nature portfolio

https://doi.org/10.1038/s41586-025-10041-x

Supplementary information

Multimodal learning with next-token prediction for large multimodal models

In the format provided by the authors and unedited

Supplementary Information for: Multimodal learning with next-token prediction for large multimodal models

Emu3 Team
https://emu.baai.ac.cn

Contents

1 Vision Tokenizer Analysis

Video compression metrics. Table S1 presents the configurations of **Emu3** vision tokenizer. We report LPIPS (computed with AlexNet features), PSNR, and SSIM scores in Table S2 using an evaluation dataset of 3,172 videos from Pexels[1]. The videos were reconstructed over 5 seconds while maintaining the aspect ratio. During evaluation, original and reconstructed videos were resized and cropped based on the shorter side and uniformly sampled with 8 frames at 12 FPS.

Configurations	VisionTokenizer
Pretrained Weights	SBER-MoVQGAN-270M [1]
Codebook Size	32768
Latent Size	4
Compression	$4 \times 8 \times 8$

Table S1: Vision tokenizer configurations.

Video Resolution	LPIPS↓	PSNR↑	SSIM↑
128×128	0.099	21.71	0.630
256×256	0.109	21.59	0.622
512×512	0.112	22.69	0.690
720×720	0.110	24.30	0.771

Table S2: Video compression metrics.

Impact of codebook size. We train a series of image tokenizers with varying codebook sizes, ranging from 4,096 to 65,536, on ImageNet [2] with image resolution of 256×256, to analyze the impact of the codebook size. Following SBER-MoVQGAN [1], we adopt the variant of 67M parameters, keeping all configurations identical except for the codebook size. Upon completing tokenizer training, we proceed to train a separate 775M-parameter image generation model for each tokenizer. All image generation models are evaluated using classifier-free guidance (CFG) of 1.80, with top-k set to the codebook size, top-p of 1.0, and a temperature of 0.98.

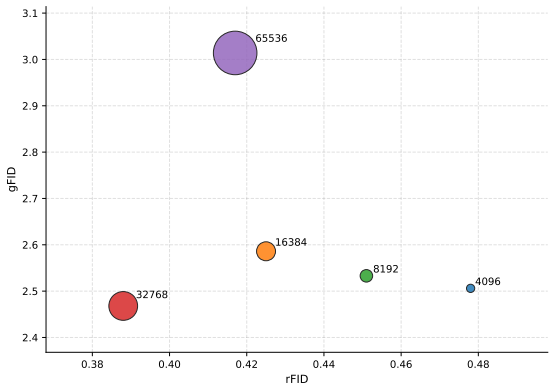


Figure S1: **Trade-off between reconstruction and generation across codebook sizes.** Scatter plot of reconstruction quality (rFID) versus generation quality (gFID) for different codebook sizes. Colored circles with number indicate codebook size. Codebook size of 32k achieves the best trade-off.

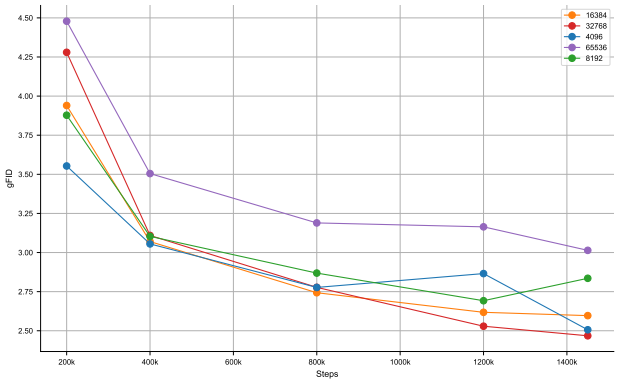


Figure S2: **Generation performance over training with different codebook sizes.** gFID curves over training steps for various codebook sizes. Each curve is color-coded by codebook size, revealing that excessively large vocabularies may hinder convergence.

[1] https://www.pexels.com/search/videos/videos

As shown in Fig. S1 and Table S3, increasing the codebook size consistently improves reconstruction metrics such as rFID, PSNR, and SSIM, up to a size of 32,768. The tokenizer with a codebook size of 32k largely outperforms the one with codebook size of 8k in both the reconstruction (rFID) and generation (gFID). However, performance drops significantly when the codebook size is further increased to 65,536, likely due to the increased difficulty of training with such a large codebook. Meanwhile, the codebook utilization also drops to 68%. These observations further support the hypothesis that training becomes more challenging as the codebook size grows excessively. Notably, the best gFID is also achieved with the codebook size of 32,768.

We also illustrate how gFID changes as both the codebook size and training steps increase in Fig. S2. The gFID consistently improves as training progresses for all experiments, with the model using a codebook size of 32,768 achieving the lowest gFID of 2.468.

Training VQ models with very large codebooks remains an open and challenging research problem. The aim of this work is not to exhaustively solve this large codebook challenge, but rather to study and quantify its effect within the proposed unified multimodal framework. We believe the observations provide useful empirical evidence for the community and motivate future research on improving VQ efficiency and utilization, which would in turn further strengthen the proposed next-token prediction paradigm.

codebook size	**utilization**	**rFID(↓)**	**PSNR(↑)**	**SSIM(↑)**	**gFID(↓)**
4,096	100%	0.478	24.38	0.7128	2.506
8,192	100%	0.451	24.62	0.7205	2.533
16,384	100%	0.425	24.75	0.7265	2.586
32,768	100%	**0.388**	**25.12**	**0.7404**	**2.468**
65,536	68%	0.417	24.91	0.7293	3.014

Table S3: **Comparison of vision tokenizers with varying codebook sizes.** Increasing the codebook size consistently improves both reconstruction quality (rFID, PSNR, SSIM) and generation performance (gFID), with gains observed up to a size of 32,768. However, further increasing the size to 65,536 leads to a significant drop in codebook utilization (68%), resulting in degraded performance.

	Resolution	**Latent Size**	**Temporal Ratio**	**Spatial Ratio**	**rFVD(↓)**	**PSNR(↑)**
Video MoVQ	320×240	$4 \times 40 \times 30$	4	8×8	27.893	27.546
Image MoVQ	320×240	$16 \times 40 \times 30$	-	8×8	**26.675**	**30.499**
	160×120	$16 \times 20 \times 15$			139.930	26.718

Table S4: **Unified video tokenizer vs. standalone image tokenizer.** Zero-shot reconstruction performance on 16-frame clips from UCF-101. The video tokenizer achieves competitive performance using 4× fewer latent tokens compared to the image tokenizer at the same resolution. When the image tokenizer is applied at a lower resolution to match the total latent count, its reconstruction quality drops significantly.

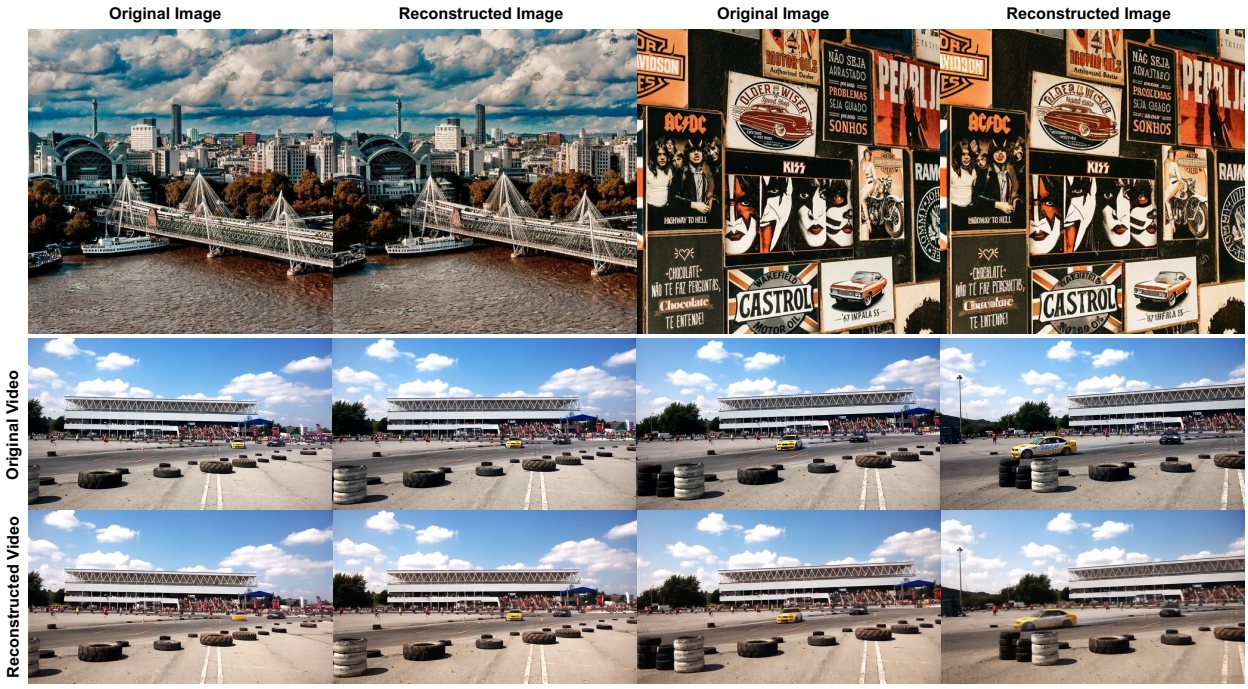


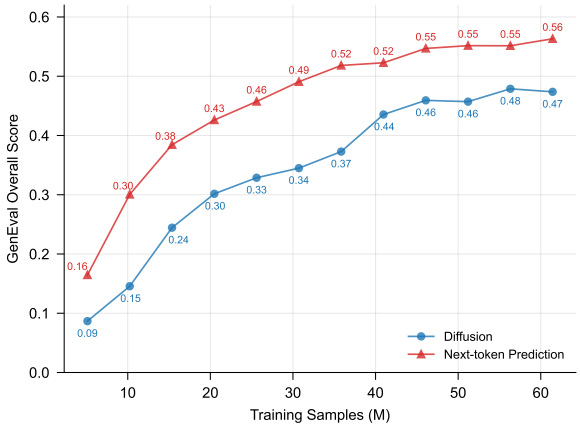
Figure S3: **Failure cases in extreme scenarios with dense details or large motion.**

Comparison between unified tokenizer and standalone image tokenizer. To evaluate the effectiveness of our unified video tokenizer, we compare its video reconstruction performance on UCF-101 [3] with the image tokenizer counterpart, which we use the SBER-MoVQ model with 270M parameters. We randomly sample 16 consecutive frames from each video in UCF-101. As shown in Table S4, under the same input resolution, our video tokenizer achieves comparable rFVD (27.893 vs. 26.675) and PSNR (27.546 vs. 30.499) while *using* $4\times$ *fewer tokens*. Moreover, when using the same number of latent tokens, the unified video tokenizer significantly outperforms the standalone image tokenizer, especially in terms of rFVD (27.893 vs. 139.930), demonstrating both its efficiency and effectiveness. Besides, we also visualize the typical failure cases as shown in Fig. S3, which mainly arise in extreme scenarios with exceptionally dense high-frequency details or pronounced motion patterns, where achieving faithful reconstruction remains notably challenging.

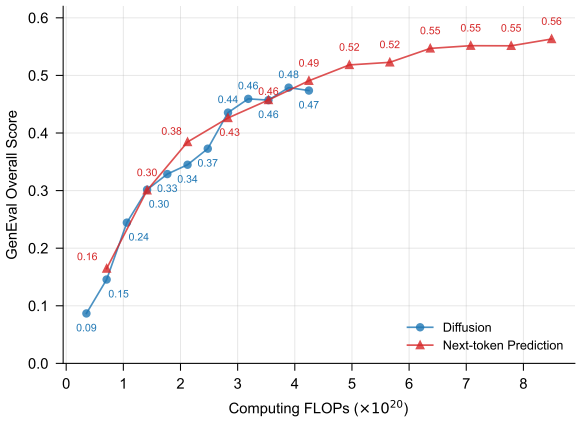
2 Architecture Analysis

2.1 Ablation Experiments on Architecture

We compare with vision–language architectures (encoder-based and continuous-space encoder-free) and diffusion baselines such as Text encoder(Flan-T5-XL) + Diffusion Transfromer(DiT). We show that decoder-only token prediction architecture trained without any pretrained vision or language components can outperform or match traditional pipelines that rely on strong unimodal priors, and offering a more unified, general-purpose design. This finding challenges the prevailing assumption that compositional or diffusion-based models are inherently superior for multimodal learning.



(a) GenEval overall scores as a function of the number of training samples.



