Kandinsky 5.0: A Family of Foundation Models for Image and Video Generation

Kandinsky Lab*

* A detailed list of the contributors can be found in the end of this paper.

Abstract: This report introduces **Kandinsky 5.0**, a family of state-of-the-art foundation models for high-resolution image and 10-second video synthesis. The framework comprises three core line-up of models: **Kandinsky 5.0 Image Lite** – a line-up of 6B parameter image generation models, **Kandinsky 5.0 Video Lite** – a fast and lightweight 2B parameter text-to-video and image-to-video models, and **Kandinsky 5.0 Video Pro** – 19B parameter models that achieves superior video generation quality. We provide a comprehensive review of the data curation lifecycle – including collection, processing, filtering and clustering – for the multi-stage training pipeline that involves extensive pre-training and incorporates quality-enhancement techniques such as self-supervised fine-tuning (SFT) and reinforcement learning (RL)-based post-training. We also present novel architectural, training, and inference optimizations that enable Kandinsky 5.0 to achieve high generation speeds and state-of-the-art performance across various tasks, as demonstrated by human evaluation. As a large-scale, publicly available generative framework, Kandinsky 5.0 leverages the full potential of its pre-training and subsequent stages to be adapted for a wide range of generative applications. We hope that this report, together with the release of our open-source code and training checkpoints, will substantially advance the development and accessibility of high-quality generative models for the research community.

Date: November 20, 2025
Website: https://kandinskylab.ai/
GitHub: https://github.com/kandinskylab/kandinsky-5
Hugging Face: https://huggingface.co/kandinskylab

Contents

1 Introduction

Over the past few years, diffusion models [1, 2] and the subsequent flow matching approaches [3] have led to a qualitative breakthrough in image generation, achieving unprecedented synthesis quality and diversity. This foundation enabled the rapid development of commercial and open-source systems that provide users with a wide range of generative capabilities, from text-to-image (T2I) synthesis to complex editing [4, 5, 6, 7]. To date, image generation models have not only achieved high quality but continue to improve actively, constantly raising the bar for realism and controllability, as demonstrated by models such as Stable Diffusion 3 [8], Flux [9], Seedream 3 & 4 [10, 11], and Hunyuan Image 3 [12].

A natural extension of this progress has been the growing interest in video generation, leading to numerous methods that adapt and extend architectures proven successful for images [13, 14, 15, 16]. However, the direct translation of these approaches faced fundamental scalability issues due to the exponential growth in computational complexity when working with time-dependent three-dimensional video data. Partial resolution of these limitations was achieved through the active adoption of architectures like the Diffusion Transformer (DiT) [17], which provided the necessary scalability and efficiency [18, 19], along with a series of modifications to attention mechanisms aimed at handling video data [20, 21, 22].

Today, a number of video generation models demonstrate a high level of quality, such as Sora [23, 24] and Veo [25]. A significant part of this progress is driven by open-source initiatives. Projects such as HunyuanVideo [26], Mochi [27], CogVideoX [28], Wan [29] and VACE [30], democratize access to foundational architectures and pre-trained weights, accelerating research and development, and demonstrating results close to professional-grade video production. All this opens up broad opportunities for the application of video models and lays the groundwork for creating multimedia generation systems [31], "world models" [32], and foundational visual models, analogous in their significance to Large Language Models (LLMs) in Natural Language Processing (NLP) [33, 34, 35].

Despite the rapid development, critical challenges persist in video generation. Beyond processing vast amounts of data, creating such systems requires complex, multi-stage optimizations for both the training process and subsequent inference. Therefore, the efficient creation of high-quality, consistent, and controllable video remains one of the most challenging tasks in generative AI.

In this work, we aim to address some of the key challenges in the field of video generation. We present **Kandinsky 5.0** – a family of fundamental generative models for high-resolution image and video synthesis, designed to achieve state-of-the-art quality and operational efficiency. The Kandinsky 5.0 suite comprises three line-up of models:

- **Kandinsky 5.0 Video Pro:** A high-power 19B parameter models for text-to-video and image-to-video generation, creating up to 10-second videos at high resolution .
- **Kandinsky 5.0 Video Lite:** A lightweight 2B parameter model for text-to-video and image-to-video generation, producing up to 10-second clips.
- **Kandinsky 5.0 Image Lite:** A 6B parameter models for text-to-image generation and image editing at high resolution.

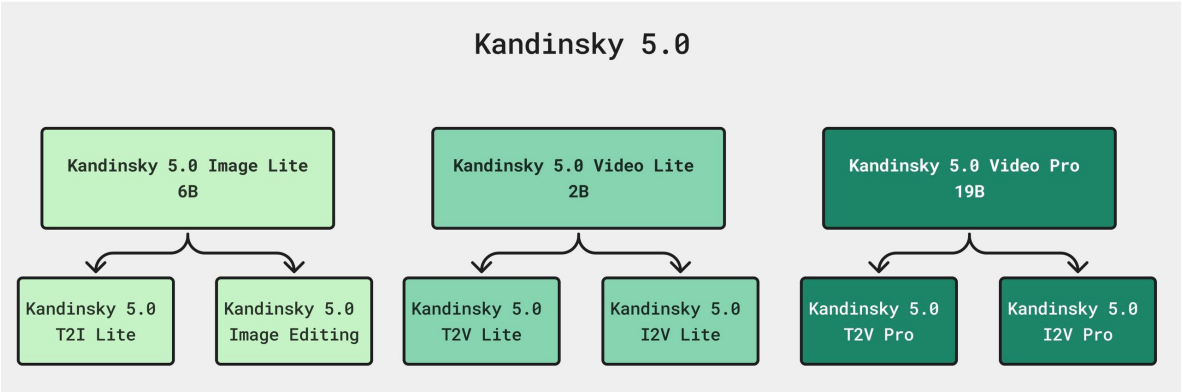


Figure 1: Kandinsky 5.0 Models Family

The key contributions of this technical report are as follows:

1. We provide a comprehensive description of the data collection and processing pipeline, including data preparation for instructive image editing tuning and self-supervised fine-tuning (SFT) for both video and image modalities.
2. We detail the multi-stage training pipeline for all six models, encompassing a pretraining phase for learning general patterns of the visual world and SFT for enhancing visual quality. We also introduce our RLHF post-training adversarial method based on comparing generated images with those from the SFT dataset. This approach achieves superior realism, visual quality, and prompt alignment.
3. We present the architecture of our core CrossDiT model, featuring our key optimization of the attention mechanism for high-resolution video (exceeding 512 px) with durations longer than 5 seconds via the NABLA method [36]. This overcomes the quadratic complexity of standard spatio-temporal attention, achieving a $2.7\times$ reduction in training and inference time with 90% sparsity ratio while maintaining generated video quality, as confirmed by FVD [37], VBench [38], CLIP-score [39] and human evaluation through side-by-side testing.
4. We describe multiple optimizations implemented across the pipeline to accelerate inference, training, and reduce memory consumption. These techniques include variational autoencoder (VAE) optimization, text encoder quantization, and CrossDiT training optimizations using Fully or Hybrid Sharded Data Parallel (F/HSDP) [40], activation checkpointing [41], among others.
5. For video model distillation, we employ a combined approach that integrates Classifier-Free Guidance Distillation [42], Trajectory Segmented Consistency Distillation (TSCD) [43], and subsequent adversarial post-training [44] to enhance visual quality. This reduces the Number of Function Evaluations (NFE) from 100 to 16 while preserving visual quality, as evidenced by side-by-side human evaluation results.
6. We evaluate our final models against several state-of-the-art approaches and demonstrate superior video generation quality through human evaluation on a prompt set from MovieGen [45].
7. Finally, we open-source the code and weights for all models from various training stages, and provide access through the `diffusers` library.

