$$

$$v_t(x_t) = x - \epsilon.$$

$$\mathcal{L}_v = \int_0^1 \mathbb{E}[\| v_\theta(x_t, t) - v_t(x_t) \|^2] dt.$$

Method-Sampling Arbitrary Resolution

Smax Adjustment :

Adjust Smax with corresponding resolution

$$s_h^g(\mathbf{x}) = \text{sigmoid}(\mathbf{W}_s^T \mathbf{x} + s_0^g) \cdot S_{\max} \cdot \frac{H_{\text{test}}}{H_{\text{train}}},$$

$$s_w^g(\mathbf{x}) = \text{sigmoid}(\mathbf{W}_s^T \mathbf{x} + s_0^g) \cdot S_{\max} \cdot \frac{W_{\text{test}}}{W_{\text{train}}}.$$

Model	FLOPs (G)	Params (M)	Latency(ms)	FID↓	sFID↓	IS↑
SiT-S/2	6.06	33	0.026	57.64	9.05	24.78
SiT-S/2 †	6.06	33	0.026	57.9	**8.72**	24.64
FlowDCN-S/2	4.36 (-28%)	30.3 (-8.1%)	0.027	**54.6**	8.8	**26.4**
SiT-B/2	23.01	130	0.084	33.5	6.46	43.71
SiT-B/2 †	23.01	130	0.084	37.3	6.55	40.6
FlowDCN-B/2	17.87 (-22%)	120 (-7.6%)	0.076	**28.5**	**6.09**	**51**
w/o RMS & SwiGLU	17.88 (-22%)	120 (-7.6%)	0.072	29.1	6.13	50.4
DiT-L/2	80.71	458	0.291	23.3	-	-
SiT-L/2	80.71	458	0.291	18.8	5.29	72.02
FlowDCN-L/2	63.51 (-21%)	421 (-8.0%)	0.254	**13.8**	**4.69**	**85**
DiT-XL/2	118.64	675	0.387	19.5	-	-
SiT-XL/2	118.64	675	0.387	17.2	5.07	76.52
FlowDCN-XL/2	93.24 (-21%)	618 (-8.4%)	0.303	**11.3**	**4.85**	**97**

ImageNet 256×256 Benchmark

Generative Models	Long Residuals	Total Images(M)	Total GFLOPs	FID ↓	sFID ↓	IS ↑	P ↑	R ↑
ADM-U [10]	✓	507	3.76×10^{11}	7.49	5.13	127.49	0.72	0.63
CDM [39]	✓	-	-	4.88	-	158.71	-	-
LDM-4 [40]	✓	213	2.22×10^{10}	10.56	-	103.49	0.71	0.62
DiT-XL/2 [12]	✗	1792	2.13×10^{11}	9.62	6.85	121.50	0.67	**0.67**
DiffusionSSM-XL[16]	✗	660	1.85×10^{11}	9.07	5.52	118.32	0.69	0.64
SiT-XL/2[23]	✗	1792	2.13×10^{11}	8.61	6.32	**131.65**	0.68	**0.67**
FlowDCN-XL/2	✗	**384**	$\mathbf{3.57 \times 10^{10}}$	**8.36**	**5.39**	122.5	**0.69**	0.65
Classifier-free Guidance								
ADM-U[10]	✓	507	3.76×10^{12}	3.60	-	247.67	0.87	0.48
LDM-4 [40]	✓	213	2.22×10^{10}	3.95	-	178.22	0.81	0.55
U-ViT-H/2 [11]	✓	512	6.81×10^{10}	2.29	-	247.67	0.87	0.48
DiT-XL/2 [12]	✗	1792	2.13×10^{11}	2.27	4.60	**278.24**	0.83	0.57
DiffusionSSM-XL [16]	✗	660	1.85×10^{11}	2.28	4.49	259.13	**0.86**	0.56
SiT-XL/2[23]	✗	1792	2.13×10^{11}	2.06	4.50	270.27	0.82	**0.59**
FiT-XL/2[18]	✗	450	-	4.27	9.99	249.72	0.84	0.51
FlowDCN-XL/2 (cfg=1.375; ODE)	✗	**384**	$\mathbf{3.57 \times 10^{10}}$	2.13	**4.30**	243.46	0.81	0.57
FlowDCN-XL/2 (cfg=1.375; SDE)	✗	**384**	$\mathbf{3.57 \times 10^{10}}$	2.08	**4.38**	257.53	0.82	0.57
FlowDCN-XL/2 (cfg=1.375; ODE)	✗	**486**	$\mathbf{4.52 \times 10^{10}}$	**2.01**	**4.33**	254.36	0.81	0.58
FlowDCN-XL/2 (cfg=1.375; SDE)	✗	**486**	$\mathbf{4.52 \times 10^{10}}$	**2.00**	**4.37**	263.16	0.82	0.58