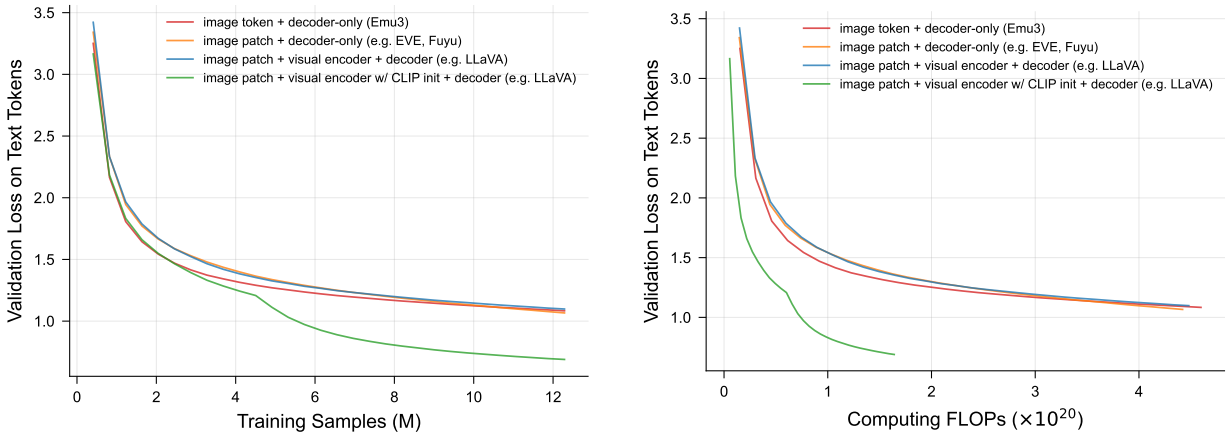
(b) GenEval overall scores as a function of estimated training compute (FLOPs, $\times 10^{20}$).

Figure S4: **Comparison of latent diffusion and next-token prediction paradigms.** The two plots illustrate how GenEval overall performance scales with (a) the amount of training data and (b) the estimated training compute budget.

Architectural comparisons with diffusion models. To fairly compare the next-token prediction paradigm with diffusion models for visual generation tasks, we use Flan-T5-XL [4] as the text encoder and train both a 1.5B diffusion transformer [5, 6] and a 1.5B decoder-only transformer [7] on the OpenImages [8] dataset. The diffusion model leverages the VAE from SDXL [9], while the decoder-only transformer uses the video tokenizer in **Emu3** to encode images into latent tokens. Both models are trained with identical configurations, including a linear warm-up of 2,235 steps, a constant learning rate of 1e-4, and a global batch size of 1,024. In our comparisons, both models begin with 1,536 multimodal tokens. The decoder-only transformer processes the full sequence, whereas the diffusion model follows the standard DiT convention by applying a patch-embedding layer first to the visual tokens, reducing the token length to 768 before fed into the diffusion transformer blocks. Estimated training FLOPs are reported using the approximation $\mathrm{FLOPs} \approx 6ND$, where N denotes the number of transformer parameters and D denotes the total number of multimodal tokens processed.

As shown in Fig. S4, the next-token prediction model converges faster than the diffusion counterpart given equal training samples; under equal training compute, the next-token prediction model continues to match the diffusion counterpart in GenEval overall score. These observations indicate that next-token prediction is competitive with latent diffusion for visual generation tasks, challenging the prevailing belief that diffusion architectures are inherently superior for visual generation. Moreover, the next-token paradigm has the potential to provide a unified, sequence-based framework that more naturally accommodates multimodal tasks.

Architectural comparisons with encoder+LLM compositional paradigm. To fairly evaluate different vision-language architectures, we compare four model variants with comparable parameter counts and training samples on the image-to-text validation set (i.e., an image understanding task), as shown in Fig. S5. We explicitly report both trends based on the size of the training samples and the training computation. The models compared are: (1) a decoder-only model that consumes discrete image tokens as input (Emu3 variant, 1.22B parameters); (2) an early-fusion decoder-only



(a) Validation loss on text tokens as a function of the number of training samples.

(b) Validation loss on text tokens as a function of the estimated training compute (FLOPs, $\times 10^{20}$).

Figure S5: **Comparisons with encoder–LLM compositional paradigms.** The left plot illustrates how the validation loss on text tokens scales with the size of the training samples, while the right plot shows the corresponding scaling with respect to the estimated amount of training compute. The compared architectures include: (1) a decoder-only model using discrete image tokens as input (Emu3); (2) a decoder-only model consuming continuous image patches (e.g., EVE, Fuyu); (3) a late-fusion model employing a vision encoder and a decoder (e.g., LLaVA without pretraining); and (4) its counterpart initialized with a pretrained CLIP encoder (e.g., LLaVA). Training compute is estimated as $6ND$ for early-fusion models, and $6(N_v D_v + ND)$ for late-fusion models.

model using continuous image patch embeddings (EVE-style variant [10], 1.17B); (3) a late-fusion architecture comprising a vision encoder and decoder (LLaVA-style variant [11], 1.22B = 1.05B decoder + 0.17B vision encoder); and (4) a late-fusion architecture initialized with a pretrained CLIP vision encoder (LLaVA-style variant, 1.35B = 1.05B + 0.30B).

The above models are trained without any pretrained LLM initialization. All configurations are trained with an identical pipeline implemented on the EVE codebase [10] using the HuggingFace Trainer [12]. We use a global batch size of $1{,}024$, a base learning rate of 1×10^{-4} with cosine decay scheduling, and $12{,}000$ training steps; supervision is applied only to text tokens via next-token prediction. The first three configurations employ an $8\times$ visual downsampling with 512×512 inputs and a context length of $5{,}120$. The CLIP-initialized late-fusion variant preserve CLIP's native $14\times$ downsampling and 224×224 resolution with a shorter context length of $2{,}048$, which requires less computation when trained on the same number of samples. Accordingly, we have revised the reported computation estimates to precisely reflect this specific configuration. All models are trained on the EVE-33M multimodal corpus [10], and evaluated on a held-out validation set of $1,024$ samples under comparable parameter. FLOPs are estimated as $6ND$ for early-fusion models and $6(N_v D_v + ND)$ for late-fusion models, where N denotes the number of transformer decoder parameters, D denotes the number of multimodal tokens, N_v denotes the number of vision encoder parameters, and D_v denotes the number of vision encoder tokens.

The late-fusion LLaVA-style model initialized with a pretrained CLIP(ViT-L/14) vision encoder attains substantially lower validation loss and converges faster than the other variants. We

attribute this to CLIP's extensive image–text pretraining (hundreds of millions of samples), which yields strong, task-relevant visual representations and likely explains the common perception that encoder+LLM compositions excel in multimodal understanding. Notably, when that pretrained advantage is removed (i.e., comparing the first three experiments trained from scratch), the apparent superiority of the encoder-based compositional architecture largely disappears. The decoder-only next-token prediction model not only achieves comparable performance but also converges faster, challenging the prevailing belief that encoder+LLM architectures are inherently superior for multimodal understanding. While the EVE-style early-fusion model also adopts a decoder-only architecture, it lacks generative capability. In summary, the next-token prediction architecture supports both understanding and generation, and can match or even surpass compositional encoder+LLM paradigms in learning efficiency when compared under equal conditions. We further expect the advantage of a pretrained vision encoder to diminish as the scale of end-to-end multimodal pretraining increases.

2.2 Flexible Prediction Paradigm

To further demonstrate the robustness and flexibility of our approach, we extended **Emu3** to different token prediction orders (e.g., diagonal, block-raster, spiral-in) and found that it transfers very fast and generalizes well across these settings.

Order	Initialization	Score (↑)
Raster	Pretrained	0.66
Diagonal	Pretrained	0.61
	From Scratch	0.17
Block-Raster	Pretrained	0.66
	From Scratch	0.18
Spiral-In	Pretrained	0.60
	From Scratch	0.16

Table S5: **Ablation study demonstrating the flexibility of Emu3 across different token prediction orders.** All transfer experiments are fine-tuned for 50B tokens. "Pretrained" denotes initialization from the Emu3 checkpoint trained with raster order. GenEval [13] overall score results show that the Emu3 trained with raster next-token prediction can rapidly adapt and maintain high performance across various token orders, highlighting the generality and robustness of the framework.

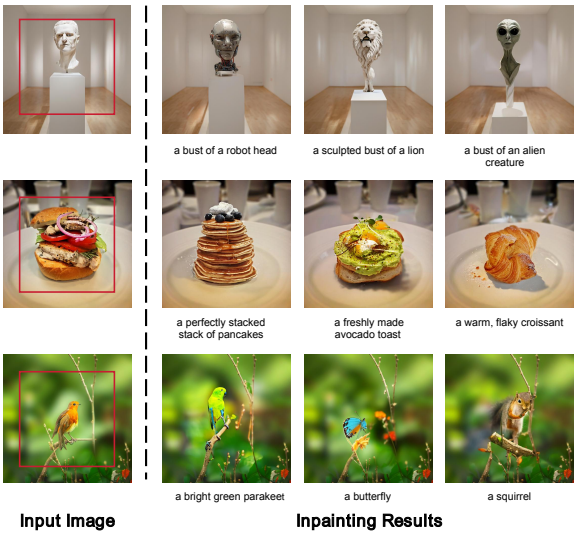


Figure S6: **Zero-shot image inpainting with Emu3 using spiral-in token order.** Given the input images (left of each row) and corresponding prompts, Emu3 accurately fills the masked regions within the bounding boxes, generating semantically aligned content without task-specific fine-tuning.

(1) **Raster order**: standard left-to-right, top-to-bottom scan of image tokens. (2) **Diagonal order**: tokens are predicted along diagonals from the top-left to bottom-right, increasing spatial

dispersion of context. (3) **Block-raster order**: raster scan within fixed-size local blocks (8×8 tokens per block), followed by sequential traversal of the blocks, which encourages local consistency before global composition. (4) **Spiral-in order**: decoding begins at the image boundaries and proceeds inward in a spiral pattern, prioritizing outer context before filling the center.

Unlike the conventional raster-scan decoding, these alternative token orders modify the autoregressive dependency structure and alter the spatial locality of context tokens, thereby posing a more challenging learning and generalization problem. Our approach leverages the same pretrained Emu3 model, originally optimized with raster next-token prediction, and fine-tunes it on the new token orders. All transfer experiments follow the training recipe of Emu3-Gen for fairness: The sequence length is 5,120 tokens, global batch size 512, learning rate 1×10^{-5} with cosine decay and warmup fraction 0.002, and Adam optimizer with weight decay 0.1. Each model is fine-tuned for 50B tokens on data uniformly sampled from the Emu3-Gen SFT dataset, with full-token supervision applied throughout.

We observe that Emu3's inductive biases, learned from large-scale raster training, transfer effectively to novel token orders, resulting in significantly higher performance than training from scratch (Table S5). We further validate functional utility through zero-shot image inpainting (Fig. S6). The spiral-in order, in particular, aligns with the spatial requirements of region completion, enabling Emu3 to generate coherent fills without task-specific tuning, underscoring the general-purpose adaptability of the backbone.

3 Pretraining Details

3.1 Data Details

Emu3 is pretrained from scratch on a mix of language, image, and video data. Details of data construction including sources, filtering and preprocessing are provided in Table S6.

Modality	Data Source	Processing Steps & Filtering Criteria
Text	- Aquila corpus [14] - FineWeb-Edu [15]	- Direct use
Image Data	- Open-source web data [16]	- Extract features using EVA-CLIP [17] - Apply K-means clustering to obtain representative image clusters - Caption generation using our image captioner
Image Data	- Open-source web data [18, 19, 20] - Open-source image data [8, 21] - AI-generated image data [22, 23] - In-house Emu3 image data	- Resolution Filter (remove resolution<512×512) - Aesthetic filtering by LAION aesthetic score [18] (remove aesthetic score<5.2) - Keep aesthetic score ≥5.5 - For 5.2 ≤ aesthetic score < 5.5: exclude text-rich & monochrome images - Caption generation using our image captioner
Video-Text	- Open-source video data [24, 25] - In-house Emu3 video data	- Scene-based splitting via PySceneDetect [26] (remove clip length<5s) - Removed text-rich clips using PaddleOCR [27] (remove text area>1%) - Motion filtering by RAFT [28] (remove optical flow score<8) - Aesthetic filtering by LAION aesthetic score [18] (removed aesthetic score<5) - Clips captioned using our video captioner

Table S6: Dataset construction including data sources and preprocessing steps of pretraining data.

Language Data. We use the same language data as in Aquila [14], supplemented with high-quality open-source data FineWeb-Edu [15]; this combined corpus consists of both Chinese and English data.

Image Data. We curate a large-scale image-text dataset comprising open-source web data, AI-generated data, and high-quality in-house data. The filtering process involves several key steps: **1**) We apply a resolution filter, discarding samples with a resolution below 512×512 pixels. **2**) We assess the aesthetic quality of each image using the LAION-AI aesthetic predictor[2], excluding images with scores below 5.5 to ensure the overall aesthetic quality. **3**) For images that did not pass the aesthetic filter($5.2 \leq$ aesthetic score < 5.5), we employ text detection[3] and color filtering to retain non-monochromatic images and those with minimal text, improving the filtering recall of open-world images. **4**) Additionally, we prepare supplementary data for image understanding. By following the data processing pipeline in DenseFusion [29], we extract millions of representative images that encompass a wide range of categories, including charts, tables, text-rich content, and more, sourced from diverse open-source web data.