2 Report Overview

The report is structured to provide a comprehensive understanding of the model's design, training, and evaluation:

- **Section 3: Background: The Evolution of Kandinsky models.** Traces the history of the Kandinsky model family, from early autoregression-based models to the current latest version of Kandinsky 5.0.
- **Section 4: Data Processing Pipeline.** Describes the large-scale, multi-stage pipeline used for curating and annotating datasets for text-to-image and text-to-video pretraining, self-supervised fine-tuning, image instruction tuning, and the collection of Russian-specific multicultural data. We emphasize quality control and scalability in our approach.
- **Section 5: Kandinsky 5.0 Architecture.** Presents the architecture of the Kandinsky 5.0 models, which is common for all models in this family. The core components include a Cross-Attention Diffusion Transformer (**CrossDiT**), a corresponding **CrossDiT-block** scheme, and the Neighborhood Adaptive Block-Level Attention (**NABLA**) mechanism, which is essential for optimizing both training and inference.
- **Section 6: Training Stages.** Outlines the multi-phase training process, from pre-training on large-scale datasets to self-supervised fine-tuning, distillation, and RL-based post-training, tailored for both image and video models.
- **Section 7: Optimizations.** Covers techniques such as VAE encoder acceleration, CrossDiT training optimization and efficient use of GPU memory.
- **Section 8: Results.** Presents a growth of visual quality at different training stages and human side-by-side (SBS) evaluations, demonstrating superior performance in motion consistency, visual quality, and prompt alignment compared to existing models.
- **Section 9: Use Cases.** Highlights practical applications across text-to-image, image editing, text-to-video, and image-to-video generation, supported by visual examples and technical prompts.
- **Section 10: Related Work.** Contextualizes Kandinsky 5.0 within the broader landscape of generative models, covering advancements in text-to-image and text-to-video generation, distillation, post-training techniques, and evaluation methodologies for generative models.
- **Section 11: Limitations and Further Work.** Discusses remaining challenges, that guide future research directions.
- **Section 12: Border Impacts and Ethical Considerations.** Details the responsible AI framework implemented, including data curation, runtime safeguards, and ethical use guidelines to ensure safe deployment.
- **Sections 13–14: Conclusion, Contributors & Acknowledgments.** Summarizes contributions and acknowledges the teams and collaborators involved.

3 Background: The Evolution of Kandinsky models

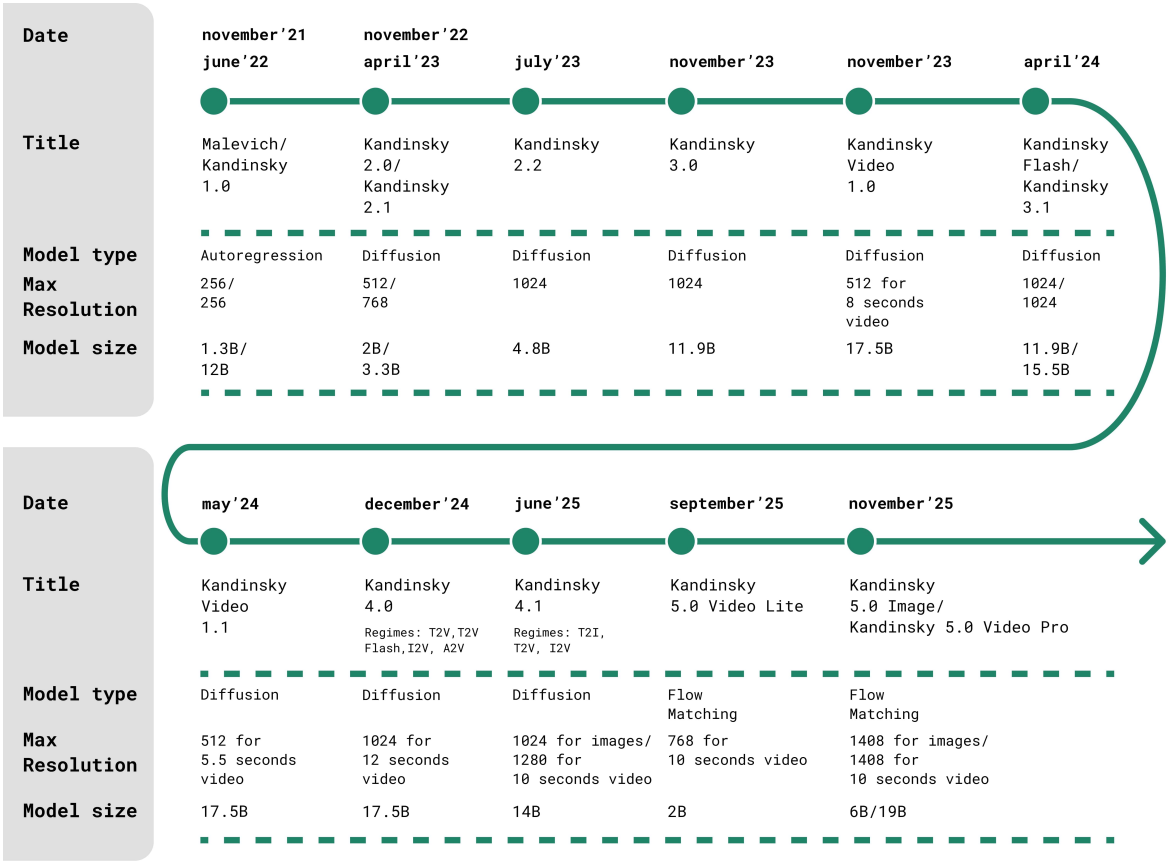


Figure 2: The evolution of Kandinsky models.

The Kandinsky family of visual generative models, named after Russian abstract artist Wassily Kandinsky (1866 – 1944), has evolved significantly since its inception (Figure 2). Its history began in June 2022 with the release of **Kandinsky 1.0**, a 12B-parameter model. This model was an enlarged version of the earlier **Malevich**[1,2] model (aka ruDALL-E XL), which had 1.3B parameters, was inspired by DALL-E [46], and was released in November 2021. Compared to Malevich model, Kandinsky 1.0 featured more layers, increased hidden space, and was trained on 120 million text-image pairs. Both models utilized an autoregressive architecture to generate 256×256 resolution images.

A radical shift occurred in November 2022 with **Kandinsky 2.0**[3,4] [47], the family's first diffusion model. To ensure multilingual capability, it employed two encoders – XLMR-clip [48] and mT5-small [49] – enabling support for queries in 101 languages. Trained on one billion text-image pairs, the 2B-parameter model could generate images at a 512×512 resolution.