Class-Conditional ImageNet 512×512

Model	FID↓	sFID↓	IS↑	Precision↑	Recall↑
BigGAN-deep [6]	8.43	8.13	177.90	0.88	0.29
StyleGAN-XL [7]	2.41	4.06	267.75	0.77	0.52
ADM [10]	23.24	10.19	58.06	0.73	0.60
ADM-U [10]	9.96	5.62	121.78	0.75	0.64
ADM-G [10]	7.72	6.57	172.71	0.87	0.42
ADM-G, ADM-U	3.85	5.86	221.72	0.84	0.53
DiT-XL/2 [12]	12.03	7.12	105.25	0.75	0.64
DiT-XL/2-G [12] (cfg=1.50)	3.04	5.02	240.82	0.84	0.54
SiT-XL/2-G [23] (cfg=1.50)	2.62	4.18	252.21	0.84	0.57
FlowDCN-XL/2(cfg=1.375, ODE-50)	2.76	5.29	240.6	0.83	0.51
FlowDCN-XL/2(cfg=1.375, SDE-250)	**2.44**	**4.53**	**252.8**	0.84	0.54

Method	256×256 (1:1)			320×320 (1:1)			224×448 (1:2)			160×480 (1:3)		
	FID↓	**sFID↓**	**IS↑**	**FID↓**	**sFID↓**	**IS↑**	**FID↓**	**sFID↓**	**IS↑**	**FID↓**	**sFID↓**	**IS↑**
DiT-B	44.83	8.49	32.05	95.47	108.68	18.38	109.1	110.71	14.00	143.8	122.81	8.93
DiT-B + EI	44.83	8.49	32.05	81.48	62.25	20.97	133.2	72.53	11.11	160.4	93.91	7.30
DiT-B + PI	44.83	8.49	32.05	72.47	54.02	24.15	133.4	70.29	11.73	156.5	93.80	7.80
FiT-B	36.36	11.08	40.69	61.35	30.71	31.01	44.67	24.09	37.1	56.81	22.07	25.25
FiT-B + VisionYaRN	36.36	11.08	40.69	44.76	38.04	44.70	41.92	42.79	45.87	62.84	44.82	27.84
FiT-B + VisionNTK	36.36	11.08	40.69	57.31	31.31	33.97	43.84	26.25	39.22	56.76	24.18	26.40
FlowDCN-B	28.5	6.09	51	34.4	27.2	52.2	71.7	62.0	23.7	211	111	5.83
FlowDCN-B (+VAR)	23.6	7.72	62.8	29.1	15.8	69.5	31.4	17.0	62.4	44.7	17.8	35.8
+ $S_{\max}$ Adjust	23.6	7.72	62.8	30.7	19.4	68.5	37.8	22.8	54.4	53.3	22.6	31.5

FlowDCN

Presenter: WangShuai

- Motivations
- Methods
- Experiments
- Visualizations

FlowDCN: Exploring DCN-like Architectures for Fast Image Generation with Arbitrary Resolution

Shuai Wang Nanjing University
Zexian Li Alibaba Group
Tianhui Song Nanjing University
Xubin Li Alibaba Group
Tiezheng Ge Alibaba Group
Bo Zheng Alibaba Group
Limin Wang ✉ Nanjing University, Shanghai AI Lab

Figure 1: **Selected arbitrary-resolution samples (384x384, 224x448, 448x224, 256x256).** Generated from a single FlowDCN-XL/2 model trained on ImageNet 256×256 resolution with CFG = 4.0.

FlowDCN

- Direct Motivation
  - DCN-like arch models are much powerful than others (Generally)
  - DCN-like arch owns relatively higher dynamics compared to CNN
  - DCN-like arch enjoys relatively sparse pattern compared to Transformer

Table from DCNv3/DCNv4

Model	Size	Scale	Acc	Throughput
Swin-T	29M	224^2	81.3	1989 / 3619
ConvNeXt-T	29M	224^2	82.1	2485 / 4305
InternImage-T	30M	224^2	83.5	1409 / 1746
FlashInternImage-T	30M	224^2	**83.6**	2316 / 3154 (+64%/ + 80%)
Swin-S	50M	224^2	83.0	1167/2000
ConvNeXt-S	50M	224^2	83.1	1645/2538
InternImage-S	50M	224^2	84.2	1044/1321
FlashInternImage-S	50M	224^2	**84.4**	1625 / 2396
Swin-B	88M	224^2	83.5	934 / 1741
ConvNeXt-B	89M	224^2	83.8	1241 / 1888
InternImage-B	97M	224^2	**84.9**	779 / 1030
FlashInternImage-B	97M	224^2	**84.9**	1174 / 1816 (+51%/ + 76%)
Swin-L	197M	384^2	87.3	206 / 301
ConvNeXt-L	198M	384^2	87.5	252 / 436
InternImage-L	223M	384^2	87.7	158 / 214
ConvNeXt-XL	350M	384^2	87.8	170 / 299
InternImage-XL	335M	384^2	88.0	125 / 174
FlashInternImage-L	223M	384^2	**88.1**	248 / 401 (+57%/ + 87%)

Table 4. **Image classification performance on ImageNet-1K.** We show relative speedup between FlashInternImage w/ DCNv4 and its InternImage counterparts. DCNv4 significantly improves the speed while shows state-of-the-art performance.

