DPAR: Dynamic Patchification for Efficient Autoregressive Visual Generation

Supplementary Material

1. Training

Algorithm 1 DPAR Patchification Algorithm

Input: Image tokens $I_{\text{tok}} = [x_0, x_1, \ldots, x_{T-1}]$, entropy model $\mathcal{E}_\phi$, threshold $\mathcal{H}_{\text{Th}}$, max patch length $P_{\max}$
Output: Patch sequence I_{patch}

$I_{\text{patch}} \leftarrow [[x_0]]$ ▷ Initialize first patch with first token
$P \leftarrow [x_1]$ ▷ Current patch
for $i = 2$ to $T-1$ **do**
  $e_i \leftarrow \mathcal{H}(x_{<i}; \mathcal{E}_\phi)$
  if $e_i \leq \mathcal{H}_{\text{Th}}$ **and** $|P| < P_{\max}$ **and** x_i not at row start **then**
    $P \leftarrow P \cup [x_i]$ ▷ Add token to current patch
  else
    $I_{\text{patch}} \leftarrow I_{\text{patch}} \cup [P]$ ▷ Finalize current patch
    $P \leftarrow [x_i]$ ▷ Start new patch
  end if
end for
$I_{\text{patch}} \leftarrow I_{\text{patch}} \cup [P]$ ▷ Add last patch
return I_{patch}

1.1. Detailed Results

We have provided detailed results for all model variants across different training epochs and CFG scales in Table S7 and Table S8. We also provide resolution-scaling CFG sweep results across 128, 256, 384, and 512 resolutions for DPAR-B and LlamaGen-B in Table S9. Finally, we have provided a training cost comparison with LlamaGen, which was used for generating FID-FLOPs chart in Table S1.

1.2. Patchification Algorithm

The patchification algorithm assigns each token in the 1D sequence $I_{\text{tok}} = [x_0, \ldots, x_{T-1}]$ to a patch index such that all token indices within a patch remain contiguous. Formally, we construct a patch sequence $I_{\text{patch}} = [P_0, \ldots, P_{M-1}]$, where each patch $P_m = [s_m, \ldots, f_m]$ is a contiguous non-overlapping span of token indices starting at s_m and ending at f_m in the original token sequence. The number of patches M varies per image and is typically smaller than the total token count T.

1.3. Ablation Study: Encoder-Decoder Layers

We conduct an ablation study to analyze the impact of varying the number of encoder and decoder layers on model performance, keeping the total number of encoder and decoder layers constant. As shown in Table S2, configurations with shallower encoders and deeper decoders (E1D4) yield the best FID scores. This suggests that allocating more capacity to the decoder is beneficial for generating high-quality images, while a lighter encoder suffices for aggregating tokens into patches.

Variant	LlamaGen (GFLOPs)	DPAR (GFLOPs)	Reduction (%)
B-256	24.98	19.21	23.1
L-256	83.26	56.74	31.9
XL-256	192.69	125.52	34.9
B-384	56.21	40.92	27.2
L-384	187.35	117.92	37.1
XL-384	433.57	258.53	40.4
XXL-384	797.59	459.63	42.4

Table S1. **Compute comparison across all LlamaGen and DPAR variants**. DPAR consistently reduces FLOPs across both 256×256 and 384×384 model families.

Layers (E#D#)	E1D4	E2D3	E3D2	E4D1
FID-50K (↓)	**3.32**	3.35	3.51	3.85

Table S2. **Ablation on encoder–decoder depth.** E_iD_j indicates i encoder layers and j decoder layers. Shallow encoders with deeper decoders (E1D4) provide the best FID.

1.4. Dynamic RoPE

2D Rotary Positional Embedding (RoPE) [1] encodes each token's 2D spatial coordinate (x, y) by rotating its query and key representations in latent space with dimensionality d using sinusoidal functions of the coordinates. For a token located at 2D coordinates (x, y) in the image, the positional encoding is given by:

$$\begin{aligned} \omega_i &= 10000^{-4(i-1)/d}, \quad i = 1, \ldots, \tfrac{d}{4}, \\ \mathbf{r}_x &= [\sin(\omega_i x), \cos(\omega_i x)]_{i=1}^{d/4}, \\ \mathbf{r}_y &= [\sin(\omega_i y), \cos(\omega_i y)]_{i=1}^{d/4}, \\ \mathbf{r}_{(x,y)} &= [\mathbf{r}_x, \mathbf{r}_y] \end{aligned} \tag{S1}$$

where ω_i are the frequency terms, and the resulting $\mathbf{r}_{(x,y)}$ is used to rotate the query and key vectors to encode 2D spatial relationships. We propose Dynamic RoPE, an extension of 2D RoPE to handle variable length patches by encoding the start and end coordinates of each patch along the y-axis. This is possible since each row starts a new patch. The updated positional encoding for a patch P_m that spans tokens