The next stage, **Kandinsky 2.1**[5] (April 2023), brought major architectural improvements: a

[1] https://github.com/ai-forever/ru-dalle
[2] https://huggingface.co/ai-forever/rudalle-Malevich
[3] https://github.com/ai-forever/Kandinsky-2
[4] https://huggingface.co/ai-forever/Kandinsky_2.0
[5] https://huggingface.co/ai-forever/Kandinsky_2.1

single XLM-Roberta-Large-Vit-L-14[6] text encoder, a MoVQ image autoencoder [50], and the addition of an image prior model for better text-image alignment. The diffusion mechanism was refined to use CLIP visual embeddings [51]. After fine-tuning on an additional 170 million text-image pairs, the model surpassed DALL-E 2 [52], GigaGAN [53], and Stable Diffusion 2.1 [4] on the FID metric [54] on the COCO 30k dataset [55]. The model has 3.3B parameters, a resolution of 768×768, and natively supports inpainting, outpainting, image blending, synthesis of variations of an input image, and text-guided image editing.

In July 2023, **Kandinsky 2.2**[7] (4.8B) further enhanced photorealism and increased resolution to 1024 pixels with support for various aspect ratios. Other innovations included the introduction of a ControlNet mechanism [56] for local editing, and the capability to generate 4-second animated clips.

November 2023 saw the release of two major models. **Kandinsky 3.0**[8,9] (11.9B) [57], which focused on precise text-image alignment, used the FLAN-UL2 language encoder. Released in parallel, **Kandinsky Video 1.0**[10] (17.5B) [15] was a state-of-the-art, open-source video generation model based on Kandinsky 3.0. It operated in two stages: generating keyframes and then interpolating between them. Generating an 8-second video at 512×512 resolution took approximately three minutes, despite the model being trained on only 220,000 text-video pairs.

The line-up expanded again in April 2024 with two models. **Kandinsky Flash** (11.9B) was a distilled version of Kandinsky 3.0 that reduced the number of generation steps from 50 to 4 using the Adversarial Diffusion Distillation approach [58]. **Kandinsky 3.1** (15.5B) [7] combined Kandinsky 3.0 and Kandinsky Flash, using the latter as a refiner on the final steps in the reverse diffusion process, which significantly enhanced visual quality. This version also incorporated improvements to multicultural awareness, particularly for the Russian-culture domain [59, 60]. The model supports image variation, image blending, image-and-text blending, ControlNet-based editing, and inpainting. Additionally, we trained a state-of-the-art super-resolution model **KandiSuperRes**[11] on the base Kandinsky 3.0 model.

A video model update, **Kandinsky Video 1.1** [16], followed in May 2024. It used Kandinsky 3.0 for first frame generation and was trained on an enlarged dataset of 4.6 million text-video pairs. The model generates 5.5-second videos, and its dataset preparation leveraged automatic captioning via LLaVA-1.5 [61].

Kandinsky 4.0[12] (17.5B) was released in December 2024 as the family's first Diffusion Transformer model. Using an MMDiT-like architecture [8], it generates both images and 12-second videos from text and images. In VBench benchmarks [38] and human evaluations, Kandinsky 4.0 demonstrated superior results against models like CogVideoX-1.5 [28], Open-Sora-Plan v1.3 [62], Mochi v1.0 [27], and Pyramid Flow [63]. This version also supported audio generation from an input video clip. Its refinement, **Kandinsky 4.1** (14B, June 2025), adopted a DiT architecture for image generation and underwent a Supervised Fine-Tuning (SFT) stage using data manually selected by a team of expert artists to enhance aesthetics.

The pinnacle of development to date is the **Kandinsky 5.0**[13] model family described in this report. This is the first Kandinsky models based on the Flow Matching [3], comprising six models of different sizes for various high-quality image and video generation tasks.

[6] XLM-Roberta-Large-Vit-L-14
[7] https://huggingface.co/docs/diffusers/api/pipelines/kandinsky_v22
[8] https://github.com/ai-forever/Kandinsky-3
[9] https://huggingface.co/docs/diffusers/api/pipelines/kandinsky3
[10] https://github.com/ai-forever/KandinskyVideo
[11] https://github.com/ai-forever/KandiSuperRes
[12] https://github.com/ai-forever/Kandinsky-4
[13] https://github.com/ai-forever/Kandinsky-5

4 Data Processing Pipeline

The training of the video generation model leverages multiple datasets across different training stages. The primary datasets for pretraining are large Kandinsky T2I and Kandinsky T2V, while tiny Kandinsky SFT dataset is utilized during fine-tuning stages to significantly boost visual quality. The distinct Kandinsky RCC dataset is used to improve culturally specific video generation capabilities. We also invent a comprehensive routine to collect Kandinsky I2I (instruct dataset) allowing us to train a precise image editing model. Large-scale data collection and an efficient training process are impossible without properly installed and configured infrastructure.

4.1 Data Processing Infrastructure

A core part of the data processing pipeline is a database, which performs as a metadata storage and as a task broker for a distributed network of data processors. This database has several indexes (including vector ones) that allow quick retrieval of required data for each training stage and prevent duplicates from appearing in the dataset. Table partitioning and load-balancing are used to reduce the load on the database, allowing thousands of data processors to work with it simultaneously. Data processors run on different hardware and can utilize CPU and GPU resources for different tasks.

4.2 Text-to-Image Dataset Processing

The Kandinsky T2I dataset is a large-scale collection of more than 500 million general-domain images, designed to support the pretraining and fine-tuning stages of the Kandinsky model. These images originate from a diverse set of sources, including prominent open datasets (e.g., LAION [64], COYO[14]) and large online image repositories. A meticulous data processing pipeline ensures the dataset's quality and suitability for its intended use.

Processing Pipeline

The processing pipeline for the Kandinsky T2I dataset comprises the following several sequential and filtering stages:

- **Initial Resolution Filtering:** Images with a shorter side measuring fewer than 256 pixels are discarded. This step ensures a baseline level of visual detail.
- **Deduplication:** To eliminate redundant content, an image perceptual hash [65] is calculated for each image. This technique identifies and removes exact duplicates and images that are visually very similar.
- **Advanced Filtering:** A series of models are applied to assess image quality and content, filtering out undesirable data:
  - *Watermark Detection:* Watermark detection is performed using a combination of two models. The system employs the `watermark_resnext101_32x8d-large` classifier for perceptual analysis alongside a YOLO-based detector that locates watermark-like objects. Images flagged by either model with a confidence score above a defined threshold are filtered out.
  - *Quality Assessment:* Models predict both technical and aesthetic quality to prioritize visually appealing and well-constructed images:

[14] https://github.com/kakaobrain/coyo-dataset

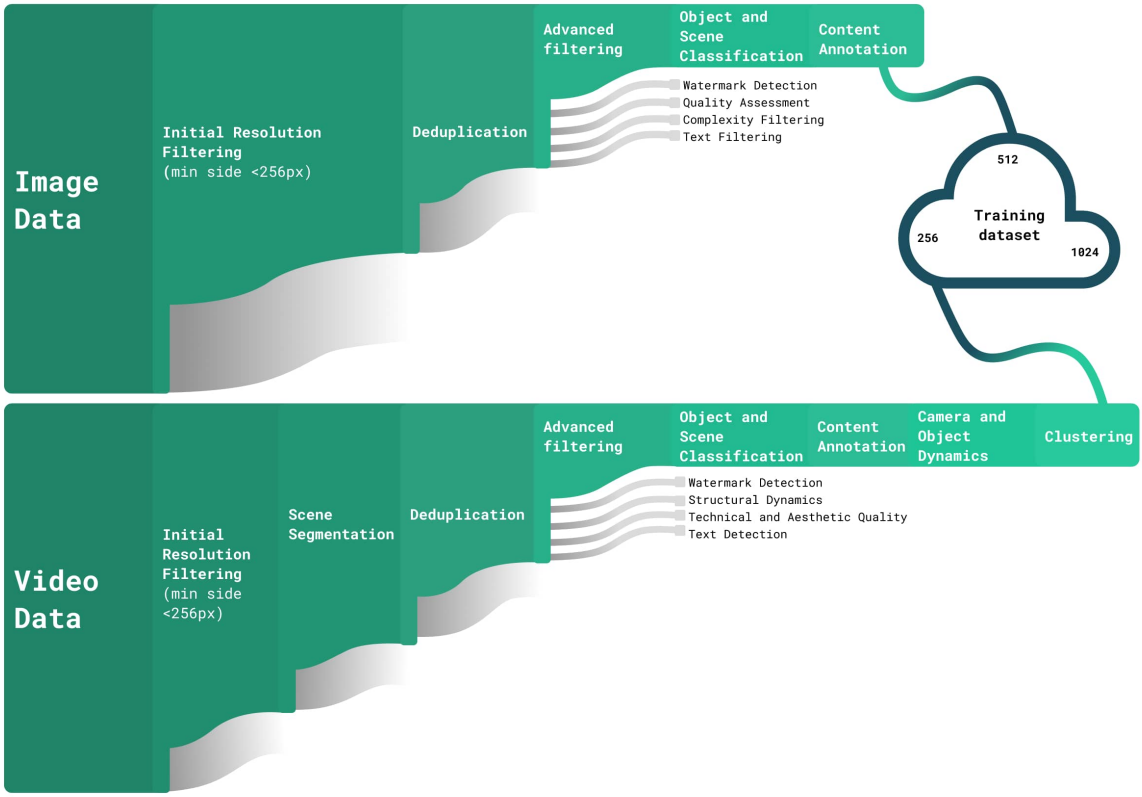


Figure 3: Data processing pipeline for Kandinsky T2V (text-to-video) and Kandinsky T2I (text-to-image) datasets. The workflow begins with raw image and video data, followed by initial filtering and deduplication it processed through advanced filtering (including watermark detection, quality assessment, complexity and text filtering), classification and content annotation stages. Final processed data is stored grouped by resolution (256, 512, and 1024 minimal side lengths) to use in correspondent pretrain stage.

  - The TOPIQ model [66] provides separate scores for technical quality and aesthetic quality.
  - The Q-Align model [67] offers an alternative assessment of technical and aesthetic aspects.
- *Text Filtering:* The CRAFT text detection model [68] identifies regions of text within images. Images containing an excessive amount of text, based on the number of text boxes or the total text area, are excluded to avoid bias towards heavily annotated or subtitle-rich content.
- *Complexity Filtering:* Visual complexity is assessed using a combined approach where the SAM 2 model [69] generates segmentation masks, complemented by a Sobel filter for detailed edge analysis. This pipeline effectively filters out overly simple images (e.g., plain backgrounds) based on quantified complexity metrics.
- *Object and Scene Classification:* To enrich metadata and enable conditional generation or analysis, models classify image content:
  - The YOLOv8 model[15], trained on OpenImagesV7 [70], detects and classifies ob-

[15] https://github.com/ultralytics/ultralytics

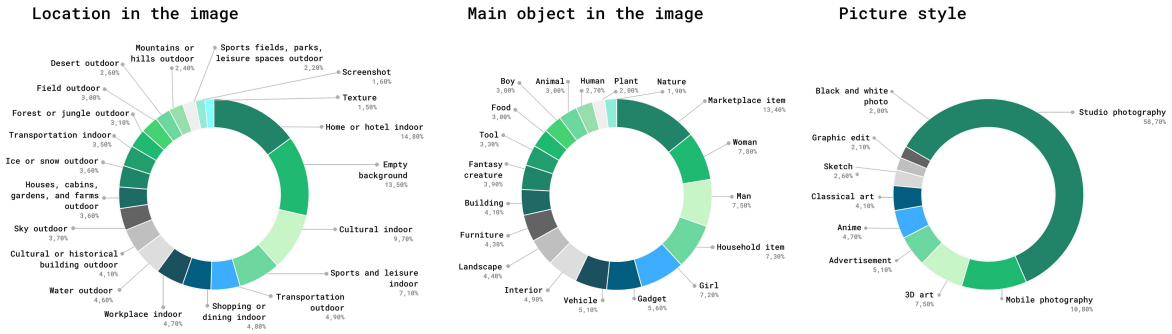


Figure 4: Distribution of key data categories across the curated Kandinsky T2I dataset by Location, Main object and Picture style.

jects present in the image.

* A CLIP-based classifier [51] categorizes the image's location, style, main subject, and detailed place type based on CLIP embeddings.

- **Content Annotation (Synthetic Captions):** High-quality synthetic English captions are generated for the images using powerful multimodal models to provide textual descriptions for training:
  - *InternVL2-26B* [71]*:* Generates initial detailed captions. Variants of this caption, refined or shortened by the InternLM3-8B model [72], are also included.
  - *Qwen2.5VL-32B* [73]*:* Generates Russian captions for higher-resolution images ($width * height \geq 512^2$).

  Post-processing is applied to these synthetic captions to improve their quality and consistency:
  - Regular expressions clean common introductory phrases (e.g., "The image shows").
  - Further filtering removes sentences containing non-English characters (e.g., Cyrillic, Chinese) to maintain language purity in English captions. This mitigates OCR errors from InternVL2, which performs poorly on scripts like Cyrillic (e.g., Russian), ensuring higher quality captions.
- **Organization and Storage:** Processed images, along with their metadata, filter scores, and captions, are stored in Parquet files. These files are organized into subdirectories based on shortest image side (256, 512, and 1024) and source dataset name on the specified S3 storage endpoint.

4.3 Image Editing Instruct Dataset Processing

The image editing instruct dataset (Kandinsky I2I) was constructed through a sophisticated multi-stage pipeline designed to identify high-quality image pairs suitable for editing tasks and to accomplish it with precise text instructions.

Processing Pipeline

The procedure involved the following steps:

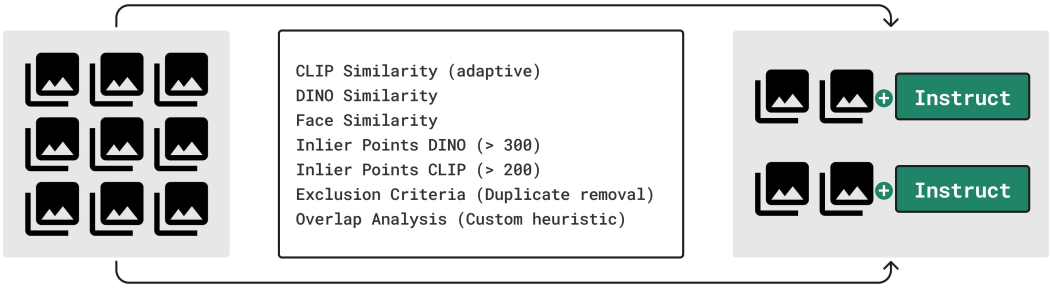


Figure 5: Instructive dataset processing pipeline. Initial image data (left) is processed through a set of similarity and exclusion criteria — including CLIP and DINO embeddings, face similarity, duplicate removal, and custom overlap heuristics — to produce filtered image-instruction pairs (right). Each retained image is paired with an `Instruct` token, forming high-quality instruction-tuned training samples.