Model	#param	FPS	Mask R-CNN 1× AP^b	Mask R-CNN 1× AP^m	Mask R-CNN 3×+MS AP^b	Mask R-CNN 3×+MS AP^m
Swin-T	48M	66 / 106	42.7	39.3	46.0	41.6
ConvNeXt-T	48M	78 / 113	44.2	40.1	46.2	41.7
InternImage-T	49M	54 / 69	47.2	42.5	49.1	43.7
FlashInternImage-T	49M	72 / 102	**48.0**	**43.1**	**49.5**	**44.0**
Swin-S	69M	45 / 77	44.8	40.9	48.2	43.2
ConvNeXt-S	70M	54 / 83	45.4	41.8	47.9	42.9
InternImage-S	69M	44 / 56	47.8	43.3	49.7	44.5
FlashInternImage-S	69M	57 / 83	**49.2**	**44.0**	**50.5**	**44.9**
Swin-B	107M	33 / 59	46.9	42.3	48.6	43.3
ConvNeXt-B	108M	43 / 70	47.0	42.7	48.5	43.5
InternImage-B	115M	33 / 43	48.8	44.0	50.3	44.8
FlashInternImage-B	115M	44 / 67	**50.1**	**44.5**	**50.6**	**45.4**

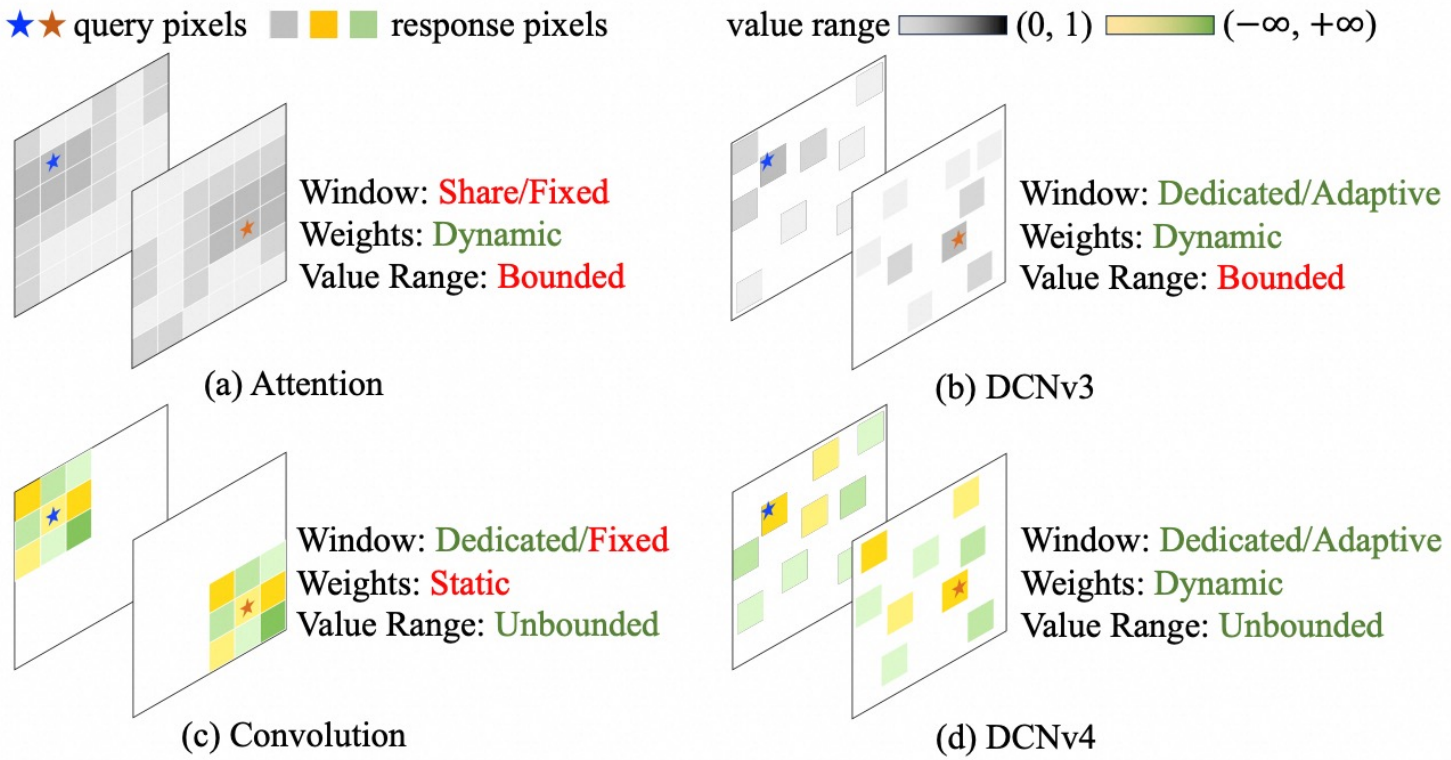
Table from MambaOut

Table 2: **Performance of object detection and instance segmentation on COCO with Mask R-CNN.** The MACs are measured with input size of 800 × 1280.