To annotate the filtered dataset, we develop an image captioning model based on Emu2 [30] to construct dense synthetic captions. We leverage GPT-4V [31] with detailed prompts to generate approximately 1 million image-caption pairs. This annotated dataset is then used to fine-tune the Emu2-17B [30] model as our image captioner. Additionally, we utilize the open-source vLLM library [32] to accelerate the labeling process.

Video Data. We collect videos covering a wide range of categories, such as landscapes, animals, plants, games, and actions. These videos are preprocessed with a sophisticated pipeline [26] with the following four stages: **1**) We split the videos to scenes with PySceneDetect[4], employing both ContentDetector and ThresholdDetector to identify content changes and fade-in/out events, respectively. **2**) Text detection is performed using PaddleOCR[3] and clips with excessive text coverage were removed. To reduce computational costs, we sample video frames at 2 FPS and resize the shorter edge to 256. **3**) We further calculate the optical flow [28] to eliminate clips with minimal or extreme motion. As with the previous step, we sample and resize video frames for efficiency. The flow score is defined as the ratio between the average flow magnitude of all pixels and the shorter edge. We exclude clips with flow scores outside the acceptable range. **4**) Finally, we assess the aesthetic quality of each clip using the LAION-AI aesthetic predictor[1]. We sample three frames and get three scores for each clip, and clips whose lowest score is smaller than 5 are discarded.

We caption the filtered video clips using a video captioner trained based on our image captioner. The training data is initially labeled by GPT-4V [31]. For each video clip, we sample eight frames and create a detailed prompt for GPT-4V to describe both the content and motion within these frames. Some of the labeled data undergoes manual revision. We then fine-tune our image captioner on this labeled data to develop our video captioner. For large-scale deployment, we accelerate captioning with vLLM [32]. Clips shorter than 20 seconds are captioned using 12 evenly sampled frames, while longer clips are split into 10-20 second sub-clips, each captioned independently.

We analyze the distribution of the remaining clips. The duration distribution of the remaining clips is shown in Fig. S7. The flow score distribution of filtered clips is shown in Fig. S8.

[2] https://github.com/LAION-AI/aesthetic-predictor

[3] https://github.com/PaddlePaddle/PaddleOCR

[4] https://github.com/Breakthrough/PySceneDetect

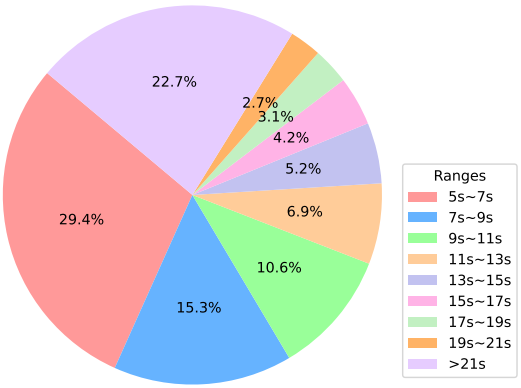


Figure S7: Duration distribution.

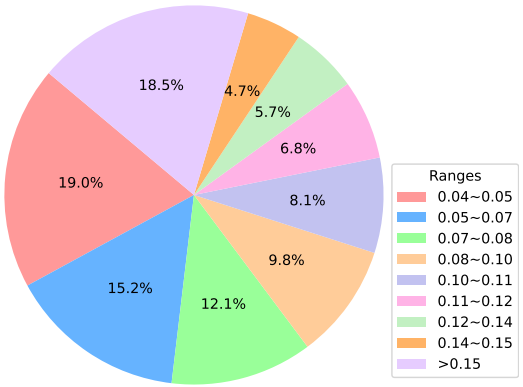


Figure S8: Flow score distribution.

3.2 Training Details

3.2.1 Training Cost

Table S7 summarizes the training computations. The model contains 8.49 billion parameters, with 6.98B in transformer layers and 1.51B in embedding layers, and is trained over three stages. Stage 1 and Stage 2 share the same setup and respectively consume 96k GPU-hours, running at 3.0 seconds per iteration and 370 TFLOP/s per GPU. Stage 3, which employs longer sequences (length 65,536), is substantially more compute-intensive, requiring 25.1 seconds per iteration at 391 TFLOP/s per GPU, totaling 803k GPU-hours. In total, the pretraining consumes approximately 995k GPU hours. The training cost calculation is based on a cluster of 1,280 high-performance GPUs, each equivalent to 80GB of memory and a theoretical bf16 computational capacity of 1,979 teraFLOPS.

	Stage 1	**Stage 2**	**Stage 3**
Training Performance			
Elapsed time per iteration (s)	3.0	3.0	25.1
TFLOP/s per GPU	370	370	391
Equivalent GPU-hours	96k	96k	803k
Model Parameters			
Model size (B)	8.49 (6.98 Transformer + 1.51 Embedding)		
Total Training Cost			
Equivalent GPU-hours	995k		

Table S7: **Training computations for Emu3 pretraining.**

3.2.2 Training Recipe

Training video data demands a longer context length, which causes training inefficiency. We thus adopt a multi-stage pretraining approach: the first two stages train on image data for efficient

initialization, and the third stage extends the context length and integrates video data. To facilitate training, we employ a combination of tensor parallelism (TP), context parallelism (CP), and data parallelism (DP). We simultaneously pack text-image data into the maximum context length to fully utilize computational resources, while ensuring that complete images are not segmented during the packing process.Table S8 details the training pipeline, including stage configurations, parallelism strategies, loss weights, optimization settings, and training steps. We have updated the reported training configurations to ensure full consistency.

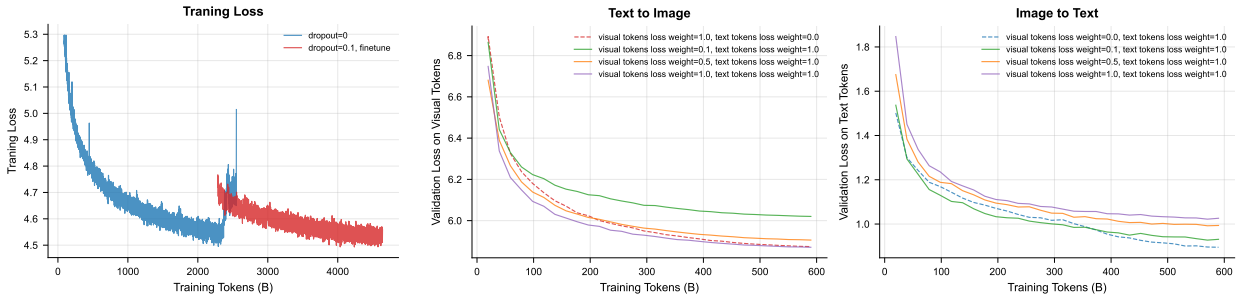
	Stage 1	**Stage 2**	**Stage 3**
Hyperparameters			
Learning rate	1×10^{-4}	5×10^{-5}	5×10^{-5}
LR scheduler	Cosine	Constant	Constant
Weight decay		0.1	
Dropout	0.0	0.1	0.1
Gradient norm clip	1.0	5.0	5.0
Optimizer	AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 1.0 \times 10^{-6}$)		
Loss weight (visual : text)	0.5 : 1	0.5 : 1	0.5 : 1
Warm-up steps	3,600	1,800	1,800
Training steps	90K	90K	90K
Sequence length	5,120	5,120	65,536
Training seen tokens	2.4T	2.4T	7.5T
Resolution	512	512	512
Parallelism Strategies			
Tensor parallelism	2	2	2
Pipeline parallelism	1	1	1
Context parallelism	1	1	4
Data sampling ratio			
Text	0.2	0.2	0.1
Image-Text pair (T2I)	0.64	0.64	0.08
Image-Text pair (I2T)	0.16	0.16	0.02
Video-Text pair (T2V)	0.0	0.0	0.64
Video-Text pair (V2T)	0.0	0.0	0.16

Table S8: Training recipe for **Emu3** pretraining.

3.2.3 Experiments on Training Settings

Large-scale unified multimodal learning is highly sensitive due to the diverse distributions of multimodal data. An improper recipe easily leads to training collapse, underscoring the fundamental difficulty of stable optimization at scale. We conduct ablation experiments of several key factors below.

Multimodal dropout for training stability. During pretraining of **Emu3**, we initially trained the model with a dropout rate of 0 (Stage 1). However, as shown in Fig. S9a, the training became unstable and eventually collapsed. We conducted extensive ablations to address this issue, exploring z-loss, adding masked tokens, reducing the learning rate for fine-tuning, skipping optimizer state loading, label smoothing, and packing-based training. None of these strategies effectively resolved the instability. In contrast, as illustrated in Fig. S9a, applying a small dropout rate (`dropout`=0.1)



(a) **Effect of dropout on training stability during Emu3 pretraining.** Training loss curves across Stage 1 and Stage 2 are shown.

(b) **Ablation on token loss weighting.** We examine the effect of balancing visual and text token losses during pretraining. The curves show validation loss on generation and understanding tasks under different weight configurations. Dashed lines represent settings with one loss weight set to zero, resulting to single-task training.

Figure S9: **Impact of dropout and modality loss weight on Emu3 pretraining.** Proper choices of dropout rate and loss balancing are critical to ensuring training stability and achieving competitive performance in large-scale multimodal learning.

and resuming training from a stable checkpoint in Stage 2 ensured stable convergence. This result highlights the critical role of dropout in maintaining stable convergence during large-scale multimodal learning.

Token loss weight. We ablate the weights of visual and text token losses during pretraining. As shown in Fig. S9b, visual-only or text-only supervision optimizes task-specific performance at the expense of cross-modal generalization. Joint supervision (visual: 1.0, text: 1.0) achieves the best generation performance, demonstrating the benefit of unified learning. However, it underperforms on understanding compared to the text-only setting. To balance both tasks, we adopt a moderate configuration (visual: 0.5, text: 1.0), which achieves competitive results on both tasks. This ablation underscores the necessity of careful loss balancing in large-scale multimodal learning, where naive parameter choices can lead to degraded performance or task bias. All experiments are conducted using the same transformer architectural configuration as Qwen2.5-1.5B [7]. We train with a global batch size of $3{,}840$ and an initial learning rate of 5×10^{-4} for $30{,}000$ steps under a cosine decay schedule. We adopt a context length of $5{,}120$ tokens. The training corpus is a high-quality subset of our pretraining data, with data-type proportions matched to those in Stage 1.

LLM-based initialization and MoE architecture. While the effectiveness of next-token prediction in large language models (LLMs) is well-established, our work focuses on exploring whether this paradigm can serve as a unified and scalable solution for multimodal learning, beyond language alone. To this end, we deliberately avoid incorporating strong priors from pretrained LLMs in our primary experiments, so as to clearly evaluate the capability of next-token prediction from scratch in a multimodal setting. That said, we also conducted controlled experiments with LLM-based initialization. All experiments follow the same training setup as the ablation experiments of token loss weight.

As shown in Fig. S10, we show that using a pretrained LLM to initialize the language compo-

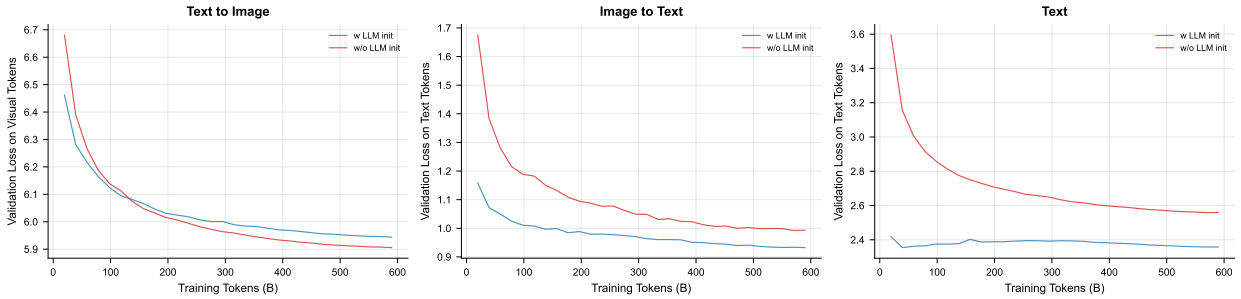


Figure S10: **Ablation experiments with LLM-based initialization on a 1.5B model.** We compare validation loss curves across three tasks: (1) vision token prediction loss on the text-to-image validation set, (2) text token prediction loss on the image-to-text validation set, and (3) text token prediction loss on the text-only validation set. Models initialized with a pretrained LLM (i.e., Qwen2.5-1.5B [7]) show faster convergence in early training stages, particularly for text-related losses. However, the performance gap narrows over time. The model without LLM initialization eventually achieves comparable or even better convergence on vision token loss, demonstrating the robustness of our unified multimodal learning.

nent of our model does provide a better starting point in early training stages (e.g., faster initial loss reduction). However, as training progresses, the gap between models with and without LLM initialization diminishes. Both converge to comparable validation loss, and the later even converges better on vision token loss. This indicates that our multimodal learning is robust and effective even without relying on pretrained language representations. While incorporating LLMs may offer efficiency benefits in the short term, Emu3's design can scale effectively from scratch, supporting its potential as a general-purpose, unified multimodal learner.