1. **Initial Image Collection:** A diverse set of about 240 million images was compiled from various sources, ensuring broad coverage of different visual content, styles, and subjects.
2. **Similarity Matching:** To identify potential editing pairs, we employed multiple similarity metrics:
   - **CLIP Score [51]:** Measured semantic similarity between images using CLIP embeddings
   - **DINO Score [74]:** Computed visual similarity using DINOv2 features
   - **Face Recognition:** Specifically applied to images containing faces (only single-face images considered)
3. **Adaptive Thresholding:** Implemented an adaptive thresholding approach for CLIP similarity:
   - Images were clustered into 10,000 groups
   - Thresholds were determined per cluster based on top similarity scores
   - Final threshold: clip_sim $> (1 - T) *$ clip_thr $+ T$ where $T = 0.15$
4. **Geometric Verification:** Potential pairs underwent rigorous geometric verification using LoFTR [75] for feature matching:
   - Extracted feature points and matches between image pairs
   - Applied RANSAC algorithm [76] to estimate fundamental matrix and Euclidean transformation
   - Iteratively identified inlier groups (minimum 20 points per group)
   - Calculated total inliers as the sum of matched points across all valid groups
5. **Quality Filtering:** Applied multiple filtering criteria to ensure high-quality pairs:
   - **DINO inliers:** dino_sim > 0.8 AND inliers > 300
   - **CLIP inliers:** clip_sim > 0.8 AND inliers > 200 AND adaptive threshold condition

- **Face similarity:** face_sim > 0.7 for images containing faces
- **Exclusion criteria:** Removed pairs with dino_sim > 0.97 OR dino_aligned_sim > 0.97 OR (face_sim < 0.5 AND face_sim > 0)

6. **Aligned DINO Score Calculation:** For additional verification:
   - Used LoFTR [75] to find matching points between images
   - Applied RANSAC [76] to find Euclidean transformation
   - Cropped images to overlapping regions
   - Computed DINO similarity [74] on aligned crops (dino_aligned_sim)
7. **Overlap Analysis:** Filtered out simple crops by ensuring significant transformation between images:
   - Analyzed overlap percentage between images
   - Removed pairs with excessive overlap (indicating simple crops rather than edits)
8. **Caption Generation:** Finally, for each qualified image pair, we generated descriptive captions that explicitly highlighted the visual transformations to create suitable training data for editing models. Initially, the model was selected through an extensive Side-by-Side (SBS) comparative evaluation of several state-of-the-art multimodal models, including GPT-4o, GPT-4 Mini with reasoning, Gemini 2.5 Pro, and Qwen2.5-VL-32B. This qualitative human evaluation assessed the models' ability to produce coherent, precise, and instructional captions. Results ranked Gemini 2.5 Pro first, with the GLM 4.5 model [77] achieving competitive results. Given its favorable performance-to-cost ratio, we selected GLM 4.5 without reasoning and fine-tuned it using Low-Rank Adaptation (LoRA) [78] on a curated dataset to optimize it specifically for generating instructional captions. This adaptation resulted in a robust and cost-effective solution for our task.

Dataset Summary

This meticulous multi-stage filtering and verification process ensures the creation of an extensive, high-quality dataset comprising approximately **150 million** carefully curated image pairs (see Figure 6), each accompanied by comprehensive textual instructions that precisely describe the transformations between images, thereby providing an optimal foundation for effective training of advanced image editing models. However, to further enhance the model's aesthetic perception and instruction-following capabilities for final output refinement, we constructed an additional specialized dataset for supervised fine-tuning.

Supervised Fine-Tuning (SFT) Dataset Curation

To create a high-quality dataset for supervised fine-tuning, a subset was curated from the Kandinsky I2I dataset used for pre-training. We applied quality filters based exclusively on the target image (the second image in the pair), as our observations indicate that this approach is sufficient for identifying high-quality transformation examples while ignoring the source image. The filters required a Q-Align score greater than 4 and a Q-Align aesthetic score greater than 2 to select visually superior results. This filtering yielded approximately 600k candidate pairs. From this pool, human annotators manually select the most appropriate and high-quality image-editing pairs for the final SFT dataset. This meticulous manual curation ensures the SFT training data consists of exemplary instances, which is critical for effective instruction tuning.

(a) Instruction: Add a second, idetical glove and place it next to the first one, sligtly overlapping it.

(b) Instruction: Replace the black and white tracksuit with a grey hoodie with a yellow yoke and white swoosh, and brown trousers.

(c) Instruction: Replace the model with a man wearing a beige pinstripe suit with a white shirt underneath, and add a pair of glasses hanging from the lapel.

(d) Instruction: Change the woman's pose to walking forward, replace her black high heels with pink ones, and change the camera view to a full-length shot.

(e) Instruction: Open the car's butterfly doors.

(f) Instruction: Change the camera view to a wider shot, showing more of the stairs and the background, and make the lighting more dramatic with a dark, stormy sky.

Figure 6: Examples of collected pairs and generated instructions from Kandinsky I2I dataset

Table 1: Image editing dataset filtering criteria and parameters

Filtering Stage	Description	Threshold
CLIP Similarity	Semantic similarity with adaptive thresholding	> 0.8 + adaptive
DINO Similarity	Visual feature similarity	> 0.8
Face Similarity	For images containing single faces	> 0.7
Inlier Points (DINO)	Minimum matched points for DINO pairs	> 300
Inlier Points (CLIP)	Minimum matched points for CLIP pairs	> 200
Exclusion Criteria	Remove near-duplicates and bad face matches	Various
Overlap Analysis	Ensure significant transformation between images	Custom heuristic

4.4 Text-to-Video Dataset Processing

The Kandinsky T2V dataset comprises more than 250 million video scenes sourced from various open datasets and large video platforms. The data processing pipeline is a multi-stage process designed to ensure high-quality, diverse, and suitable data for pretraining the Kandinsky video generation model.

Processing Pipeline

The procedure involves the following stages:

- **Scene Segmentation:** The initial step involves segmenting raw videos into individual scenes using the PySceneDetect tool[16], which detects shot changes. This process isolates sequences of consecutive frames with a consistent visual perspective. Each extracted scene has a duration constrained between 2 and 60 seconds to ensure temporal coherence and manageability for subsequent processing.
- **Initial Filtering and Deduplication:** After segmentation, scenes undergo a series of initial filtering steps:
  - *Resolution Filtering:* Scenes with a shorter side of fewer than 256 pixels are removed to maintain a minimum visual quality standard.
  - *Deduplication:* To eliminate redundant content, a video perceptual hash [79] is computed for each scene. This hash is a fingerprint designed to be similar for visually alike videos, allowing for the identification and removal of identical or highly similar scenes. While the primary deduplication uses this method, some duplicates may still persist in the dataset.
- **Advanced Quality and Content Filtering:** Following initial filtering, a comprehensive suite of models is applied to assess various aspects of scene quality and content:
  - *Watermark Detection:* Similar to the approach for images, video watermark detection employs a combination of a dedicated classifier and an object detector. Scenes containing watermarks are filtered out by averaging the models' confidence scores across five evenly spaced frames and applying specific thresholds.
  - *Structural Dynamics:* This filter evaluates scene motion by sampling frames at 2 FPS and calculating the Multi-Scale Structural Similarity (MS-SSIM) index [80] between

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

consecutive pairs. A low average MS-SSIM score indicates high dynamism (rapid scene changes), while a high score indicates a static scene. Scenes deemed too static or excessively dynamic are filtered out based on threshold values.