Backbone	Token Mixing Type	Param (M)	MAC (G)	Mask R-CNN 1× schedule AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
ConvNeXt-T [49]	Conv	48	262	44.2	66.6	48.3	40.1	63.3	42.8
FocalNet-T [89]	Conv	49	268	46.1	68.2	50.6	41.5	65.1	44.5
Swin-T [51]	Attn	48	267	42.7	65.2	46.8	39.3	62.2	42.2
ViT-Adapter-S [10]	Attn	48	403	44.7	65.8	48.3	39.9	62.5	42.8
CSWin-T [22]	Attn	42	279	46.7	68.6	51.3	42.2	65.6	45.4
PVTv2-B2 [80]	Conv + Attn	45	309	45.3	67.1	49.6	41.2	64.2	44.4
SG-Former-S [65]	Conv + Attn	41	–	47.4	69.0	52.0	42.6	65.9	46.0
TransNeXt-Tiny [69]	Conv + Attn	48	356	**49.9**	71.5	54.9	**44.6**	68.6	48.1
VMamba-T [50]	Conv + SSM	42	286	46.5	68.5	50.7	42.1	65.5	45.3
LocalVMamba-T [37]	Conv + SSM	45	291	46.7	68.7	50.8	42.2	65.7	45.5
EfficientVMamba-B [58]	Conv + SSM	53	252	43.7	66.2	47.9	40.2	63.3	42.9
VMambaV9-T [50]	Conv + SSM	50	270	**47.4**	69.5	52.0	**42.7**	66.3	46.0
PlainMamba-L1 [88]	Conv + SSM	31	388	44.1	64.8	47.9	39.1	61.6	41.9
MambaOut-Tiny	Conv	43	262	45.1	67.3	49.6	41.0	64.1	44.1

FlowDCN

- Revisit DCN module



- DCN prediction
  - Deformable field (offsets)
  - Dynamic weight (DCNv3/v4)
    - Softmax (DCNv3)
    - unbounded (DCNv4)

$$\Delta \mathbf{P}(\mathbf{x}) = \mathbf{W}_{\text{deformable}}^{T}\mathbf{x} + \mathbf{b}_{\text{deformable}},$$
$$\mathbf{W}(\mathbf{x}) = \mathbf{W}_{\text{weight}}^{T}\mathbf{x} + \mathbf{b}_{\text{weight}}.$$

$$\mathbf{y}^{g}(p_0) = \sum_{k=0}^{K} w_k^g \mathbf{x}^g(p_0 + p_k + \Delta p_k(\mathbf{x})),$$
$$\mathbf{y} = \text{concat}(\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^G).$$

FlowDCN

- Improve DCN module with MultiScale
  - Multiscale is pivotal for CV tasks
  - Deformable-DETR employs FPN (explict) as MultiScale feature
  - Retentive Net uses different gamms decay rates for different heads

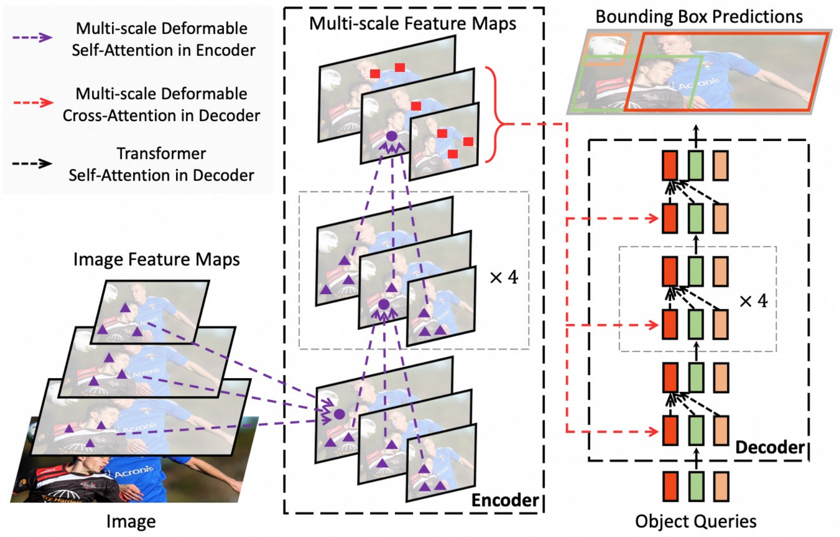


Figure 1: Illustration of the proposed Deformable DETR object detector.

2.2 Gated Multi-Scale Retention

We use $h = d_{\text{model}}/d$ retention heads in each layer, where d is the head dimension. The heads use different parameter matrices $W_Q, W_K, W_V \in \mathbb{R}^{d \times d}$. Moreover, **m**ulti-**s**cale **r**etention (MSR) assigns different γ for each head. For simplicity, we set γ identical among different layers and keep them fixed. In addition, we add a swish gate [HG16, RZL17] to increase the non-linearity of retention layers. Formally, given input X, we define the layer as:

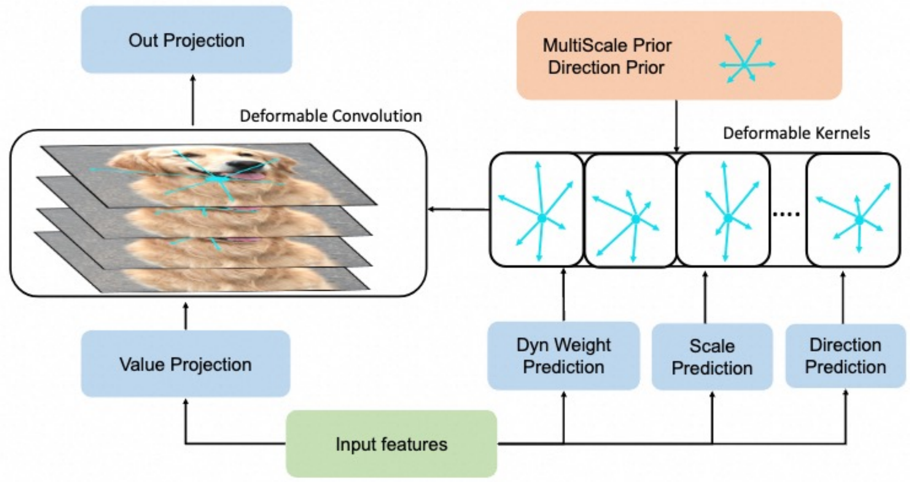
$$
\begin{aligned}
\gamma &= 1 - 2^{-5-\text{arange}(0,h)} \in \mathbb{R}^h \\
\text{head}_i &= \text{Retention}(X, \gamma_i) \\
Y &= \text{GroupNorm}_h(\text{Concat}(\text{head}_1, \cdots, \text{head}_h)) \\
\text{MSR}(X) &= (\text{swish}(XW_G) \odot Y)W_O
\end{aligned} \tag{8}
$$

