A Architecture and Hyperparameters

We proposed NUWA-Infinity as a two-stage approach. In the first stage, we encode raw visual pixels $1024 \times 1024$ into discrete tokens $64 \times 64$ by VQGAN with a compression ratio of 16. In second stage, we train a rendering model based on transformer architecture.

<table>
<thead>
<tr>
<th>Setting</th>
<th>NUWA-Infinity (Text-to-Image)</th>
<th>NUWA-Infinity-V (Image-to-Video)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VQGAN codebook</td>
<td>16384</td>
<td>16384</td>
</tr>
<tr>
<td>VQGAN dimension</td>
<td>256</td>
<td>256</td>
</tr>
<tr>
<td>VQGAN compression ratio</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>Transformer layer number</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>Transformer hidden dimen</td>
<td>1280</td>
<td>1280</td>
</tr>
<tr>
<td>Transformer head number</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>Transformer self-attention</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Transformer cross-attention</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Patch size</td>
<td>256</td>
<td>256</td>
</tr>
<tr>
<td>Nearby expansion size</td>
<td>(2, 2, 0)</td>
<td>(1, 1, 3)</td>
</tr>
<tr>
<td>Dataset</td>
<td>LHQC</td>
<td>LHQ-V</td>
</tr>
<tr>
<td>Training number</td>
<td>85K</td>
<td>38K</td>
</tr>
<tr>
<td>Test number</td>
<td>5K</td>
<td>2K</td>
</tr>
<tr>
<td>Training epoch</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Visual input size</td>
<td>$1024 \times 1024 \times 1$</td>
<td>$1024 \times 1024 \times 5$</td>
</tr>
<tr>
<td>Text input size</td>
<td>77</td>
<td>N/A</td>
</tr>
<tr>
<td>Batch size</td>
<td>256</td>
<td>256</td>
</tr>
<tr>
<td>Learning rate</td>
<td>1e-4</td>
<td>1e-4</td>
</tr>
<tr>
<td>Warmup ratio</td>
<td>5%</td>
<td>5%</td>
</tr>
</tbody>
</table>

Table 1: Implementation details for the large model

As shown in Tab.1 NUWA-Infinity by default refers to the model of text-to-image synthesis. Texts will be encoded into the tensors of $77 \times 512$ size by pretrained text encoder of CLIP, and they are fed into cross-attention as key and value to interact with visual features. We use three data augmentations on the images including RandomResizedCrop, RandomHorizontalFlip and ColorJitter. The image is only spatially expanded, so we choose $(e^h, e^w, e^f) = (2, 2, 0)$ expansion size for image synthesis. NUWA-Infinity-V refers to the model of image-to-video synthesis. Since it mainly expands the videos in the temporal axis, we choose a smaller spatial receptive field $(e^h, e^w) = (1, 1)$ but a larger temporal receptive field $e^f = 3$. Each $1024 \times 1024 \times 5$ clip is cropped from video with 5fps, and they also apply the three data augmentations mentioned above.

For training, we employ an Adam optimizer for 50 epochs using 5% of linear warm-up to a peak learning rate of 1e-4 and a linear decay learning rate scheduler. For inference, we use different sampling strategies for each rendering model. AR models use top-k of 768, NAR models use gumbel sampling and P-NAR models use gumbel sampling with temperature annealing from 4.5 to 1.0.

B Computational Cost

Sparsity can greatly reduce the computational cost. Supposing that each training data contains $(H \times W \times F)$ patches, and defining nearby size $(e^h, e^w, e^f)$. Our proposed nearby context pool can make the computational complexity $O((HWF)(e^h e^w e^f))$ instead of $O((HWF)^2)$ in Transformer. In addition, due to “render-and-optimize” strategy, the training input of each step is just the sequence of one patch, so our training complexity of one step is just $O(e^h e^w e^f)$ to support larger batch size.

Benefit from above design. We trained NUWA-Infinity (Large) for 1.25 days and NUWA-Infinity (Base) for 0.35 days on 64 A100 GPUs. Since video generation additionally needs to consider temporal information, we trained NUWA-Infinity-V (Large) for 3.7 days and NUWA-Infinity-V (Base) for 1 day on 64 A100 GPUs.
C Details for Arbitrary Direction

To support generation in arbitrary direction, we train NUWA-Infinity with different orders of patch-level sequence. Our default temporal direction is forward, and there are four basic spatial directions including right, left, down and up.

We design four directions in Fig. 1a. During training we randomly sample one direction from these four directions to flatten patches. For example, up extension requires the rendering window to move to the right and then move up. In addition, we propose a loop circle way in Fig. 1b to iteratively use the basic four directions to extend images in arbitrary direction at the same time, and the default order of circle direction is left, down, right and then up.

D One-shot Training

One-shot training means the model trained from scratch with one training sample. The RiverSide of Qingming Festival on the homepage of this paper and Fig. 6 are the results of one-shot training. The “render-and-optimize” strategy can train each patch separately, it allows us make the most of every small area of a large image. Furthermore, to avoid generating the original image directly, we use label smoothing=0.15 and more data augmentations, additionally including Mirror, Rotation, Noise and Shear.

E Limitations

While NUWA-Infinity has significant performance in infinite visual synthesis, it still has the following limitations:

1. Our approach is a patch-level autoregressive generation model, the minimum granularity for generating images and videos is limited by the size of the patch, such as 128 × 128 and 256 × 256. Although we can generate larger results and crop them, this has some waste of computing resources and inconvenience in operation.

2. Although the context can be transferred implicitly through the pool, as the length of the generating sequence increases, long-distant information will inevitably be forgotten. Fig. 11 shows that the background color of the last frame has been inconsistent with the input frame.

3. When expanding images and videos spatially or temporally, we scan a given area serially to produce the context, not in parallel. This leads to an increase in the inference time.
Figure 2: Text-to-Image 1024×1024 samples.
a red sky with clouds over a city  

a green aurora borealis above a body of water  

an aerial view of a beach and ocean  

a close up of a mossy surface  

a close up of a water splashing out of a cave  

a close up of a beach with rocks and water  

a waterfall with a rock pile  

a forest filled with leaves and trees  

a foggy forest with trees in the foreground

Figure 3: Text-to-Image 1024×1024 samples.
a field with a house and a cloudy sky

a large lake surrounded by green vegetation

a desert landscape with trees and mountains in the background

a cliff with a large canyon

Figure 4: Text-to-Image 1024x4096 samples.
Figure 5: Text-to-Image 1024x4096 samples.
Figure 6: One-Shot Training 1024×2048 samples. Note that the last row shows only the partial image.
a tree branch with green leaves and a forest in the background

a fallen tree over a waterfall in a forest

a house on a snowy hill under a starry sky

a city skyline with lights reflecting in the water

a cliff with a green field and a blue ocean

a waterfall with a green river in the background

a group of trees in a field with leaves

a dirt road that is on a hill

Figure 7: Image Extension 1024x2048 samples.
Figure 8: Image Extension 2048×1024 samples.
The Starry Night

The Gleaners

Figure 9: Arbitrary Image Extension 1536x3328 samples.
Figure 10: Image-to-Video 1024×1024×40 samples
Figure 11: Image-to-Video 1024×1024×40 samples