- *Technical and Aesthetic Quality:* The DOVER model [81] is employed to provide separate scores for technical quality (e.g., sharpness, noise) and aesthetic quality (e.g., composition, color). An overall quality score is derived as a weighted sum of these two components. The Q-Align model [67], which uses large multimodal models to assess visual quality, is also applied to provide an additional quality assessment.
- *Text Detection:* The CRAFT model [68] is used to identify text regions within video frames. The number of detected text boxes and their total area are averaged over three evenly spaced frames. Scenes with an excessive amount of text are filtered to prevent the generation of videos dominated by subtitles or on-screen graphics.
- *Object and Scene Classification:* The YOLOv8 model, trained on the OpenImagesV7 dataset [70], detects objects in five frames per scene. A CLIP-based classifier [51] is used to classify the scene's location, style, main subject, and more detailed place categories by analyzing frame embeddings.
- *Camera and Object Dynamics:* A specialized model [82] based on VideoMAE [83] architecture was trained to predict scores for camera movement, object movement, and dynamics of light and color changes to further refine the assessment of scene activity.

- **Content Annotation:** For scenes that pass the filtering stages, synthetic English captions are generated to provide textual descriptions. This is achieved using the large multimodal model Tarsier2-7B[17]. To improve caption quality, post-processing steps are applied:
  - Regular expressions are used to remove common introductory phrases (e.g., "The video starts").
  - Captions are filtered to exclude sentences containing non-English characters (e.g., Cyrillic or Chinese), as the models can sometimes produce incorrect or mixed-language output.
- **Additional Processing (Clustering):** As an auxiliary process, scene embeddings are generated using the InternVideo2-1B model [84]. K-Means clustering [85] is then performed on these embeddings to group visually or semantically similar scenes into 10,000 clusters. This information is used for balanced sampling and dataset analysis.
- **Organization and Storage:** The final processed scenes, along with their metadata, filter scores, and captions, are organized into Parquet files. These files are grouped by the scene's shortest frame side (256, 512, and 1024 - see Table 2 for list of supported resolutions) and stored on an S3-compatible storage endpoint for efficient access during model training.

4.5 Russian Cultural Code Dataset Processing

The Kandinsky RCC dataset contains 229,504 video scenes and 768,555 images focused on the Russian cultural code (specific features linked to faith, language, historical memory, nature, architecture, personality, etc.).

[17] https://github.com/netease-youdao/Tarsier

Table 2: Supported Resolutions (height × width in pixels)

Lite Version	Pro Version
512 × 512	1024 × 1024
512 × 768	640 × 1408
768 × 512	1408 × 640
	768 × 1280
	1280 × 768
	896 × 1152
	1152 × 896

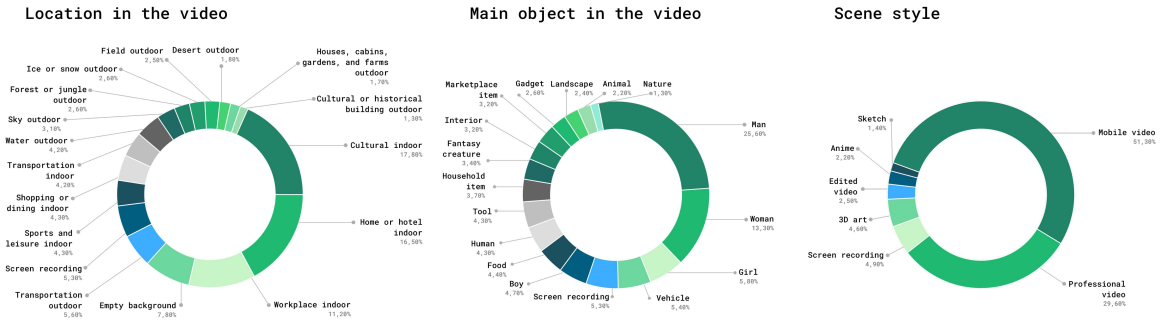


Figure 7: Distribution of key data categories across the curated Kandinsky T2V dataset by Location, Main object and Scene style.

Dataset Characteristics

- **Curation Method:** Unlike the T2V and T2I datasets, the RCC data is manually curated by annotators based on relevance to the Russian cultural code, visual quality, and the absence of watermarks or subtitles.
- **Annotations:** The images and scenes are accompanied by manually written Russian descriptions, which are then machine-translated into English with special handling for proper names.
- **Usage:** This dataset is used both for pretraining and for specialized fine-tuning to improve the model's performance on culturally specific content.

4.6 Supervised Fine-Tuning Dataset Processing

A high-quality Supervised Fine-Tuning (SFT) dataset was meticulously curated to align the model's outputs with human preferences and significantly enhance visual and compositional quality. This dataset is distinct from the large-scale pretraining data, as it consists of a smaller set of high-fidelity examples manually selected by expert annotators through a rigorous multi-stage evaluation process. The dataset comprises both images and video content. Additionally, using a video language model (VLM), all data was classified into 9 domains (see Figure 8). Moreover, images in SFT dataset were split into more detailed hierarchy of classes to obtain the best possible model.

The domain-based organization enables:

- Domain-specific training approaches and curriculum learning strategies
- Composition of SFT-soup models by weights averaging (following the approach described in [86])

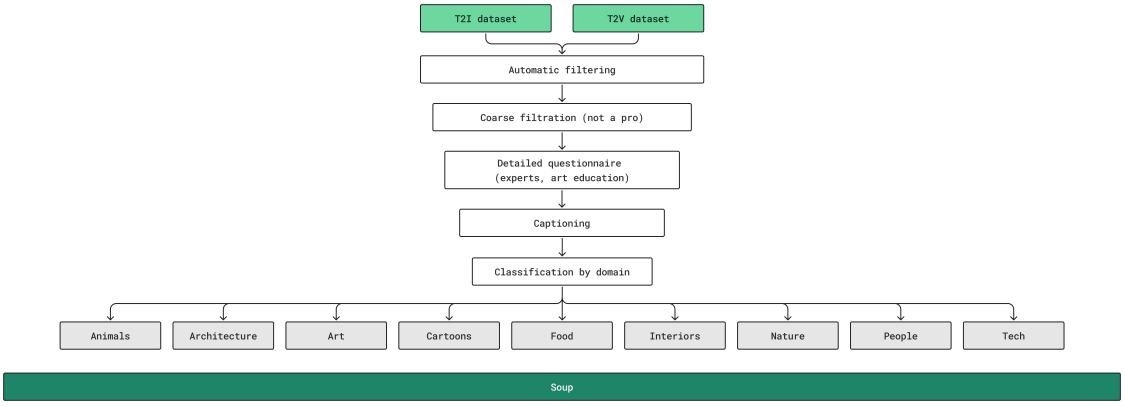


Figure 8: SFT dataset curation pipeline. Data from Kandinsky T2V and T2I datasets undergoes automatic technical quality and aesthetic filtering, followed by coarse human filtering ("not a pro"), detailed expert evaluation (including art education specialists), captioning, and domain-based classification into certain categories. All curated data is used to finetune class-specialized models and mix them into final soup SFT model.