where $W_G, W_O \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$ are learnable parameters, and GroupNorm [WH18] normalizes the output of each head, following SubLN proposed in [SPP+19]. Notice that the heads use multiple γ scales, which results in different variance statistics. So we normalize the head outputs separately.

The pseudocode of retention is summarized in Figure 4.

FlowDCN

- Groupwise MultiScale DCN module



(b) **MultiScale DCN Block.** Dynamic weight and scale& direction deformable field are predicted from input features, then merged with priors to form the deformable kernels to extract features.

- Decouple deformable field
  - Into Directions
  - Into Scales

$$s(\mathbf{x}) = S_{\max} * \mathrm{sigmoid}(\mathbf{W}_s^T \mathbf{x}),$$
$$p = p_0 + s(\mathbf{x}) * (p_k + \Delta p_k(\mathbf{x})),$$

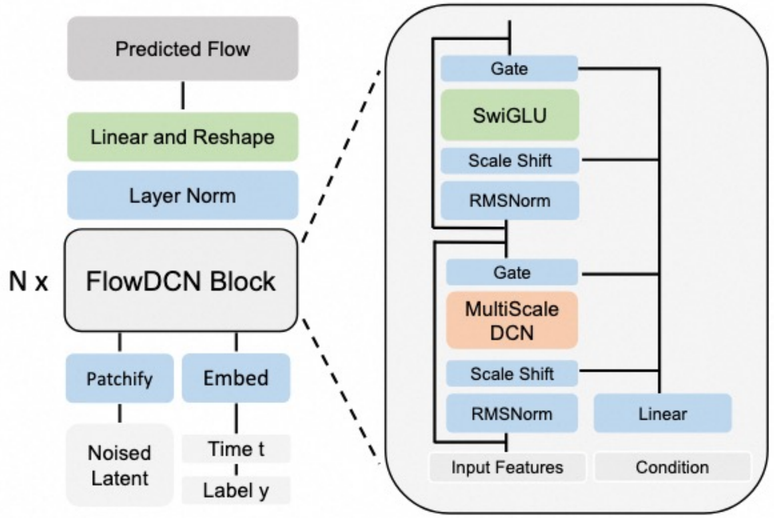
- Introducing scale priors groupwise

$$s_0^{g+1} = \log(\frac{g}{G-g}).$$

$$s^g(\mathbf{x}) = S_{\max} * \mathrm{sigmoid}(\mathbf{W}_s^T \mathbf{x} + s_0^g),$$
$$p = p_0 + s^g(\mathbf{x}) * (p_k + \Delta p_k(\mathbf{x})).$$

FlowDCN

- Experiments on ImageNet



(a) **FlowDCN Architecture.** Our FlowDCN consists of stacked MultiScaleDCN blocks and SwiGLU blocks. We also employ RMSNorm to stabilize training.

ImageNet 256×256 Benchmark								
Generative Models	Long Residuals	Total Images(M)	Total GFLOPs	FID ↓	sFID ↓	IS ↑	P ↑	R ↑
ADM-U [10]	✓	507	3.76×10^{11}	7.49	5.13	127.49	0.72	0.63
CDM [39]	✓	-	-	4.88	-	158.71	-	-
LDM-4 [40]	✓	213	2.22×10^{10}	10.56	-	103.49	0.71	0.62
DiT-XL/2 [12]	✗	1792	2.13×10^{11}	9.62	6.85	121.50	0.67	**0.67**
DiffusionSSM-XL[16]	✗	660	1.85×10^{11}	9.07	5.52	118.32	0.69	0.64
SiT-XL/2[23]	✗	1792	2.13×10^{11}	8.61	6.32	**131.65**	0.68	**0.67**
FlowDCN-XL/2	✗	**384**	$\mathbf{3.57 \times 10^{10}}$	**8.36**	**5.39**	122.5	**0.69**	0.65
Classifier-free Guidance								
ADM-U[10]	✓	507	3.76×10^{12}	3.60	-	247.67	0.87	0.48
LDM-4 [40]	✓	213	2.22×10^{10}	3.95	-	178.22	0.81	0.55
U-ViT-H/2 [11]	✓	512	6.81×10^{10}	2.29	-	247.67	0.87	0.48
DiT-XL/2 [12]	✗	1792	2.13×10^{11}	2.27	4.60	**278.24**	0.83	0.57
DiffusionSSM-XL [16]	✗	660	1.85×10^{11}	2.28	4.49	259.13	**0.86**	0.56
SiT-XL/2[23]	✗	1792	2.13×10^{11}	2.06	4.50	270.27	0.82	**0.59**
FiT-XL/2[18]	✗	450	-	4.27	9.99	249.72	0.84	0.51
FlowDCN-XL/2 (cfg=1.375; ODE)	✗	**384**	$\mathbf{3.57 \times 10^{10}}$	2.13	**4.30**	243.46	0.81	0.57
FlowDCN-XL/2 (cfg=1.375; SDE)	✗	**384**	$\mathbf{3.57 \times 10^{10}}$	2.08	**4.38**	257.53	0.82	0.57
FlowDCN-XL/2 (cfg=1.375; ODE)	✗	**486**	$\mathbf{4.52 \times 10^{10}}$	**2.01**	**4.33**	254.36	0.81	0.58
FlowDCN-XL/2 (cfg=1.375; SDE)	✗	**486**	$\mathbf{4.52 \times 10^{10}}$	**2.00**	**4.37**	263.16	0.82	0.58