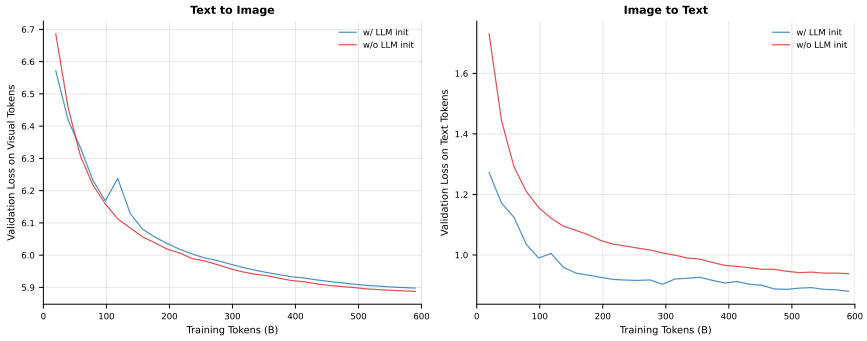


Figure S11: **Ablation experiments with LLM-based initialization on a 1.5B (active) MoE model.** Validation loss for MoE models with/without pretrained LLM initialization. Left: text-to-image generation; Right: image-to-text understanding. Pretrained initialization is detrimental to text-to-image. The MoE model is upcycled from Qwen2.5-1.5B.

We also explored a mixture-of-experts (MoE) architecture variant in our framework. We conducted controlled experiments with MoE architectures to further validate our findings. Fig. S11 presents validation loss trajectories for Text-to-Image (T2I) and Image-to-Text (I2T) tasks in MoE models, providing additional empirical evidence for our architectural decisions. In fact, loading

pretrained models can be detrimental to text-to-image tasks, as it introduces biases that hinder the model's ability to learn effective visual representations from scratch. Furthermore, our investigation into MoE architectures reveals significant efficiency limitations. Our implementation, which uses 4 experts with 1 activated per token, requires a 4x larger total parameter count in its expert layers compared to a dense equivalent to match the number of activated parameters. This design increases memory requirements, communication overhead, and training complexity.

Note that our MoE comparison was deliberately performed using a canonical and minimal MoE configuration to isolate the effects of initialization and model scale, rather than optimizing the routing or expert design itself. This setup enables a clean and controlled comparison between dense and sparse architectures under equivalent computational budgets. More advanced MoE variants (e.g., separating generation and understanding experts) may further improve expert utilization, but we chose to evaluate a minimal and well-understood MoE baseline to ensure a fair and interpretable comparison across architectures and initialization strategies.

These comparisons, combined with the observed asymmetry between understanding and generation tasks, support our design choice of a unified dense architecture with random initialization. Our comprehensive empirical analysis demonstrates that the unified next-token prediction paradigm, when properly scaled and trained from scratch, provides a robust and efficient foundation for general multimodal learning, eliminating pretrained initialization or complex architectural modifications.

4 Post-training Details

4.1 Text-to-Image Generation

4.1.1 Human Evaluation

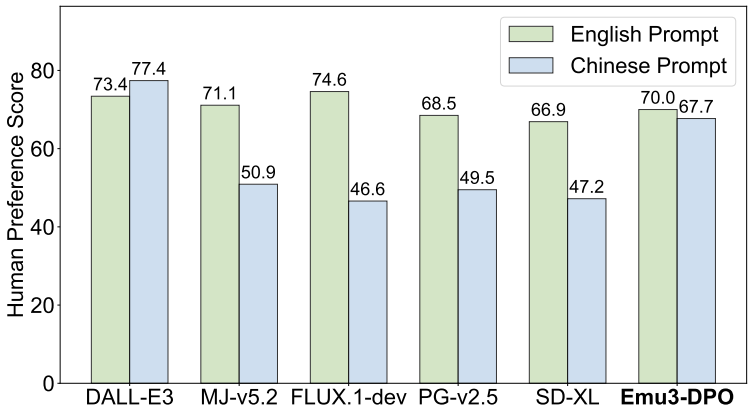


Figure S12: Human evaluation overall score comparison of closed and open generative image models under English and Chinese prompts.

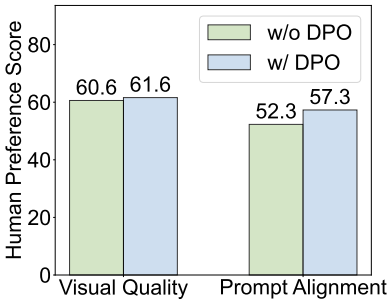


Figure S13: DPO improves visual quality and prompt alignment.

We conduct a human evaluation comparing the text-to-image generation capabilities of different models. A set of 100 diverse user prompts is collected, and each is evaluated by three inde-

pendent voters. The evaluation focuses on two main aspects: visual quality and prompt following, with a weighted score reflecting the overall performance. As shown in Fig. S12, we present a comparison of human preferences for current closed and open generative image models. The results indicate that **Emu3** achieves competitive performance comparable to SDXL and reaches a level on par with DALL-E 3 and MJ-v5.2 in terms of overall score. Furthermore, Fig. S13 demonstrates the impact of alignment through DPO fine-tuning, which effectively improves visual quality and prompt following.

4.1.2 Experimental Results and Evaluation Details

We present the performance of **Emu3** through automated metric evaluation on popular text-to-image benchmarks: MSCOCO-30K [33], GenEval [13], T2I-CompBench [34], and DPG-Bench [35]. The comparison results of **Emu3** against diffusion methods, autoregressive diffusion methods, and autoregressive-based methods across these four benchmarks are shown in Table S9. We report the results of GenEval and T2I-CompBench after employing a rewriter to expand short prompts. Following DALL-E 3, we also using GPT-4V as the rewriter. For all T2I evaluations, we set top-k to 16,384 and top-p to 1.0 for image generation. The output resolution for **Emu3** is 512 x 512. The output resolution for **Emu3-DPO** is 720 x 720.

Method	Text Pretrain	MSCOCO			GenEval	T2I-CompBench			DPG-Bench
		CLIP-I	CLIP-T	FID	Overall	Color	Shape	Texture	Average
Diffusion-based									
SDv1.5 [36]	CLIP ViT-L/14	0.667	0.302	9.93	0.43	0.3730	0.3646	0.4219	63.18
DALL-E 2 [37]	CLIP ViT-H/16	-	0.314	10.93	0.52	0.5750	0.5464	0.6374	-
SDv2.1 [36]	CLIP ViT-H/14	-	-	-	0.50	0.5694	0.4495	0.4982	-
SDXL [38]	CLIP ViT-bigG	0.674	0.310	-	0.55	0.6369	0.5408	0.5637	74.65
PixArt-alpha [39]	Flan-T5-XXL	-	-	7.32	0.48	0.6886	0.5582	0.7044	71.11
DALL-E 3 [40]	T5-XXL	-	0.320	-	0.67†	0.8110	0.6750	0.8070	83.50
SD3 [41]	C-G/14 + C-L/14 + T5-XXL‡	-	-	-	0.74	-	-	-	-
FLUX.1 [Dev] [42]	CLIP ViT-L/14 + T5-XXL	-	-	-	0.66	-	-	-	83.84
Autoregressive meets diffusion									
Emu [43]	LLaMA-7B	0.656	0.286	11.6	-	-	-	-	-
Show-o [44]	Phi-1.5	-	-	9.24	0.53	-	-	-	-
Transfusion [45]	-	-	-	6.78	0.63	-	-	-	-
Autoregressive-based									
Chameleon [46]	-	-	-	26.74	0.39	-	-	-	-
LlamaGen [47]	FLAN-T5 XL	-	-	-	0.32	-	-	-	-
Emu3	-	0.689	0.313	12.8	0.66†	0.7913†	0.5846†	0.7422†	80.60
Emu3-DPO	-	0.680	0.312	19.3	0.64†	0.7544†	0.5706†	0.7164†	81.60

Table S9: **Comparison with state-of-the-art models on text-to-image benchmarks.** We evaluate on MSCOCO-30K [33]; GenEval [13]; T2I-CompBench [34] and DPG-Bench [35]. † result is with rewriting. ‡ SD3 uses CLIP ViT-G/14, CLIP ViT-L/14 and T5-XXL as text encoder.

MSCOCO 30K. We present zero-shot CLIP score and FID of **Emu3** and **Emu3-DPO** on MSCOCO 30K in Table S9. Following [43], we randomly sample 30k prompts from the validation set and calculate the zero-shot FID [48]. We employ CLIP-ViT-B [49] to calculate the CLIP-T score to assess prompt-following ability. Additionally, we utilize CLIP-ViT-L [49] to compute the CLIP-I

score for measuring image similarity. For the DALL-E3 and DALL-E2, CLIP-T score is calculated on 4,096 samples. We adopt classifier-free guidance [50] for better generation quality. The guidance scale is set to 5.0. The results of other methods in the MSCOCO 30K are sourced from [43, 44, 45]

DPG-bench. To assess the ability to follow dense text, we compared our models with diffusion models on the DPG-Bench, which provides longer prompts containing more detailed information for evaluation. We measured DPG-bench follows ELLA [35] shown in the Table S9, and our model achieved an overall score of 81.60, which is higher than SDXL and PixArt-alpha, and is comparable to the results of Dalle-3. We utilized mPLUG-large model to evaluate the generated images according to the designated questions. The results of other methods in the DPG-Benchmark are sourced from [35, 51]. We generate 4 images for each prompt with guidance scale is 5.0.

GenEval. Following SD3 [41], we evaluate text-to-image generation capability of **Emu3** on the GenEval benchmark [13]. We present the scores for the GenEval benchmark in Table S10 across six dimensions including "Single Object", "Two Objects", "Counting", "Colors", "Position", "Color Attribute". We generate 4 images for each prompt with a guidance scale of 5.5. Following with Dalle-3, we also report our evaluation results utilizing GPT4-V as a rewriter. The results of other methods in the GenEval are sourced from [13, 44, 45, 41].

T2I-CompBench. Following the Dalle-3 [40], we report the scores of color binding, shape binding and texture binding in Table S10. We use the BLIP-VQA model to evaluate these results. We generate 10 images for each prompt with a guidance scale of 5.0. The results of other methods in the T2I CompBench are sourced from [40, 52, 39]

	GenEval							T2I-CompBench		
Method	Overall	Single Obj.	Two Obj.	Counting	Colors	Position	Color Attri.	Color	Shape	Texture
Diffusion-based										
DALL-E 2 [37]	0.52	0.94	0.66	0.49	0.77	0.10	0.19	0.5750	0.5464	0.6374
SDv1.5 [36]	0.43	0.97	0.38	0.35	0.76	0.04	0.06	0.3730	0.3646	0.4219
SDv2.1 [36]	0.50	0.98	0.51	0.44	0.85	0.07	0.17	0.5694	0.4495	0.4982
SDXL [38]	0.55	0.98	0.74	0.39	0.85	0.15	0.23	0.6369	0.5408	0.5637
PixArt-alpha [39]	0.48	0.98	0.50	0.44	0.80	0.08	0.07	0.6886	0.5582	0.7044
DALL-E 3 [40]	0.67	0.96	0.87	0.47	0.83	0.43	0.45	0.8110	0.6750	0.8070
SD3 [41]	0.74	0.99	0.94	0.72	0.89	0.33	0.60	-	-	-
Autoregressive meets diffusion										
Show-o [44]	0.53	0.95	0.52	0.49	0.82	0.11	0.28	-	-	-
Transfusion [45]	0.63	-	-	-	-	-	-	-	-	-
Autoregressive-based										
Chameleon [46]	0.39	-	-	-	-	-	-	-	-	-
LlamaGen [47]	0.32	0.71	0.34	0.21	0.58	0.07	0.04	-	-	-
Emu3	0.54	0.98	0.71	0.34	0.81	0.17	0.21	0.6107	0.4734	0.6178
+ Rewriter	0.66	0.99	0.81	0.42	0.80	0.49	0.45	0.7913	0.5846	0.7422
Emu3-DPO	0.52	0.98	0.69	0.33	0.78	0.15	0.16	0.5514	0.4641	0.5476
+ Rewriter	0.64	0.99	0.76	0.38	0.85	0.45	0.40	0.7544	0.5706	0.7164

Table S10: **Comparison with state-of-the-art models on GenEval and T2I CompBench.** Obj.: Object. Attri.: Attribute.

4.1.3 Visualization Results

Fig. S14 shows images generated by **Emu3** to showcase its capabilities. **Emu3** supports flexible resolutions, aspect ratios, and is capable of handling various styles.