Method	FID↓
2D Embedding	3.32
Dynamic Embedding w/o redundancy	3.42
Dynamic Embedding	**3.31**

Table S3. **Comparison of positional embedding schemes.** Dynamic Embedding achieves the best FID on ImageNet 256×256.

from (x, y_{s_m}) to (x, y_{f_m}) is defined as:

$$\begin{aligned}
\omega_i &= 10000^{-4(i-1)/d}, \quad i = 1, \ldots, \tfrac{d}{4}, \\
\alpha_i &= 10000^{-16(i-1)/d}, \quad i = 1, \ldots, \tfrac{d}{16}, \\
\mathbf{r}_x &= [\sin(\omega_i x), \cos(\omega_i x)]_{i=1}^{d/4}, \\
\mathbf{r}_{y_s} &= [\sin(\alpha_i y_{s_m}), \cos(\alpha_i y_{s_m})]_{i=1}^{d/16}, \\
\mathbf{r}_{y_f} &= [\sin(\alpha_i y_{f_m}), \cos(\alpha_i y_{f_m})]_{i=1}^{d/16}, \\
\mathbf{r}_{(x,y_s,y_f)} &= [\mathbf{r}_x, \mathbf{r}_{y_s}, \mathbf{r}_{y_f}, \mathbf{r}_{y_f}, \mathbf{r}_{y_s}]
\end{aligned} \tag{S2}$$

Our idea is to encode both the starting and ending y-coordinates of each patch, allowing the model to capture the horizontal span of each patch in addition to its vertical position. Further, adding redundancy by repeating the start and end positional encodings leads to better representation as observed in Table S3.

1.5. Entropy-Model Dataset Transferability

We evaluate whether an entropy model trained on a different dataset can still provide useful patchification for ImageNet generation. As shown in Table S4, using an entropy model trained on COCO-17 yields similar FID to an ImageNet-trained entropy model while maintaining a strong correlation between the resulting entropy maps. This suggests that entropy models trained on reasonably diverse dataset can be generalized for entropy prediction on a different dataset.

Entropy Model Dataset	DPAR-B FID	p-corr
ImageNet	3.98	1.00
COCO-17 (118K)	3.96	0.85

Table S4. **Entropy-model dataset transferability.** DPAR-B is trained on ImageNet while entropy maps are generated by entropy models trained on different datasets.

1.6. Entropy-Model Size Sensitivity

A lightweight EM is sufficient for patchification, with only marginal gains from increasing its size. Table S5 shows a lightweight 50M entropy model performs comparably to a 110M model, indicating that patchification quality is not highly sensitive to entropy-model scale and a small model is sufficient for high-quality patchification.

Entropy Model Size	DPAR-B FID	p-corr
110M	3.976	1.00
50M	3.982	0.88

Table S5. **Effect of entropy-model size.** A lightweight entropy model yields comparable patchification quality and generation performance.

1.7. Scaling to 1.4B Parameters

We also scale DPAR to the XXL regime at 384 resolution to test whether the training-efficiency gains persist at higher capacity. As shown in Table S6, DPAR-384-XXL improves FID over the LlamaGen baseline while reducing training FLOPs by 42.37%.

Model	FID	Train FLOPs
LlamaGen-384-XXL (1.4B)	2.34	797.59
DPAR-384-XXL (1.4B)	2.30	459.63 (-42.4%)

Table S6. **Scaling to 1.4B parameters at 384 resolution** from rebuttal experiments.

1.8. Relation with Image Compression Algorithms

For such images, popular compression algorithms such as PNG, which compress based on information content, also tend to have a larger size after compression. We observed a high correlation (p-corr = 0.54) between the image size after compression (indicative of information content) versus number of patches in an image, suggesting that NTPE aligns well with information content of an image.