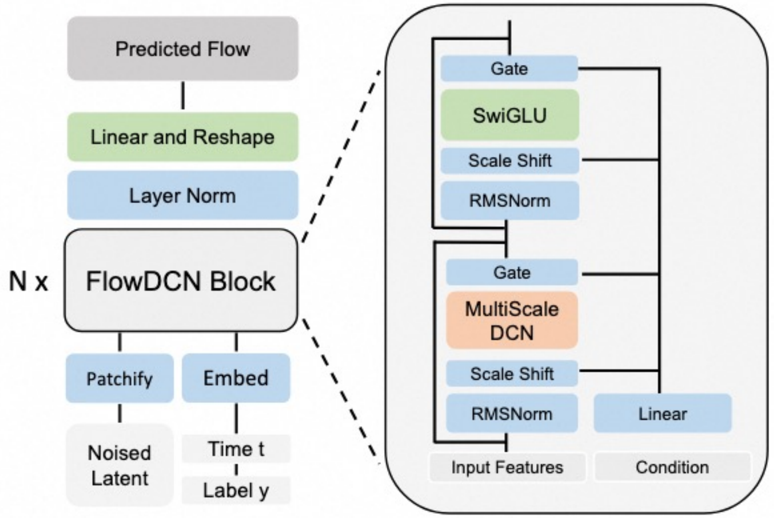
Table 4: **Image generation quality evaluation of and existing approaches on ImageNet 256× 256.** Total images by training steps × batch size as reported, and total GFLOPs by Total Images × GFLOPs/Image. P refers to Precision and R refers to Recall.

Class-Conditional ImageNet 512×512					
Model	FID↓	sFID↓	IS↑	Precision↑	Recall↑
BigGAN-deep [6]	8.43	8.13	177.90	0.88	0.29
StyleGAN-XL [7]	2.41	4.06	267.75	0.77	0.52
ADM [10]	23.24	10.19	58.06	0.73	0.60
ADM-U [10]	9.96	5.62	121.78	0.75	0.64
ADM-G [10]	7.72	6.57	172.71	0.87	0.42
ADM-G, ADM-U	3.85	5.86	221.72	0.84	0.53
DiT-XL/2 [12]	12.03	7.12	105.25	0.75	0.64
DiT-XL/2-G [12] (cfg=1.50)	3.04	5.02	240.82	0.84	0.54
FlowDCN-XL/2(cfg=1.375, ODE-50)	**2.76**	5.29	240.6	0.83	0.51
FlowDCN-XL/2(cfg=1.375, SDE-250)	**2.44**	**4.53**	**252.8**	0.84	0.54

Table 5: **Benchmarking class-conditional image generation on ImageNet 512×512.** Our FlowDCN-XL/2 is fine-tuned for 100k steps from the same model trained on 256 × 256 resolution setting of 1.5M steps

FlowDCN

- Experiments on ImageNet



(a) **FlowDCN Architecture.** Our FlowDCN consists of stacked MultiScaleDCN blocks and SwiGLU blocks. We also employ RMSNorm to stabilize training.

Model	FLOPs (G)	Params (M)	Latency(ms)	FID↓	sFID↓	IS↑
SiT-S/2	6.06	33	0.026	57.64	9.05	24.78
SiT-S/2 †	6.06	33	0.026	57.9	**8.72**	24.64
FlowDCN-S/2	4.36 (-28%)	30.3 (-8.1%)	0.027	**54.6**	8.8	**26.4**
SiT-B/2	23.01	130	0.084	33.5	6.46	43.71
SiT-B/2 †	23.01	130	0.084	37.3	6.55	40.6
FlowDCN-B/2	17.87 (-22%)	120 (-7.6%)	0.076	**28.5**	**6.09**	**51**
w/o RMS & SwiGLU	17.88 (-22%)	120 (-7.6%)	0.072	29.1	6.13	50.4
DiT-L/2	80.71	458	0.291	23.3	-	-
SiT-L/2	80.71	458	0.291	18.8	5.29	72.02
FlowDCN-L/2	63.51 (-21%)	421 (-8.0%)	0.254	**13.8**	**4.69**	**85**
DiT-XL/2	118.64	675	0.387	19.5	-	-
SiT-XL/2	118.64	675	0.387	17.2	5.07	76.52
FlowDCN-XL/2	93.24 (-21%)	618 (-8.4%)	0.303	**11.3**	**4.85**	**97**

Table 3: **Image generation metrics comparisons between SiT [23], DiT [12] under 400k training steps budgets**. All metrics are calculated from the sampled 50k images under 250 Euler SDE sampling steps without classifier-free guidance. †: reproduced result. Latency(ms) is the 1-NFE latency and collected from Nvidia A10 GPU with 16 batchsize under float32.