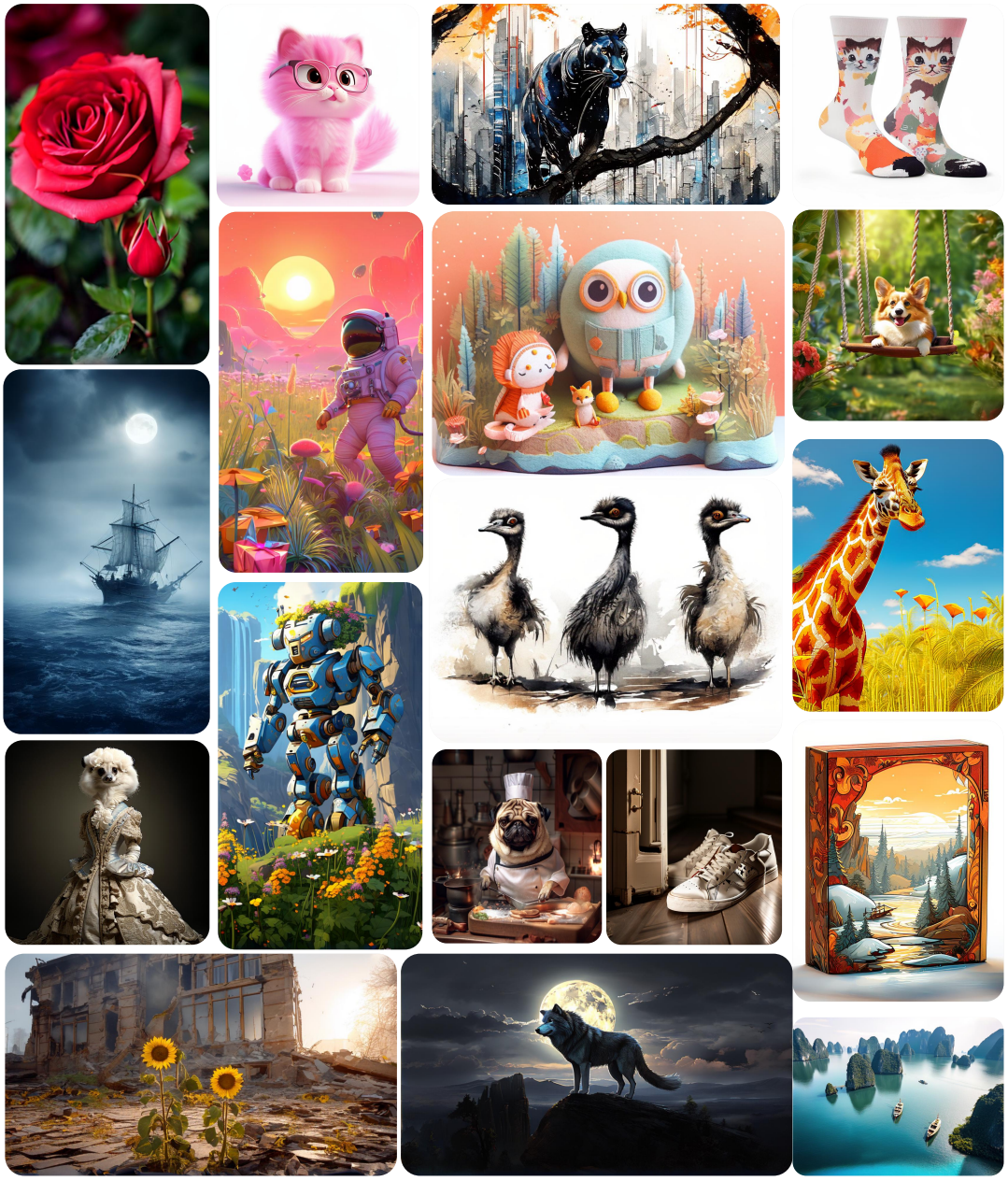


Figure S14: **Visualization results of Emu3 text-to-image generation.**

4.2 Text-to-Video Generation

4.2.1 Video Post Processing

To further improve the temporal consistency and visual quality, we apply stabilization and super resolution techniques to the generated videos. Video evaluation is also conducted on the processed videos. Specifically, we train specialized models for these two tasks.

Video Stabilization. We train the video stabilization model based on the temporal VAE of stable video diffusion [53]. The model is trained on our curated video data with a combined objective comprising L1 loss, LPIPS perceptual loss [54], GAN loss, and KL penalty [55, 56]. A training data pair consists of an autoencoded video clip output from our tokenizer and the groundtruth video clip, both having dimensions of $16 \times 256 \times 256$.

Super-Resolution. We implement a spatial-temporal unet model for super-resolution task, capable of upsampling any image or video clip by a factor of 4. We adopt the BlurPool [57] for downsample operations and sub-pixel [58] for upsample operations. The model is trained on random crops of $8 \times 256 \times 256$ from part of our curated videos, which have a resolution greater than 1024×1024, with a combined loss of L2 loss, LPIPS perceptual loss [54], and GAN loss.

4.2.2 Experimental Results and Evaluation Details

The stage 3 pretrained model of **Emu3** natively supports the generation of 5-second videos at 24 FPS. We conducted a quantitative comparison between **Emu3** and the 13 open-source and proprietary text-to-video models. The used benchmark is VBench [59], a comprehensive toolkit for evaluating video generation performance, which assesses the quality and semantic capabilities of each model across 16 dimensions. For inference parameters, we set top-k to 8,192, top-p to 1.0 and temperature to 0.95 for video generation.

Models	Type	Total score	Motion smoothness	Dynamic degree	Aesthetic quality	Object class	Multiple objects	Human action	Spatial relationship	Scene	Appearance style	Subject consistency	Background consistency
ModelScope [60]	Diff	75.75	95.79	66.39	56.39	82.25	38.98	92.4	33.68	39.26	25.67	89.87	95.29
LaVie [61]	Diff	77.08	96.38	49.72	54.94	91.82	33.32	96.8	34.09	52.69	23.56	91.41	97.47
OpenSoraPlan V1.1 [62]	Diff	78.00	98.28	47.72	56.85	76.3	40.35	86.80	53.11	27.17	22.90	95.73	96.73
Show-1 [63]	Diff	78.93	98.24	44.44	57.35	93.07	45.47	95.60	53.50	47.03	23.06	95.53	98.02
OpenSora V1.2 [64]	Diff	79.76	98.50	42.39	56.85	82.22	51.83	91.20	68.56	42.44	23.95	96.75	97.61
AnimateDiff-V2 [65]	Diff	80.27	97.76	40.83	67.16	90.90	36.88	92.60	34.60	50.19	22.42	95.30	97.68
Gen-2 [66]	Diff	80.58	99.58	18.89	66.96	90.92	55.47	89.20	66.91	48.91	19.34	97.61	97.61
Pika [67]	Diff	80.69	99.50	47.50	62.04	88.72	43.08	86.20	61.03	49.83	22.26	96.94	97.36
VideoCrafter-2.0 [68]	Diff	80.44	97.73	42.50	63.13	92.55	40.66	95.00	35.86	55.29	25.13	96.85	98.22
T2V-Turbo (VC2) [69]	Diff	81.01	97.34	49.17	63.04	93.96	54.65	95.20	38.67	55.58	24.42	96.28	97.02
CogVideoX-5B [70]	Diff	81.61	96.92	70.97	61.98	85.23	62.11	99.40	66.35	53.20	24.91	96.23	96.52
Kling (2024-07) [71]	Diff	81.85	99.40	46.94	61.21	87.24	68.05	93.40	73.03	50.86	19.62	98.33	97.60
Gen-3 [72]	Diff	82.32	99.23	60.14	63.34	87.81	53.64	96.4	65.09	54.57	24.31	97.10	96.62
HunyuanVideo [73]	Diff	83.24	98.99	70.83	60.36	86.10	68.55	94.40	68.68	53.88	19.80	97.37	97.76
Wan2.1-T2V-14B [74]	Diff	83.69	98.30	65.46	66.07	86.28	69.58	95.40	75.39	45.75	22.64	97.52	98.09
Emu3 (Ours)	AR	80.96	98.93	79.27	59.64	86.17	44.64	77.71	68.73	37.11	20.92	95.32	97.69

Table S11: **Comparison with state-of-the-art text-to-video models on VBench [59] benchmark.**. We selected 11 out of the 16 evaluation dimensions from VBench, along with the final score, for presentation. Except for Emu3, which is an autoregressive (AR) model, *all other publicly comparable method are diffusion (Diff) models*. The higher metrics indicate the better results.

Aside from **Emu3**, which adopts an autoregressive architecture, all other publicly comparable baselines are diffusion-based models. As shown in Table S11, **Emu3** attains competitive overall

performance relative to existing approaches. While it does not match the results reported by advanced proprietary systems such as Kling [71] and Gen-3 [72], its performance remains broadly comparable to that of well-established open-source text-to-video models. These observations indicate that **Emu3** can serve as a viable and unified approach for video generation.

4.2.3 Visualization Results

Fig. S15 presents qualitative examples of video generation, with 6 frames extracted from the first 3 seconds for showcase.

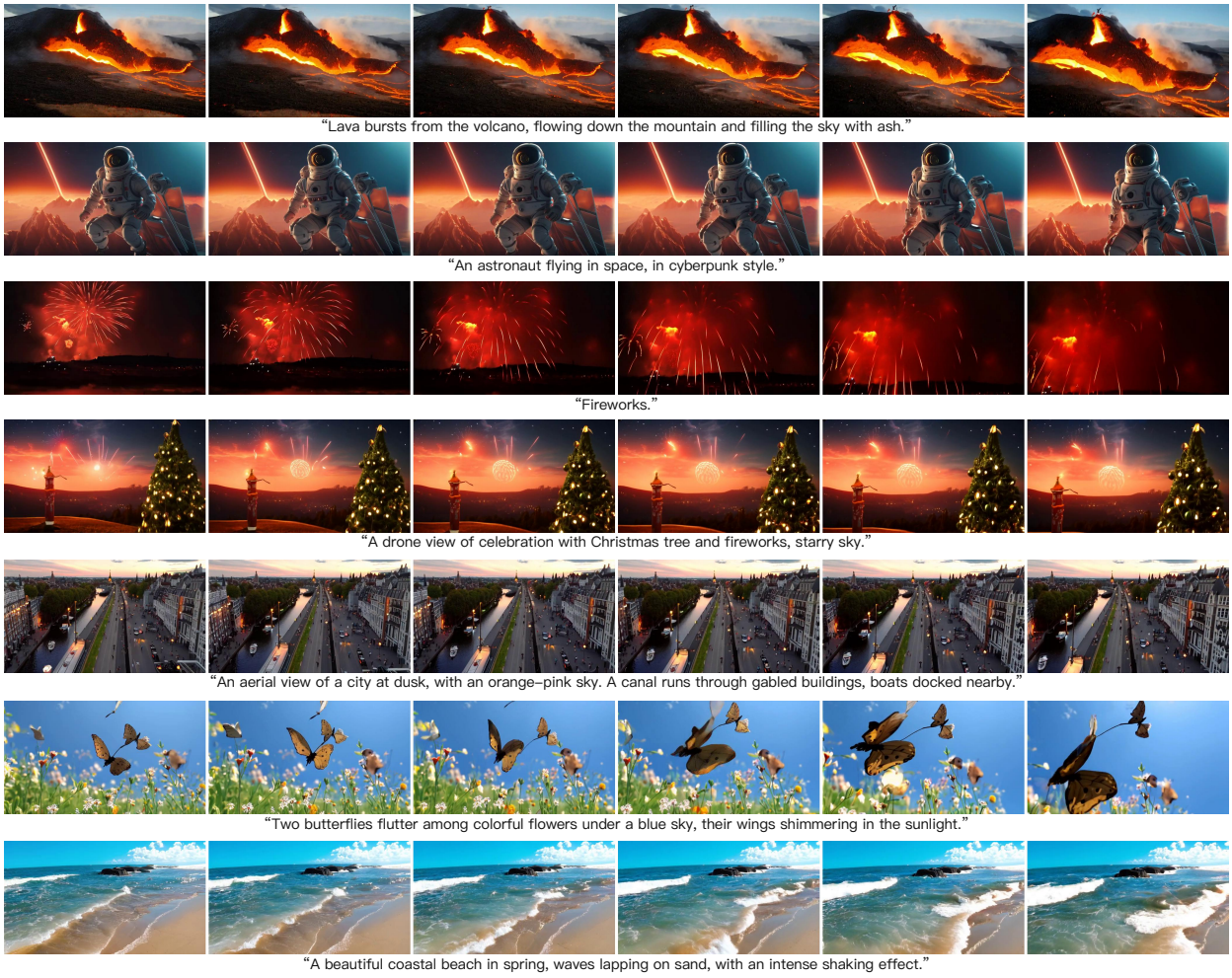


Figure S15: **Visualization results of Emu3 text-to-video generation.**

4.3 Vision-Language Understanding

Experimental Results and Evaluation Details. To assess the vision–language understanding capability of **Emu3**, we evaluate the model across a range of public vision–language benchmarks. The primary comparisons in Table S12 cover two categories of approaches: **1)** encoder-based methods that rely on pretrained CLIP vision encoders, and **2)** encoder-free methods that operate

without pretrained encoders. As a purely encoder-free approach, **Emu3** reaches a performance level comparable to representative models in both categories on several benchmarks. Notably, this is achieved without resorting to a specialized pretrained LLM or CLIP encoder, highlighting the conceptual simplicity and architectural unity of our framework. All benchmarks are evaluated using the `lmms-eval` library [75], with inference and evaluation performed under its recommended default settings. The input resolution ranges from 512×512 to 1024×1024, with aspect ratio preserved throughout.