Model	Params	Epoch	CFG	FID↓	IS↑	Prec.↑	Rec.↑
B-256	120M	300	1.75	5.02	193.99	0.78	0.54
			1.90	4.28	219.40	0.81	0.52
			2.00	4.07	235.39	0.82	0.50
			2.10	3.98	250.62	0.83	0.49
L-256	352M	300	1.75	3.24	241.05	0.79	0.58
			1.90	2.93	269.34	0.81	0.56
			2.00	2.96	284.06	0.82	0.54
			2.10	3.03	298.19	0.83	0.54
XL-256	789M	200	2.00	2.86	277.37	0.82	0.56
		300	1.75	2.82	249.57	0.79	0.60
			1.90	2.69	270.30	0.81	0.57
			2.00	2.67	281.65	0.82	0.56
			2.10	2.73	292.07	0.82	0.56

Table S7. **Comparison of models, parameters, epochs, and CFG values.** for model trained on resolution 256×256

Model	Params	Epoch	CFG	FID↓	IS↑	Prec.↑	Rec.↑
B-384	120M	50	2.00	5.96	190.16	0.79	0.47
		100	2.00	5.22	213.44	0.82	0.46
		200	2.00	4.74	230.43	0.81	0.48
		300	1.75	5.46	196.58	0.78	0.52
			1.90	4.62	223.31	0.81	0.50
			2.00	4.41	237.38	0.82	0.48
			2.10	4.29	254.54	0.83	0.47
L-384	352M	50	2.00	3.43	285.02	0.83	0.52
		100	2.00	3.10	290.11	0.82	0.53
		200	2.00	3.10	298.13	0.82	0.53
		300	1.75	3.00	256.07	0.79	0.58
			1.90	2.79	283.84	0.81	0.55
			2.00	2.84	299.32	0.82	0.55
			2.10	2.93	315.02	0.83	0.54
XL-384	789M	50	2.00	2.98	289.90	0.81	0.55
		100	2.00	2.77	307.30	0.82	0.56
		200	2.00	2.58	308.11	0.83	0.55
		300	1.75	2.81	261.09	0.79	0.59
			1.90	2.60	285.43	0.81	0.57
			2.00	2.62	299.31	0.82	0.57
			2.10	2.68	314.75	0.82	0.56

Table S8. **Comparison of models, parameters, epochs, and CFG values.** for model trained on resolution 384×384

Model	Epoch	Plen	Threshold	CFG	FID↓	IS↑	Prec.↑	Rec.↑
DPAR-B-128	299	7.60	4.00	1.75	21.41	74.52	0.53	0.59
				1.90	18.48	88.29	0.56	0.57
				2.00	16.88	97.23	0.57	0.57
				2.10	15.55	106.41	0.58	0.55
LlamaGen-B-128	299	NA	NA	1.75	24.53	63.50	0.51	0.53
				1.90	22.46	72.26	0.53	0.49
				2.00	21.56	78.81	0.54	0.49
				2.10	20.75	84.82	0.56	0.47
DPAR-B-256	299	7.80	4.00	1.75	5.02	193.99	0.78	0.54
				1.90	4.28	219.40	0.81	0.52
				2.00	4.07	235.39	0.82	0.50
				2.10	3.98	250.62	0.83	0.49
LlamaGen-B-256	299	NA	NA	2.00	5.46	193.61	7.50	0.84
DPAR-B-384	299	7.90	4.00	1.75	5.46	196.58	0.78	0.52
				1.90	4.62	223.31	0.81	0.50
				2.00	4.41	237.38	0.82	0.48
				2.10	4.29	254.54	0.83	0.47
LlamaGen-B-384	299	NA	NA	2.25	6.09	182.54	7.24	0.84
DPAR-B-512	299	8.10	4.00	1.75	8.20	165.76	0.74	0.52
				1.90	6.69	190.46	0.76	0.51
				2.00	6.05	205.51	0.78	0.49
				2.10	5.59	222.18	0.79	0.48
LlamaGen-B-512	299	NA	NA	1.75	10.61	110.63	0.76	0.48
				1.90	9.20	124.23	0.78	0.46
				2.00	8.44	133.39	0.80	0.44
				2.10	7.97	142.48	0.81	0.42

Table S9. **Resolution-scaling complete results for DPAR-B and LlamaGen-B.** Results at 128, 256, 384, and 512 resolutions across sampled CFG values.