FlowDCN

- Visualizations on ImageNet

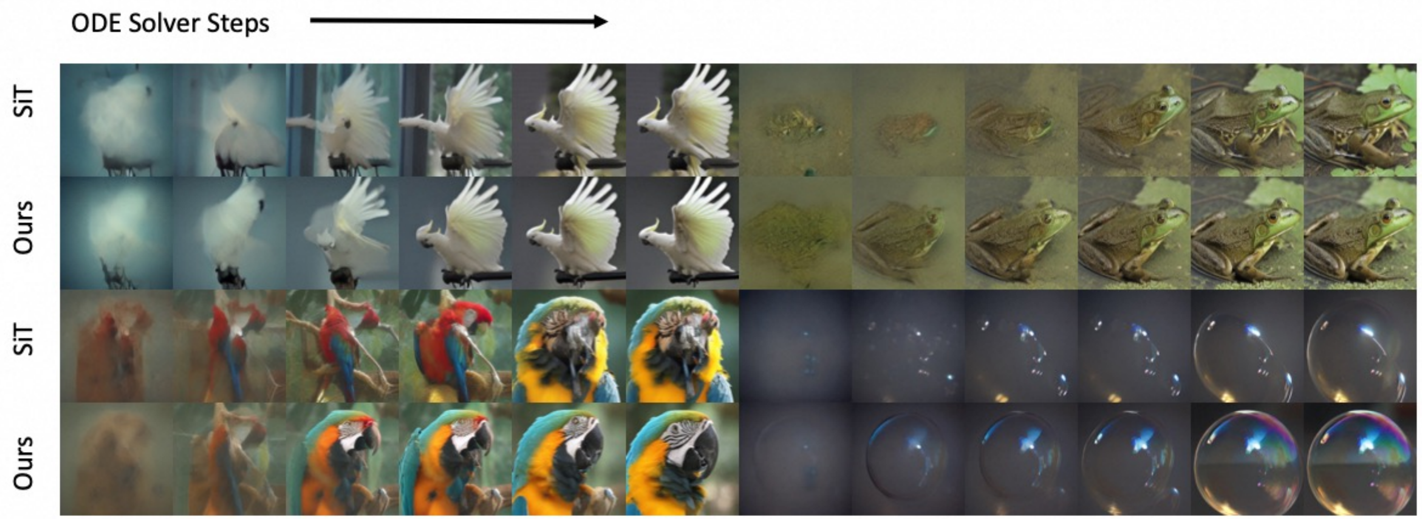
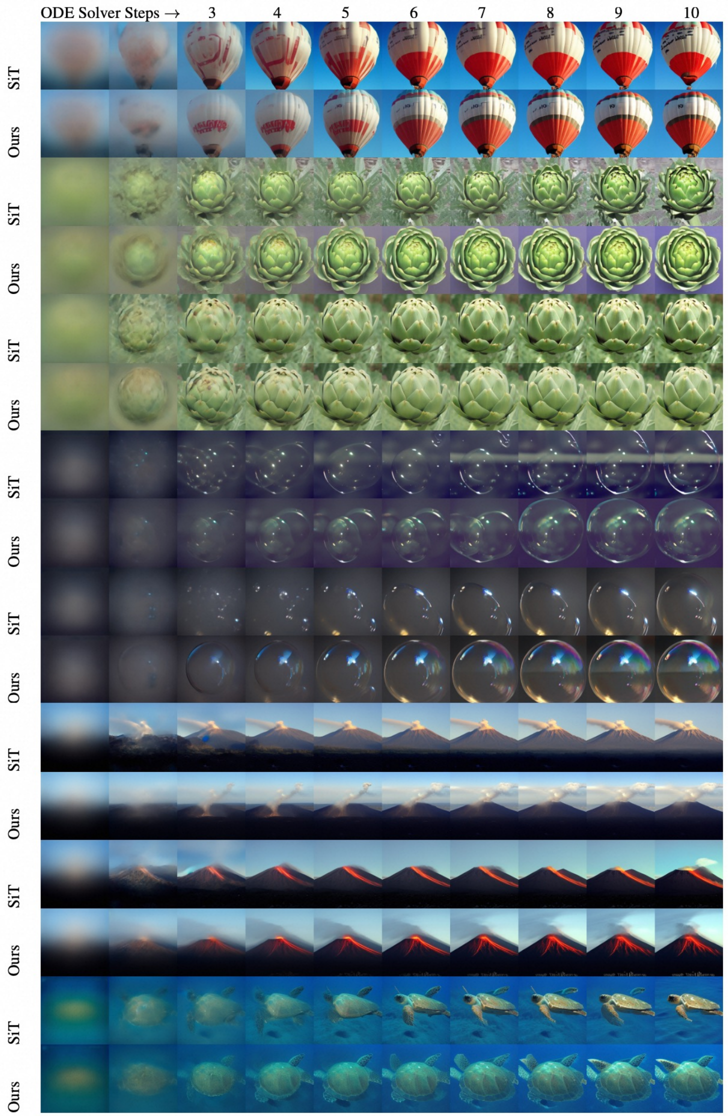


Figure 3: **Visualization Comparison with SiT**. *Best viewed zoomed-in*. We sample both our FlowDCN-XL/2 and SiT-XL/2 with Euler ODE solver under 2, 3, 4, 5, 8, 10 steps using the same latent noise. At the fewer steps sampling scenery, our FlowDCN generates slightly clearer and higher-quality images.