Method	Pretrained-LLM	SEEDB	OCRB	MMV	POPE	VQAv2	GQA	SQA	TQA	CQA	DVQA	IVQA	AI2D	RWQA	MMMU	MMB
Encoder-based																
InstructBLIP [76]	Vicuna-7B	53.4	276	26.2	–	–	49.2	60.5	50.1	12.5	13.9	–	33.8	37.4	30.6	36.0
IDEFICS-9B [77]	LLaMA-7B	–	252	–	–	50.9	38.4	–	25.9	–	–	–	42.2	42.1	18.4	48.2
QwenVL-Chat [78]	Qwen-7B	58.2	488	–	–	78.2*	57.5*	68.2	61.5	49.8	66.3	–	45.9	49.3	35.9	60.6
LLaVA-1.5 [11]	Vicuna-7B	64.3	318	30.5	85.9	78.5*	62.0*	66.8	46.1	18.2	28.1	25.8	54.8	54.8	35.3	64.3
InternVL-Chat [79]	Vicuna-7B	–	–	–	86.4	79.3*	62.9*	–	57.0	–	–	–	–	–	–	–
mPLUG-Owl2 [80]	LLaMA2-7B	57.8	255	36.5	86.2	79.4*	56.1*	68.7	58.2	22.8	–	–	55.7	50.3	32.7	64.5
ShareGPT4V [81]	Vicuna-7B	–	371	37.6	–	80.6*	63.3*	68.4	60.4	21.3	–	–	58.0	54.9	37.2	68.8
LLaVA-1.6(HD) [82]	Vicuna-7B	64.7	532	43.9	86.5	81.8*	64.2*	70.2	64.9	54.8*	74.4*	37.1	66.6*	57.8	35.1	67.4
VILA [83]	LLaMA2-7B	61.1	–	34.9	85.5	80.8*	63.3*	73.7	66.6	–	–	–	–	–	–	68.9
Qwen2.5-VL [84]	Qwen2.5-7B	–	864	–	–	–	–	–	84.9	87.3	95.7	82.6	83.9	–	58.6	83.5
Encoder-free																
Fuyu-8B(HD) [85]	Persimmon-8B	–	–	21.4	74.1	74.2	–	–	–	–	–	–	64.5	–	27.9	10.7
Chameleon-MT-34B [46]	–	–	–	–	–	69.6	–	–	–	–	–	–	–	–	–	–
Show-o [44]	Phi-1.5-1.3B	–	–	–	73.8	59.3*	48.7*	–	–	–	–	–	–	–	25.1	–
EVE-7B(HD) [10]	Vicuna-7B	56.8	–	25.7	85.0	78.6*	62.6*	64.9	56.8	–	–	–	–	–	–	52.3
Emu3	–	68.2	687	37.2	85.2	75.1*	60.3*	89.2*	64.7	68.6*	76.3*	43.8*	70.0*	57.4	31.6	58.5

Table S12: **Comparison on vision-language benchmarks.** We collect evaluations including: SEEDB: SEEDBench-Img [86]; OCRB: OCRBench [87]; MMV: MMVet [88]; POPE [89]; VQAv2 [90]; GQA [91]; SQA: ScienceQA-Img [92]; TVQA: TextVQA [93]; CQA: ChartQA [94]; DVQA: DocVQA [95]; IVQA: InfoVQA [96]; AI2D [97]; RWQA: RealWorldQA [98]; MMMU [99]; MMB: MMBench [100]. * The images of related training datasets are observed during training.

4.4 Interleaved Image-Text Generation

We extend **Emu3** to interleaved image-text generation, where structured textual steps are accompanied by corresponding illustrative images in a single output sequence. Thanks to the flexibility and generability of the framework, we directly fine-tune the model to autoregressively generate such multimodal sequences in an end-to-end manner.

Data Format. In terms of data format for interleaved image-text generation, we construct the instruction tuning dataset as follows, where all contents that should be supervised are included within two special tokens (`<|extra_100|>` and `<|extra_101|>`). Among them, `<|VIS_PLH|>` serves as an image placeholder to indicate the actual position of the image tokens, and the format of each image is consistent with the pretraining stage.

The Data Format Example of Interleaved Image-text Generation

You are a helpful assistant.
USER:
How to cook Dry Pot Cauliflower?
ASSISTANT:
<|extra_100|>
1. Break the cauliflower into small pieces and soak them in salted water<|VIS_PLH|>
2. Heat a little oil in a pot and stir-fry the cauliflower until it releases its moisture<|VIS_PLH|>
3. In the same pot, stir-fry ginger, garlic, dried chili peppers, and Sichuan pepper sauce until red oil forms<|VIS_PLH|>
4. Add the cauliflower and continue stir-frying. Season with soy sauce, oyster sauce, vegetable essence, and a little salt<|VIS_PLH|>
5. Serve and enjoy!<|VIS_PLH|>
<|extra_101|>

Visualization Results. As demonstrated in Fig. S16, even with basic fine-tuning using limited interleaved image-text data, the model exhibits promising capabilities in generating interleaved image-text sequences. This result suggests that our framework possesses greater versatility and scalability compared to conventional single-modality models limited to text or image generation.

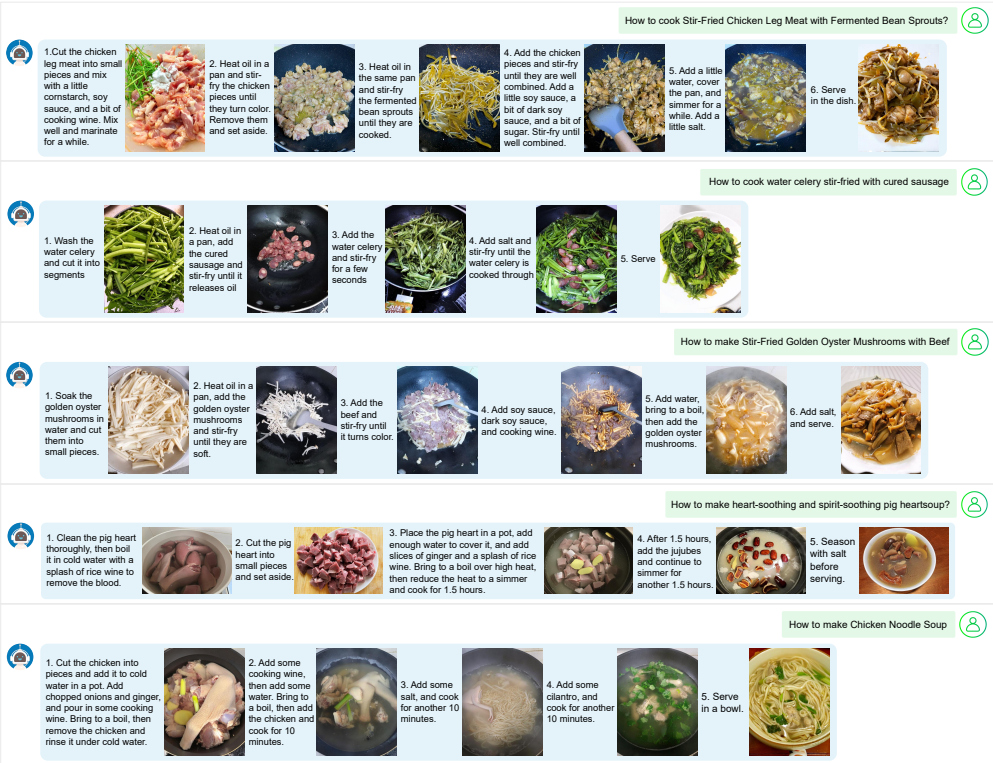


Figure S16: **Visualization of interleaved image-text generation results.**

5 Inference

5.1 Inference System

Our inference infrastructure is built on FlagScale [101], a multimodal serving system developed atop vLLM [32]. FlagScale inherits vLLM's key advantages, including the PagedAttention mechanism for fine-grained KV-cache management and continuous dynamic batching, which enables adaptive, per-token scheduling to maximize GPU utilization and minimize end-to-end latency. It is further accelerated by optimized CUDA kernels such as FlashAttention [102]. On this foundation, FlagScale extends the inference backend to support Classifier-Free Guidance (CFG) [50] for autoregressive multimodal generation. Specifically, we integrate CFG directly into the dynamic batching pipeline by co-feeding conditional and negative prompts within the same batch iteration. This design allows a unified forward pass at each decoding step, enabling correct logit fusion and guidance scaling without disrupting the batching semantics. Crucially, our CFG-aware extension incurs negligible overhead and preserves the low-latency, high-throughput characteristics of vLLM. As a result, FlagScale provides a scalable and efficient inference framework that supports fine-grained generation control across a wide range of multimodal tasks.

Task	Batch Size	vLLM		HF		Speed Up
		Time (s)	Throughput (token/s)	Time (s)	Throughput (token/s)	
Text-to-Image	1	68.11	120.62	259.69	31.64	3.81
	2	74.16	221.58	303.70	54.11	4.10
	4	82.92	396.34	379.74	86.54	4.58
	8	100.48	654.13	548.44	119.85	5.46
	16	135.22	972.16	905.58	145.16	6.70
	32	220.81	1190.70	OOM	OOM	-
	64	415.70	1264.92	OOM	OOM	-
	128	844.29	1245.59	OOM	OOM	-
Image-to-Text	1	1.17	3687.90	1.92	2252.09	1.64
	2	1.32	6554.10	2.20	3929.00	1.67
	4	1.50	11497.13	2.75	6266.80	1.83
	8	2.09	16541.42	4.00	8624.96	1.92
	16	3.26	21206.78	OOM	OOM	-
	32	5.78	23897.13	OOM	OOM	-
	64	10.15	27209.77	OOM	OOM	-
	128	20.90	26426.17	OOM	OOM	-
Text-to-Video	1	768.29	81.60	OOM	OOM	-
	2	1186.79	105.65	OOM	OOM	-
	4	1986.64	126.22	OOM	OOM	-
	8	3802.48	131.89	OOM	OOM	-
	16	7568.93	132.52	OOM	OOM	-

Table S13: **Comparison of inference latency and throughput between vLLM and Hugging Face Transformers**.

Table S13 presents a comparative analysis of inference latency and throughput between our vLLM-based backend and a standard Hugging Face (HF) Transformer [12] implementation across three representative tasks: Text-to-Image (Emu3-Gen with CFG, averaging 8,216 tokens per image), Image-to-Text (Emu3-Chat without CFG), and Text-to-Video (Emu3 with CFG, averaging

65,536 tokens per video). All experiments were conducted on a single GPU node. Our vLLM-based backend consistently outperforms the Hugging Face backend across text-to-image, image-to-text, and text-to-video tasks, offering higher throughput, supporting larger batches and longer contexts, and avoiding out-of-memory failures, thanks to its efficient KV-cache management and dynamic batching.

5.2 Inference Pseudocode

The pseudocode in Alg. 1 describes the inference process of **Emu3** for generating multimodal outputs, including text, images, and videos, from input sequences of the same types.

Algorithm 1 Multimodal token sequence generation

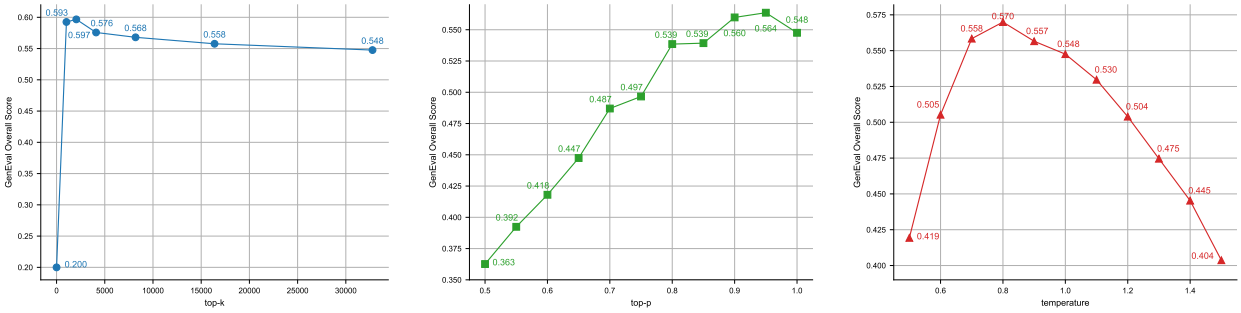
Input: text, images, videos
Output: text, images, videos
Initialize token_sequence = multimodal_tokenize(texts, images, videos)
Set prompt_length = `len`(token_sequence)
while True **do**
    next_token_prob ← model.predict_next(token_sequence)
    next_token_id ← `sample`(next_token_prob)
    Append next_token_id to token_sequence
    if next_token_id = END_OF_SEQUENCE **then**
        break
    end if
end while
result_sequence ← token_sequence[prompt_length:]
result_sequence ← multimodal_detokenize(result_sequence)
for segment in result_sequence **do**
    if segment is text **then**
        display_text(segment)
    else if segment is image **then**
        display_image(segment)
    else if segment is video **then**
        display_video(segment)
    end if
end for

The algorithm begins by tokenizing the inputs into a unified multimodal token sequence, allowing the model to handle diverse modalities within a consistent framework. During the iterative generation loop, the model predicts the next token probabilities based on the current sequence context. The most probable token is then selected and appended to the sequence. This process continues until the end-of-sequence token is encountered, signaling the completion of the generation. The resulting sequence, excluding the initial prompt, is then converted back to its original modalities through detokenization. Finally, the generated modalities are displayed appropriately based on their types (*e.g.*, textual outputs are rendered as text, while visual outputs are displayed

as images or videos). This flexible, token-based approach ensures seamless integration and generation across different modalities, highlighting the model's ability to unify diverse data types under a single, autoregressive framework.