Figure S1. Uncurated generated samples for model **DPAR-XL** trained at 256×256 resolution at **CFG-scale=1.75**

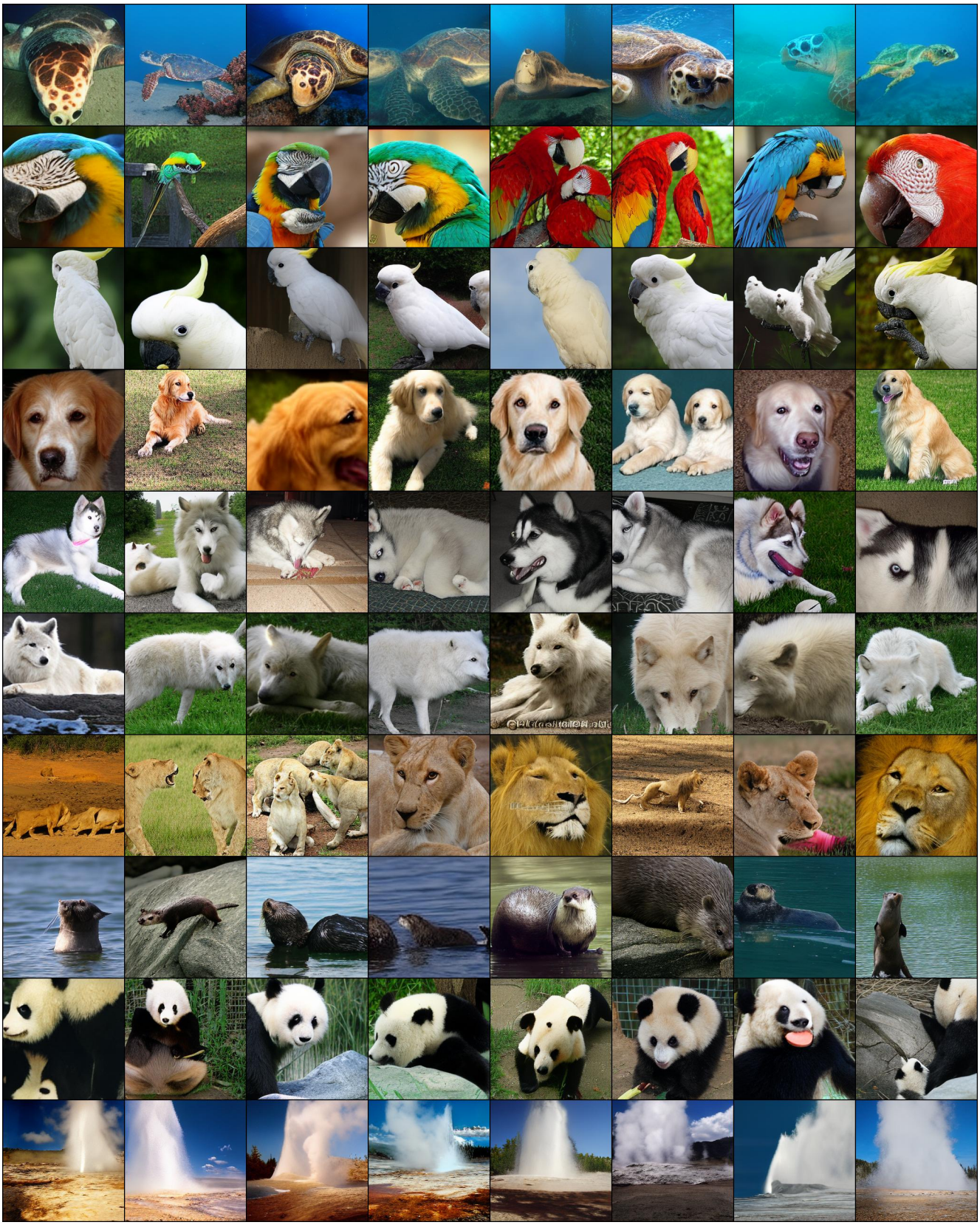

Figure S2. Uncurated generated samples for model **DPAR-XL** trained at 256×256 resolution at **CFG-scale=1.9**