FlowDCN

- Resolution Extension

$$s^g(\mathbf{x}) = S_{\max} * \text{sigmoid}(\mathbf{W}_s^T \mathbf{x} + s_0^g),$$
$$p = p_0 + s^g(\mathbf{x}) * (p_k + \Delta p_k(\mathbf{x})).$$

Adjust $S_{\max}$ to match inference resolution. As shown in Eq. (10), $S_{\max}$ controls the maximum sampling range in multiscale deformable convolution. As discussed in Sec. 3.1, we treat it as a resolution-dependent hyperparameter. It is straightforward to observe that scaling $S_{\max}$ with the relative aspect ratio between train size and inference size could match the reception field between train and inference:

$$s_h^g(\mathbf{x}) = \text{sigmoid}(\mathbf{W}_s^T \mathbf{x} + s_0^g) \cdot S_{\max} \cdot \frac{H_{\text{test}}}{H_{\text{train}}} \tag{15}$$
$$s_w^g(\mathbf{x}) = \text{sigmoid}(\mathbf{W}_s^T \mathbf{x} + s_0^g) \cdot S_{\max} \cdot \frac{W_{\text{test}}}{W_{\text{train}}} \tag{16}$$

with $S_{\max}$ Adjustment　　　　without $S_{\max}$ Adjustment

Figure 4: **Visualization Comparison about $S_{\max}$ Adjustment.** Here are the 512×512, 256×512 and 512×256, three type resolution images. We employ the same latent noise as start, sampling with Euler SDE solver for 250 steps. With $S_{\max}$ Adjustment, sampled images consistently looks better.

Method	256×256 (1:1)			320×320 (1:1)			224×448 (1:2)			160×480 (1:3)		
	FID↓	sFID↓	IS↑	FID↓	sFID↓	IS↑	FID↓	sFID↓	IS↑	FID↓	sFID↓	IS↑
DiT-B	44.83	8.49	32.05	95.47	108.68	18.38	109.1	110.71	14.00	143.8	122.81	8.93
DiT-B + EI	44.83	8.49	32.05	81.48	62.25	20.97	133.2	72.53	11.11	160.4	93.91	7.30
DiT-B + PI	44.83	8.49	32.05	72.47	54.02	24.15	133.4	70.29	11.73	156.5	93.80	7.80
FiT-B	36.36	11.08	40.69	61.35	30.71	31.01	44.67	24.09	37.1	56.81	22.07	25.25
FiT-B + VisionYaRN	36.36	11.08	40.69	44.76	38.04	44.70	41.92	42.79	45.87	62.84	44.82	27.84
FiT-B + VisionNTK	36.36	11.08	40.69	57.31	31.31	33.97	43.84	26.25	39.22	56.76	24.18	26.40
FlowDCN-B	28.5	6.09	51	34.4	27.2	52.2	71.7	62.0	23.7	211	111	5.83
FlowDCN-B (+VAR)	23.6	7.72	62.8	29.1	15.8	69.5	31.4	17.0	62.4	44.7	17.8	35.8
+ $S_{\max}$ Adjust	23.6	7.72	62.8	30.7	19.4	68.5	37.8	22.8	54.4	53.3	22.6	31.5

Table 9: **Benchmarking resolution extrapolations on ImageNet with various aspect ratio training**. VAR indicates various aspect ratios training. We follow the same evaluation pipeline of FiT without using CFG.

FlowDCN

- Resolution Extension

$$s^g(\mathbf{x}) = S_{\max} * \mathrm{sigmoid}(\mathbf{W}_s^T \mathbf{x} + s_0^g),$$

$$p = p_0 + s^g(\mathbf{x}) * (p_k + \Delta p_k(\mathbf{x})).$$

Adjust $S_{\max}$ to match inference resolution. As shown in Eq. (10), $S_{\max}$ controls the maximum sampling range in multiscale deformable convolution. As discussed in Sec. 3.1, we treat it as a resolution-dependent hyperparameter. It is straightforward to observe that scaling $S_{\max}$ with the relative aspect ratio between train size and inference size could match the reception field between train and inference:

$$s_h^g(\mathbf{x}) = \mathrm{sigmoid}(\mathbf{W}_s^T \mathbf{x} + s_0^g) \cdot S_{\max} \cdot \frac{H_{\text{test}}}{H_{\text{train}}} \tag{15}$$

$$s_w^g(\mathbf{x}) = \mathrm{sigmoid}(\mathbf{W}_s^T \mathbf{x} + s_0^g) \cdot S_{\max} \cdot \frac{W_{\text{test}}}{W_{\text{train}}} \tag{16}$$

Figure 5: Wide images examples of Class ID (972 , 980, 437)