5.3 Token Sampling Strategy

For text tokens, we can employ greedy search or sampling methods such as top-k and top-p. For visual token sampling, we utilize classifier-free guidance in conjunction with top-k sampling. We investigate the impact of different sampling strategies on generation quality for the text-to-image task. Unless otherwise specified, we use classifier-free guidance (CFG) scale of 5.0, with default settings of top-k=32768, top-p=1.0, and temperature=1.0. As shown in Fig. S17a, the performance improves significantly when increasing top-k from 1 to 2048, reaching a peak GenEval overall score of 0.597. However, as top-k continues to increase, the score gradually declines. This suggests that a moderate top-k is crucial for high-quality generation. Besides, we observe a similar trend when changing the parameter of top-p and temperature as in Fig. S17b and Fig. S17c.



(a) top-k sampling (b) top-p sampling (c) temperature sampling

Figure S17: **Impact of token sampling parameters.** We show the GenEval overall score of **Emu3** over different sampling configurations. The default setting is cfg=5.0, top-k=32768, top-p=1.0 and temperature=1.0.

6 Related Work

Vision-Language Understanding. CLIP [49] learns generalizable vision representations through contrastive learning on massive image-text pairs, achieving impressive zero-shot results in image classification tasks. Flamingo [103], by connecting pretrained language models and vision encoders akin to CLIP, initially showcases promising few-shot multimodal understanding capabilities. The increasing availability and progress of LLMs have popularized the fusion of pretrained vision encoders with LLMs, forming a common approach to train extensive vision-language models (VLMs). The BLIP series [104, 105], MiniGPT4 [106], and LLaVA [107] exhibit encouraging results by linking vision encoders with LLMs and training on image-text pairs and vision instruction tuning data. Further improvements in performance are seen in LLaVA series [11, 82] and other impressive works [78, 108] through curated datasets and improved training strategies. While models like Fuyu [85] and EVE [10] introduce encoder-free vision-language architectures that

feed image patches into LLMs, they still face challenges in competing with state-of-the-art VLMs. Here, we show that **Emu3**, a decoder-only model trained purely with next-token prediction, can reach the performance of these encoder-based systems.

Vision Generation. Recent advancements in vision generation have been largely dominated by diffusion models [36, 37, 38, 109, 40]. These models demonstrate impressive capabilities in generating high-resolution images via the diffusion process. The open-source release of the Stable Diffusion series has led to widespread research and development in this direction. Another research line is to train autoregressive models to generate images via predicting the next token in a sequence, such as DALL-E [110], CogView [111], and Parti [112]. VideoGPT [113] and VideoPoet [114] also leverage autoregressive approaches in the video domain. However, they either fail to match the performance with diffusion models or rely on cascade/compositional approaches, *e.g.*, VideoPoet uses a two-stage generate-and-refine framework and an extra text encoder. In this work, **Emu3** demonstrates powerful image and video generation capabilities with a single Transformer decoder. Notably, we open-source to support further research and development in this direction.

Unified Understanding and Generation. There have been early efforts to unify vision understanding and generation [43, 115, 116, 117], exploring various generative objectives on image and text data. Emu and Emu2 [43, 30] introduce a unified autoregressive objective: predicting the next multimodal element, by regressing visual embeddings or classifying textual tokens. CM3Leon [115] and Chameleon [46] trained token-based autoregressive models on mixed image and text data. Some works [118, 119, 120, 121, 122, 123] also explore this direction, but either with a focus on traditional vision tasks such as segmentation or barely close to the performance of specialized models across general multimodal tasks of video generation, image generation, and vision language understanding. More recent methods like TransFusion [45] and Show-o [44] attempt to combine diffusion and autoregressive approaches to boost performance. However, these models still fall behind task-specific architectures like SDXL [38] and LLaVA-1.6 [82] in terms of vision generation and understanding. **Emu3** for the first time demonstrates that next-token prediction across images, video, action, and text can match these well-established models, without relying on compositional methods. This work shows the scalability, effectiveness, and generality of next-token prediction for unified multimodal learning across AI-Generated Content (AIGC), multimodal understanding, and robotic manipulation.

Vision-Language-Action Models. Vision-Language-Action (VLA) models have recently attracted growing attention and have shown great promise across a diverse range of robotics control tasks [124, 125, 126, 127, 128, 129, 130, 131, 132, 133]. These models fine-tune pre-trained vision-language models (VLMs) for action prediction, leveraging their general understanding and adapting them to low-level control via large-scale robotics datasets [134, 135, 136, 125, 137, 138]. Previous VLA methods, such as RT-2 [124] and OpenVLA [126], generally convert continuous actions into sequences of discrete action tokens, and optimize the model through standard autoregressive next-token prediction. While this strategy is straightforward, it can result in an excessive number of tokens under high-frequency control and high-DoF systems, significantly reducing inference efficiency. Therefore, subsequent research methods have explored alternative action decoding mechanisms. For instance, FAST [139] employs frequency-domain compression encoding, while others

adopt diffusion policy prediction [140, 141, 142, 130, 143, 144, 145] for action chunk learning. However, all the aforementioned VLA methods are language-centric, aligning visual signals to the semantic space while neglecting the importance of vision in temporal dynamic information. In contrast, while a recent approach UniVLA [146] explores native VLM-based solutions with world models to facilitate downstream policy learning, our method achieves unified discrete encoding of vision, language, and actions via direct multimodal learning, obviating video post-training.

7 Limitations

Despite the promising results, our approach has several notable limitations. First, the inference can be accelerated. The current inference process employs a naive decoding strategy, while more advanced parallel decoding strategies can be leveraged to speed up. Second, the current tokenizer design presents trade-offs in both compression ratio and reconstruction fidelity, which can be further optimized for efficiency and effectiveness in downstream tasks. Third, the diversity and quality of multimodal datasets, particularly for long-horizon video-centric scenarios, remain insufficient to capture the full range of real-world complexity. While we acknowledge these challenges, addressing them lies beyond the scope of this work. We also highlight several underexplored technical directions for future research, including the development of efficient architectures for ultra-long multimodal contexts, enhancing tokenizer expressiveness, and constructing more robust and realistic benchmarks.

8 Open Weights, Code, and Data

To foster the advancement of the open-source community, we have made the model weights publicly available on HuggingFace at `https://huggingface.co/collections/BAAI/emu3`, including the Emu3 pre-trained model as well as two post-trained derivatives: the image generation model Emu3-Gen and the vision-language understanding model Emu3-Chat. The custom computer code used and the models produced in this study have been made publicly available on GitHub at `https://github.com/baaivision/Emu3`. The code is released under the Apache-2.0 license.

The training data is not fully publicly available yet due to privacy and legal restrictions. The data are available for peer review if required by the editorial process. Requests for access can be directed to the corresponding author. Specifically, the publicly available of our training data, including: FineWeb-Edu (`https://huggingface.co/datasets/HuggingFaceFW/fineweb-edu`), LAION-5b (`https://laion.ai/blog/laion-5b/`), LAION-AESTHETICS (`https://laion.ai/blog/laion-aesthetics/`), Datacomp (`https://github.com/mlfoundations/datacomp`), COYO-700m(`https://github.com/kakaobrain/coyo-dataset`), OpenImages (`https://storage.googleapis.com/openimages/web/index.html`), SA-1B (`https://segment-anything.com/dataset/index.html`), YT-Temporal-1B (`https://rowanzellers.com/merlotreserve/`), JourneyDB(`https://huggingface.co/datasets/JourneyDB/JourneyDB`), DiffusionDB(`https://huggingface.co/datasets/poloclub/diffusiondb`), midjourney-prompts(`https://huggingface.co/datasets/vivym/midjourney-p`

rompts), LLaVA-OneVision-Data(https://huggingface.co/datasets/lmms-lab/LLaVA-OneVision-Data).

We used the following publicly available databases for evaluation, including COCO (https://cocodataset.org/#download), GenEval (https://github.com/djghosh13/geneval), T2I-CompBench (https://karine-h.github.io/T2I-CompBench/), DPG-Bench (https://github.com/TencentQQGYLab/ELLA/tree/main/dpg_bench), VBench (https://github.com/Vchitect/VBench/tree/master/prompts), SEEDBench (https://huggingface.co/datasets/AILab-CVC/SEED-Bench), OCRBench (https://github.com/Yuliang-Liu/MultimodalOCR#data), MMVet (https://huggingface.co/datasets/whyu/mm-vet), POPE (https://github.com/RUCAIBox/POPE), VQAv2 (https://visualqa.org/download.html), GQA (https://cs.stanford.edu/people/dorarad/gqa/download.html), ScienceQA (https://scienceqa.github.io/#download), TextVQA (https://textvqa.org/dataset/), ChartQA (https://huggingface.co/datasets/ahmed-masry/ChartQA), DocVQA (https://rrc.cvc.uab.es/?ch=17&com=downloads), InfoVQA (https://rrc.cvc.uab.es/?ch=17&com=downloads), AI2D (https://allenai.org/data/diagrams), RealWorldQA (https://huggingface.co/datasets/visheratin/realworldqa), MMMU (https://huggingface.co/datasets/MMMU/MMMU), MMBench (https://huggingface.co/datasets/lmms-lab/MMBench), GenEval(https://github.com/djghosh13/geneval), DPG-Bench(https://github.com/TencentQQGYLab/ELLA), T2I-CompBench(https://github.com/Karine-Huang/T2I-CompBench), VBench(https://github.com/Vchitect/VBench), CALVIN(https://github.com/mees/calvin).

9 Author List

Project Lead

Xinlong Wang

Core Contributors

Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Zhen Li
(Equal Contribution)

Contributors

Yuqi Wang, Qiying Yu, Yingli Zhao, Yulong Ao, Xuebin Min, Chunlei Men, Boya Wu, Bo Zhao, Bowen Zhang, Liangdong Wang, Guang Liu, Zheqi He, Xi Yang, Jingjing Liu, Yonghua Lin, Zhongyuan Wang, Tiejun Huang

References

[1] Razzhigaev, A. *et al.* Kandinsky: An improved text-to-image synthesis with image prior and latent diffusion. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023 - System Demonstrations* (2023).

[2] Russakovsky, O. *et al.* Imagenet large scale visual recognition challenge. *International journal of computer vision* **115**, 211–252 (2015).

[3] Soomro, K., Zamir, A. R. & Shah, M. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402* (2012).

[4] Chung, H. W. *et al.* Scaling instruction-finetuned language models. *Journal of Machine Learning Research* **25**, 1–53 (2024).

[5] Peebles, W. & Xie, S. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 4195–4205 (2023).

[6] Chen, J. *et al.* Pixart-alpha: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426* (2023).

[7] Yang, A. *et al.* Qwen2.5 technical report (2025). URL `https://arxiv.org/abs/2412.15115`. `2412.15115`.

[8] Kuznetsova, A. *et al.* The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International journal of computer vision* **128**, 1956–1981 (2020).

[9] Podell, D. *et al.* Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952* (2023).

[10] Diao, H. *et al.* Unveiling encoder-free vision-language models. *arXiv preprint arXiv:2406.11832* (2024).

[11] Liu, H., Li, C., Li, Y. & Lee, Y. J. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 26296–26306 (2024).

[12] Wolf, T. *et al.* Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45 (Association for Computational Linguistics, Online, 2020). URL `https://www.aclweb.org/anthology/2020.emnlp-demos.6`.

[13] Ghosh, D., Hajishirzi, H. & Schmidt, L. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems* **36** (2024).

[14] Zhang, B.-W. *et al.* Aquila2 technical report. *arXiv preprint arXiv:2408.07410* (2024).

[15] Lozhkov, A., Ben Allal, L., von Werra, L. & Wolf, T. Fineweb-edu: the finest collection of educational content (2024). URL https://huggingface.co/datasets/HuggingFaceFW/fineweb-edu.

[16] Schuhmann, C. *et al.* Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems* **35**, 25278–25294 (2022).

[17] Sun, Q., Fang, Y., Wu, L., Wang, X. & Cao, Y. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389* (2023).

[18] LAION. Laion-aesthetics. https://laion.ai/blog/laion-aesthetics/ (2022).

[19] Byeon, M. *et al.* Coyo-700m: Image-text pair dataset. https://github.com/kakaobrain/coyo-dataset (2022).

[20] Gadre, S. Y. *et al.* Datacomp: In search of the next generation of multimodal datasets. *Advances in Neural Information Processing Systems* **36**, 27092–27112 (2023).