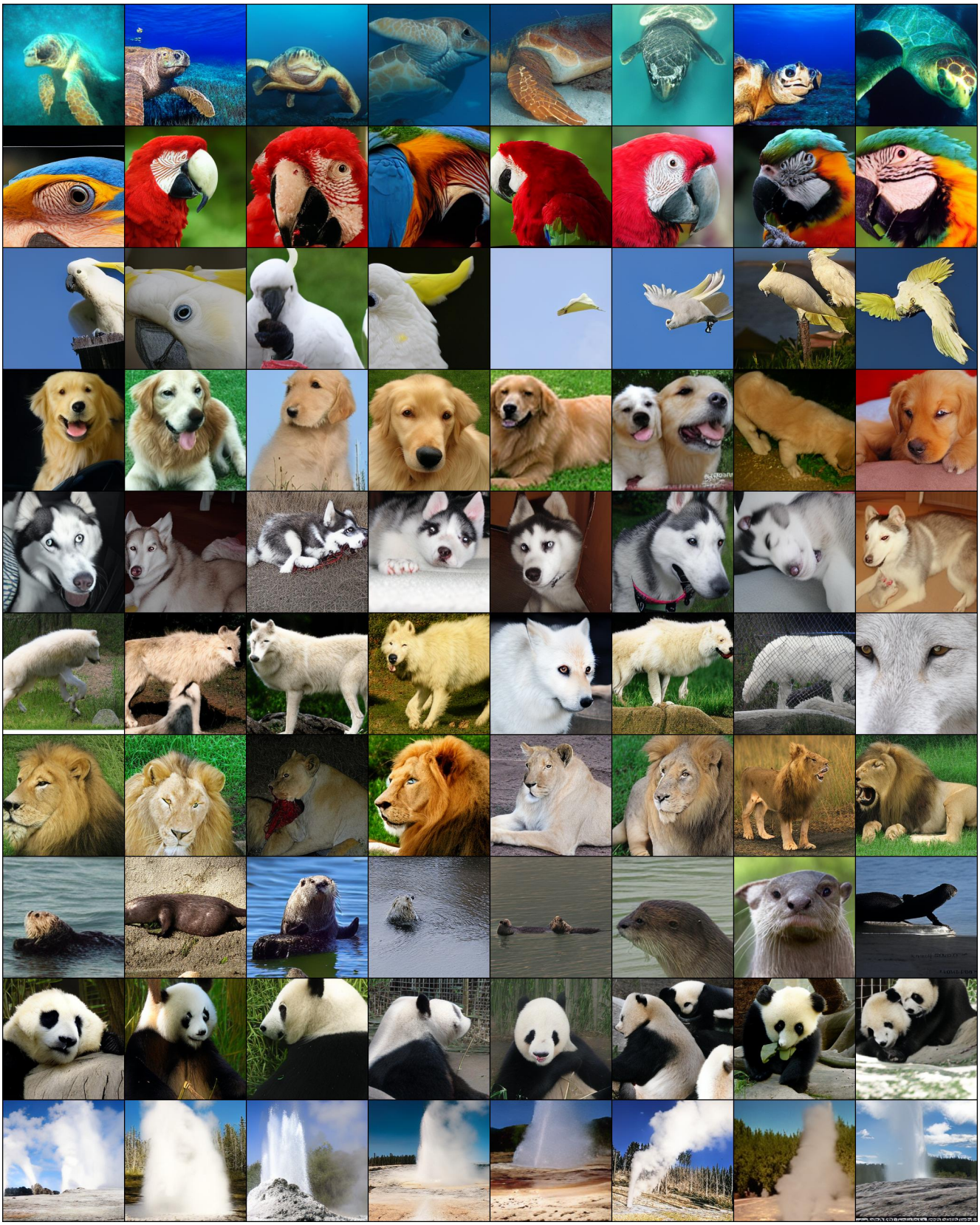

Figure S3. Uncurated generated samples for model **DPAR-XL** trained at 256×256 resolution at **CFG-scale=2.0**

Figure S4. Uncurated generated samples for model **DPAR-XL** trained at 256×256 resolution at **CFG-scale=2.1**

Figure S5. Uncurated generated samples for model **DPAR-XL** trained at 384×384 resolution at **CFG-scale=1.75**

Figure S6. Uncurated generated samples for model **DPAR-XL** trained at 384×384 resolution at **CFG-scale=1.9**

Figure S7. Uncurated generated samples for model **DPAR-XL** trained at 384×384 resolution at **CFG-scale=2.0**

Figure S8. Uncurated generated samples for model **DPAR-XL** trained at 384×384 resolution at **CFG-scale=2.1**

References

[1] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. 1