[21] Kirillov, A. *et al.* Segment anything. *arXiv:2304.02643* (2023).

[22] Pan, J. *et al.* Journeydb: A benchmark for generative image understanding (2023). 2307.00716.

[23] Wang, Z. J. *et al.* Diffusiondb: A large-scale prompt gallery dataset for text-to-image generative models. *arXiv:2210.14896 [cs]* (2022). URL https://arxiv.org/abs/2210.14896.

[24] Miech, A. *et al.* Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2630–2640 (2019).

[25] Zellers, R. *et al.* Merlot reserve: Neural script knowledge through vision and language and sound. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 16375–16387 (2022).

[26] Blattmann, A. *et al.* Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127* (2023).

[27] Cui, C. *et al.* Paddleocr 3.0 technical report (2025). URL https://arxiv.org/abs/2507.05595. 2507.05595.

[28] Teed, Z. & Deng, J. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, 402–419 (Springer, 2020).

[29] Li, X. *et al.* Densefusion-1m: Merging vision experts for comprehensive multimodal perception. *arXiv preprint arXiv:2407.08303* (2024).

[30] Sun, Q. *et al.* Generative multimodal models are in-context learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14398–14409 (2024).

[31] OpenAI. Chatgpt. `https://chat.openai.com/` (2023).

[32] Kwon, W. *et al.* Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles* (2023).

[33] Chen, X. *et al.* Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325* (2015).

[34] Huang, K., Sun, K., Xie, E., Li, Z. & Liu, X. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. *Advances in Neural Information Processing Systems* **36**, 78723–78747 (2023).

[35] Hu, X. *et al.* Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135* (2024).

[36] Rombach, R., Blattmann, A., Lorenz, D., Esser, P. & Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2022).

[37] Ramesh, A., Dhariwal, P., Nichol, A., Chu, C. & Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* (2022).

[38] Podell, D. *et al.* Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952* (2023).

[39] Chen, J. *et al.* Pixart-alpha: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426* (2023).

[40] Betker, J. *et al.* Improving image generation with better captions. `https://cdn.openai.com/papers/dall-e-3.pdf` (2023).

[41] Esser, P. *et al.* Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning* (2024).

[42] BlackForest. Flux. `https://github.com/black-forest-labs/flux` (2024).

[43] Sun, Q. *et al.* Emu: Generative pretraining in multimodality. In *The Twelfth International Conference on Learning Representations* (2023).

[44] Xie, J. *et al.* Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528* (2024).

[45] Zhou, C. *et al.* Transfusion: Predict the next token and diffuse images with one multi-modal model. *arXiv preprint arXiv:2408.11039* (2024).

[46] Team, C. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818* (2024).

[47] Sun, P. *et al.* Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525* (2024).

[48] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B. & Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems* **30** (2017).

[49] Radford, A. *et al.* Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763 (PMLR, 2021).

[50] Ho, J. & Salimans, T. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598* (2022).

[51] Liu, B. *et al.* Playground v3: Improving text-to-image alignment with deep-fusion large language models. *arXiv preprint arXiv:2409.10695* (2024).

[52] Feng, Y. *et al.* Ranni: Taming text-to-image diffusion for accurate instruction following. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4744–4753 (2024).

[53] Blattmann, A. *et al.* Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127* (2023).

[54] Zhang, R., Isola, P., Efros, A. A., Shechtman, E. & Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. *arXiv preprint arXiv:1801.03924* (2018).

[55] Kingma, D. P. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[56] Rezende, D. J., Mohamed, S. & Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, 1278–1286 (PMLR, 2014).

[57] Zhang, R. Making convolutional networks shift-invariant again. In *International conference on machine learning*, 7324–7334 (PMLR, 2019).

[58] Shi, W. *et al.* Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1874–1883 (2016).

[59] Huang, Z. *et al.* Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21807–21818 (2024).

[60] Wang, J. *et al.* Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571* (2023).

[61] Wang, Y. *et al.* Lavie: High-quality video generation with cascaded latent diffusion models. *arXiv preprint arXiv:2309.15103* (2023).

[62] Lab, P.-Y. & etc., T. A. Open-sora-plan (2024).

[63] Zhang, D. J. *et al.* Show-1: Marrying pixel and latent diffusion models for text-to-video generation. *arXiv preprint arXiv:2309.15818* (2023).

[64] Zheng, Z. *et al.* Open-sora: Democratizing efficient video production for all (2024).

[65] Guo, Y. *et al.* Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725* (2023).

[66] Runway. Gen-2: Generate novel videos with text, images or video clips. https://runwayml.com/research/gen-2/ (2023).

[67] Labs, P. Pika. https://pika.art/home/ (2023).

[68] Chen, H. *et al.* Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7310–7320 (2024).

[69] Li, J. *et al.* T2v-turbo: Breaking the quality bottleneck of video consistency model with mixed reward feedback. *arXiv preprint arXiv:2405.18750* (2024).

[70] Yang, Z. *et al.* Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072* (2024).

[71] Kuaishou. Kling ai. https://klingai.com/ (2024).

[72] Runway. Gen-3 alpha: A new frontier for video generation. https://runwayml.com/research/introducing-gen-3-alpha/ (2024).

[73] Kong, W. *et al.* Hunyuanvideo: A systematic framework for large video generative models (2025). URL https://arxiv.org/abs/2412.03603. 2412.03603.

[74] Wan, T. *et al.* Wan: Open and advanced large-scale video generative models (2025). URL https://arxiv.org/abs/2503.20314. 2503.20314.

[75] Zhang, K. *et al.* Lmms-eval: Reality check on the evaluation of large multimodal models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, 881–916 (2025).

[76] Dai, W. *et al.* Instructblip: Towards general-purpose vision-language models with instruction tuning. In *Advances in Neural Information Processing Systems*, vol. 36, 49250–49267 (2023).

[77] IDEFICS Research Team. Introducing idefics: An open reproduction of state-of-the-art visual language model. https://huggingface.co/blog/idefics (2023).

[78] Bai, J. *et al.* Qwen technical report. *arXiv preprint arXiv:2309.16609* (2023).

[79] Chen, Z. *et al.* How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821* (2024).

[80] Ye, Q. *et al.* mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13040–13051 (2024).

[81] Chen, L. *et al.* Sharegpt4v: Improving large multi-modal models with better captions. *arXiv preprint arXiv:2311.12793* (2023).

[82] Liu, H. *et al.* Llava-next: Improved reasoning, ocr, and world knowledge. `https://llava-vl.github.io/blog/2024-01-30-llava-next/` (2024).

[83] Lin, J. *et al.* Vila: On pre-training for visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 26689–26699 (2024).

[84] Bai, S. *et al.* Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923* (2025).

[85] Bavishi, R. *et al.* Introducing our multimodal models. `https://www.adept.ai/blog/fuyu-8b` (2023).

[86] Li, B. *et al.* Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125* (2023).

[87] Liu, Y. *et al.* On the hidden mystery of ocr in large multimodal models. *arXiv preprint arXiv:2305.07895* (2023).

[88] Yu, W. *et al.* Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490* (2023).

[89] Li, Y. *et al.* Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355* (2023).

[90] Goyal, Y., Khot, T., Summers-Stay, D., Batra, D. & Parikh, D. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6904–6913 (2017).

[91] Hudson, D. A. & Manning, C. D. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6700–6709 (2019).

[92] Lu, P. *et al.* Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems* **35**, 2507–2521 (2022).

[93] Singh, A. *et al.* Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8317–8326 (2019).

[94] Masry, A., Long, D. X., Tan, J. Q., Joty, S. & Hoque, E. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244* (2022).

[95] Mathew, M., Karatzas, D. & Jawahar, C. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2200–2209 (2021).

[96] Mathew, M. *et al.* Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 1697–1706 (2022).

[97] Kembhavi, A. *et al.* A diagram is worth a dozen images. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, 235–251 (Springer, 2016).

[98] XAI. Realworldqa (2024).

[99] Yue, X. *et al.* Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of CVPR* (2024).

[100] Liu, Y. *et al.* Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281* (2023).

[101] Contributors, F. Flagscale: A unified meta-framework enabling adaptive heterogeneous computing for the llm ecosystem. `https://github.com/FlagOpen/FlagScale` (2024). Accessed: 2025-06-26.

[102] Dao, T. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)* (2024).

[103] Alayrac, J.-B. *et al.* Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems* **35**, 23716–23736 (2022).

[104] Li, J., Li, D., Xiong, C. & Hoi, S. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, 12888–12900 (PMLR, 2022).

[105] Li, J., Li, D., Savarese, S. & Hoi, S. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, 19730–19742 (PMLR, 2023).

[106] Zhu, D., Chen, J., Shen, X., Li, X. & Elhoseiny, M. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592* (2023).

[107] Liu, H., Li, C., Wu, Q. & Lee, Y. J. Visual instruction tuning. *Advances in neural information processing systems* **36** (2024).

[108] Chen, Z. *et al.* Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 24185–24198 (2024).

[109] Peebles, W. & Xie, S. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748* (2022).

[110] Ramesh, A. *et al.* Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092* (2021).

[111] Ding, M. *et al.* Cogview: Mastering text-to-image generation via transformers. *arXiv preprint arXiv:2105.13290* (2021).

[112] Yu, J. *et al.* Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789* **2**, 5 (2022).

[113] Yan, W., Zhang, Y., Abbeel, P. & Srinivas, A. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157* (2021).

[114] Kondratyuk, D. *et al.* Videopoet: A large language model for zero-shot video generation. *arXiv preprint arXiv:2312.14125* (2023).

[115] Yu, L. *et al.* Scaling autoregressive multi-modal models: Pretraining and instruction tuning. *arXiv preprint arXiv:2309.02591* **2** (2023).

[116] Ge, Y. *et al.* Seed-x: Multimodal models with unified multi-granularity comprehension and generation. *arXiv preprint arXiv:2404.14396* (2024).

[117] Dong, R. *et al.* Dreamllm: Synergistic multimodal comprehension and creation. *arXiv preprint arXiv:2309.11499* (2023).

[118] Bachmann, R. *et al.* 4m-21: An any-to-any vision model for tens of tasks and modalities. In *Advances in neural information processing systems* (2024).

[119] Wang, H. *et al.* Git: Towards generalist vision transformer through universal language interface. In *ECCV* (2024).

[120] Liu, H., Yan, W., Zaharia, M. & Abbeel, P. World model on million-length video and language with blockwise ringattention. In *The Thirteenth International Conference on Learning Representations* (2025).

[121] Li, H. *et al.* Synergen-vl: Towards synergistic image understanding and generation with vision experts and token folding. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, 29767–29779 (2025).

[122] Wu, Y. *et al.* Vila-u: a unified foundation model integrating visual understanding and generation. *arXiv preprint arXiv:2409.04429* (2024).

[123] Wu, J. *et al.* Liquid: Language models are scalable multi-modal generators. *arXiv preprint arXiv:2412.04332* (2024).

[124] Brohan, A. *et al.* Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818* (2023).

[125] Vuong, Q. *et al.* Open x-embodiment: Robotic learning datasets and rt-x models. In *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition@ CoRL2023* (2023).

[126] Kim, M. J. *et al.* Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246* (2024).

[127] Zhen, H. *et al.* 3d-vla: A 3d vision-language-action generative world model. In *International Conference on Machine Learning*, 61229–61245 (PMLR, 2024).

[128] Cheang, C.-L. *et al.* Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158* (2024).

[129] Li, Q. *et al.* Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650* (2024).

[130] Black, K. *et al.* π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164* (2024).

[131] Bjorck, J. *et al.* Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734* (2025).

[132] Kim, M. J., Finn, C. & Liang, P. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645* (2025).

[133] Huang, H. *et al.* Otter: A vision-language-action model with text-aware visual feature extraction. *arXiv preprint arXiv:2503.03734* (2025).

[134] Dasari, S. *et al.* Robonet: Large-scale multi-robot learning. *arXiv preprint arXiv:1910.11215* (2019).

[135] Ebert, F. *et al.* Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv preprint arXiv:2109.13396* (2021).

[136] Fang, H.-S. *et al.* Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot. *arXiv preprint arXiv:2307.00595* (2023).

[137] Khazatsky, A. *et al.* Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945* (2024).

[138] Bu, Q. *et al.* Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669* (2025).

[139] Pertsch, K. *et al.* Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747* (2025).

[140] Chi, C. *et al.* Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research* 02783649241273668 (2023).

[141] Ke, T.-W., Gkanatsios, N. & Fragkiadaki, K. 3d diffuser actor: Policy diffusion with 3d scene representations. *arXiv preprint arXiv:2402.10885* (2024).

[142] Liu, S. *et al.* Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864* (2024).

[143] Xue, H. *et al.* Reactive diffusion policy: Slow-fast visual-tactile policy learning for contact-rich manipulation. *arXiv preprint arXiv:2503.02881* (2025).

[144] Wen, J. *et al.* Dexvla: Vision-language model with plug-in diffusion expert for general robot control. *arXiv preprint arXiv:2502.05855* (2025).

[145] Intelligence, P. *et al.* \pi_ {0.5} : a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054* (2025).

[146] Wang, Y. *et al.* Unified vision-language-action model. *arXiv preprint arXiv:2506.19850* (2025).