- Balanced sampling across different content categories during training
- Specialized fine-tuning for particular content types while maintaining overall model coherence
- Controlled mixing ratios between video and image data within training batches
- Targeted improvement of model performance in specific visual domains

Dataset Construction

- **Initial Collection:** The initial pool consisted of 93,296 high-resolution video scenes and approximately 10 million images, sourced automatically for their exceptional technical quality (e.g., resolution threshold, lack of artifacts, watermarks) and aesthetic merit.
- **Captioning:** For all images we implemented a specialized two-stage captioning pipeline to maximize instruction following capability: first, the **Qwen2.5-VL-32B-Instruct** model generated a long textual description for the input image; second, the **Qwen3-32B** model rewrote this description into several variations: very long, long, medium length, short, and very short. Ablation study was conducted to select the optimal system prompt for each model to ensure the final textual descriptions were maximally accurate and approximated human speech. Since our models were pretrained on both English and Russian textual descriptions, we want to continue this approach during SFT, requesting the language model to return descriptions in both languages simultaneously for consistency.

  To provide rich textual descriptions for each video, automatic captioning was performed using the **SkyCaptioner-V1** [87] and **Qwen2.5-VL 32B and 72B** [73] models. This process generated multiple caption variants of differing lengths and descriptive density for each record. The captions obtained were cleaned with regular expressions to avoid useless common parts such as "the image shows".

- **Multi-Stage Expert Evaluation:** The data passed through a rigorous two-stage evaluation process with specialized annotator roles:

  Stage 1 - Technical Screening: Regular annotators assessed content based on fundamental quality criteria included:

  - Presence and prominence of main object
  - Proper cropping and framing
  - Spatial depth perception
  - Absence of visual artifacts

  Stage 2 - Comprehensive Quality Assessment: Qualified experts with background in cinematography and visual arts evaluated the pre-filtered content using detailed questionnaires listed below:

  Image Evaluation Criteria

  - Exposure and contrast correctness
  - Horizon line positioning
  - Composition geometry
  - Object silhouette quality
  - Lighting and color scheme
  - Absence of unwanted objects
  - Overall artistic expressiveness
  - Natural, non-AI appearance
  - Organic integration
  - Need for modifications
  - Exceptional "wow factor" quality

  Video Evaluation Criteria

  - Video integrity and completeness
  - Clarity of content and action
  - Main object presence and highlighting
  - Dynamic properties (high/low)
  - Color solution and grading
  - Framing correctness
  - Horizon line positioning
  - Composition quality: rule of thirds, central, diagonal
  - Depth and volumetrics
  - Exposure correctness
  - Suitability for editing
  - Complexity of content
  - Aesthetic and artistic expression

- **Specialized Text Handling:** Images containing text underwent additional evaluation with specialized criteria:

  - Text clarity and legibility
  - Appropriate text overlay integration
  - Absence of unusual letter distortions
  - Harmonious text-background relationship
  - Special categorization for poster/cover designs

- **Quality Thresholds:** For the first version (v1) of the dataset, the selection was based on strict criteria requiring consensus among evaluators (minimum $\frac{3}{2}$ agreement on "good" ratings across all parameters). This rigorous approach resulted in the selection of approximately 3% of the initial video pool and 5% of images, prioritizing content with clearly defined subjects, balanced composition, and high production values. We conducted extensive research on question selection, acceptance thresholds, and artifact detection. This led to two dataset versions:

  - **v1 (strict):** 2,833 video scenes and 45,000 images with comprehensive criteria
  - **v2 (relaxed):** 12,461 video scenes and 153,000 images with optimized thresholds

  During fine-tuning, we sample from both datasets with calibrated probabilities to balance quality and diversity.

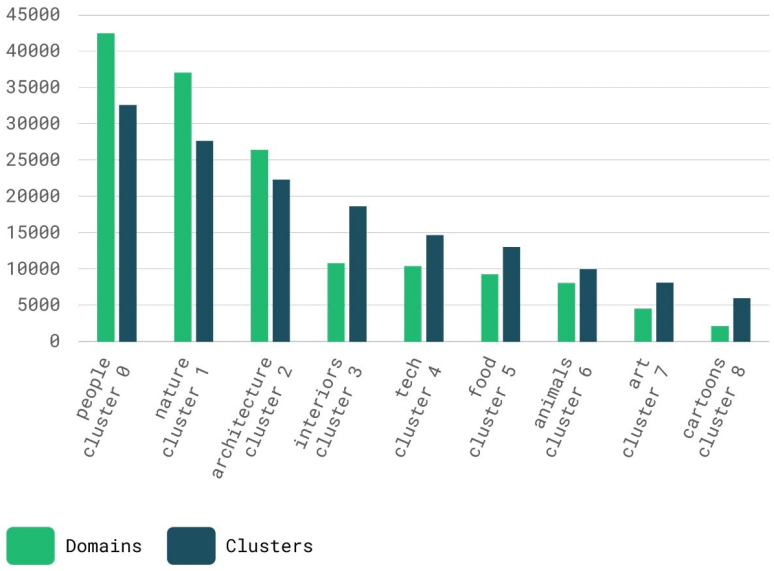
VLM Domain Organization

Both video and image data were classified into 9 consistent domains using the Qwen2.5-VL-Instruct-32B model [73]. This ensures coherence between the video and image domains during mixed-dataset training of the Text-to-Video (T2V) model. The classification used a standardized prompt approach:

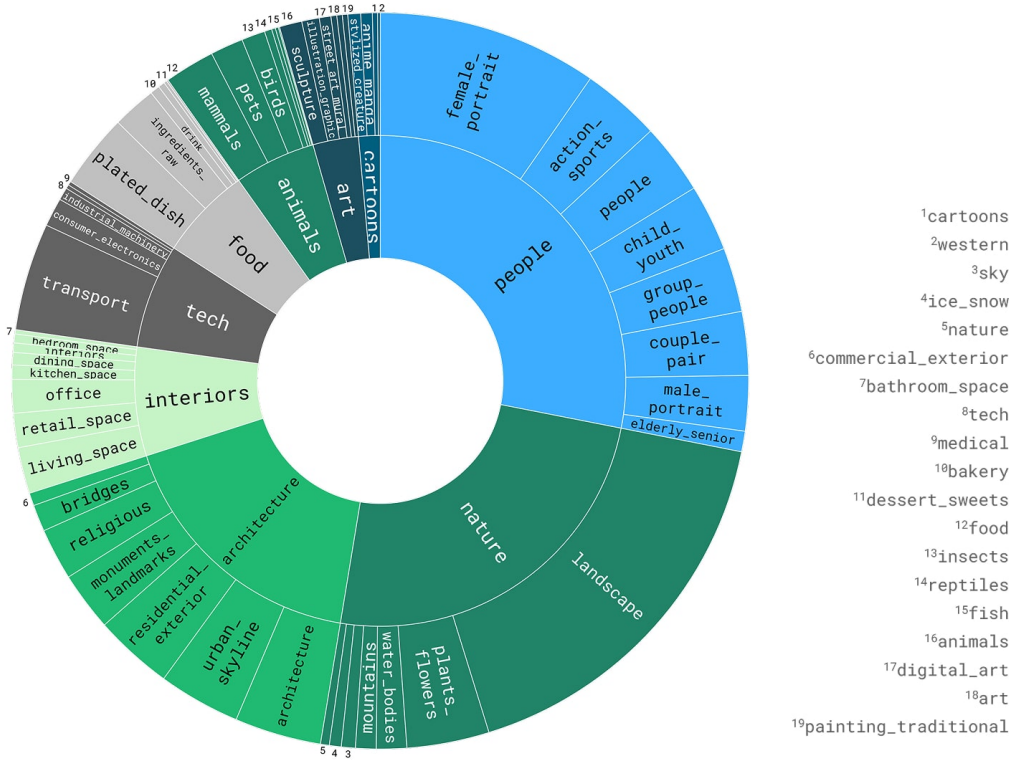
```
CATEGORY_LIST = """
1) animals
2) architecture
3) art
4) cartoons
5) food
6) interiors
7) nature
8) people
9) tech
10) other
"""

PROMPT_TEMPLATE = """
You are a professional {media_type} classificator. You are given a {media_type}.
Your task is to return one of the following classes this {media_type} relates to the most.
Choose one class from the following classes:
{CATEGORY_LIST}
Return only the name of class.
""".strip()
```

The final SFT-soup model for T2V was composed by averaging weights of models fine-tuned on meaningful 9 domains, using a simplified approach with equal weighting.



(a) Difference in distributions between VLM domains and K-Means clusters.



(b) Distribution of data across VLM domains and subdomains.

Figure 9: Data distributions for different domain classification and clustering strategies used during T2I SFT dataset organization.

Advanced Domain Organization and SFT-soup Composition Ablation Study

For the Text-to-Image (T2I) model, we employed a more sophisticated hierarchical domain organization strategy.

We experimented with multiple domain decomposition strategies for following optimal SFT-soup composition:

- **VLM Domain Classification:** Full fine-tuning was performed on each VLM domain from the previous section with reduced batch size (64 vs. 4096 in pretrain) and learning rate (1e-5 vs. 1e-4).
- **CLIP Embedding Clustering:** We alternativeively used **CLIP-ViT-H-14-quickgelu** embeddings with k-means clustering into 9 clusters. This approach increased data diversity within individual components, reducing overfitting and enabling longer training for better realism.
- **Hierarchical VLM Domains:** Each VLM domain was further divided into 2-9 semantically coherent subdomains. Separate full fine-tuning on these smaller, more homogeneous subsets allowed for targeted improvement in specific visual characteristics while maintaining overall coherence.

For weight merging in SFT-soup composition, we compared three approaches: equal weights (1/N), weights proportional to dataset size, and weights proportional to the square root of dataset size. Our experiments showed that equal weights or root-proportional weights performed best. The hierarchical domain approach yielded the best results in side-by-side evaluations, achieving high realism, improved text rendering quality, and compositionally correct generated images.

The resulting SFT dataset represents a carefully curated collection of visual content that exemplifies high aesthetics and enables robust model fine-tuning across diverse visual domains.

5 Kandinsky 5.0 Architecture

5.1 Model Overview

All models in the Kandinsky 5.0 family are built upon a unified architecture based on a latent diffusion pipeline [4] and trained using the Flow Matching paradigm [3]. The core of this architecture is a specially designed Diffusion Transformer with cross-attention (**CrossDiT**) for multimodal fusion of visual and textual information. The number of CrossDiT blocks varies depending on the model size. The architecture also integrates text and visual encoders for efficient input processing:

- **Text Encoding**: Text representations are extracted using the **Qwen2.5-VL model** [73], a transformer decoder architecture that generates rich text embeddings. These embeddings are further processed by a Linguistic Token Refiner module before being passed into the main CrossDiT backbone.
- **Visual Encoding**: We use **FLUX.1-dev VAE**[18] [9] to encode images. Video latents are obtained using an encoder from the **HunyuanVideo VAE** [26], which produces compact latent representations suitable for the diffusion process while maintaining temporal consistency.

[18] https://huggingface.co/black-forest-labs/FLUX.1-dev

Below, we examine in more detail the structure of the CrossDiT backbone architecture (Section 5.2), the CrossDiT blocks (Section 5.3), and our proposed Neighborhood Adaptive Block-Level Attention (**NABLA**) mechanism [36], which enables significant acceleration and optimization for video generation tasks (Section 5.4). In our work, we place strong emphasis on computational efficiency and training stability, which are discussed further in Section 7.

5.2 Diffusion Transformer (CrossDiT) Architecture

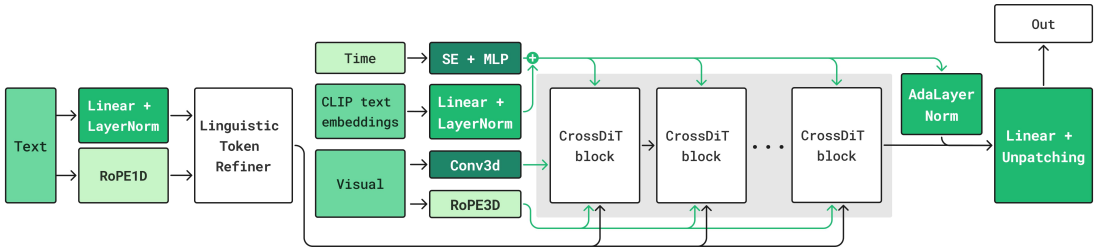


Figure 10: CrossDiT architecture.

The CrossDiT architecture is illustrated in Figure 10. Its core consists of a sequence of CrossDiT blocks. The model takes four distinct types of inputs:

- **Text:** Text embeddings from Qwen2.5-VL are passed through a Linear layer and LayerNorm [88]. One-dimensional positional embeddings are also generated for the text using Rotary Position Encoding (RoPE) [89]. These are combined and fed into the Linguistic Token Refiner (LTF) module. This module, which is a CrossDiT block without the cross-attention component (see Figure 11), serves to enhance the text representation and eliminate the positional bias inherited from the pre-trained text encoder [90]. The refined text queries are then fed into the main CrossDiT blocks via the cross-attention mechanism [91].
- **Time:** The diffusion timestep value is processed by a block comprising Sinusoidal Encoding (SE) [92] and a Multi-Layer Perceptron (MLP).
- **CLIP Text Embedding:** A single text embedding of the full video description from CLIP ViT-L/14 model[19] [51] is passed through a linear layer and LayerNorm, and then summed element-wise with the time embedding. This resulting sum, along with the output from the final CrossDiT block, is fed into the Adaptive Normalization Layer [17].
- **Visual:** Image latents from FLUX.1-dev VAE or video latents from the HunyuanVideo VAE encoder are used to generate 3D Rotary Positional Embeddings, which are then fed into every CrossDiT block.

Architecture hyperparameters. For all models, we use Qwen2.5-VL as the main text encoder with 7 billion parameters, an embedding size of 3584 and a maximum context length of 256. We also use CLIP ViT-L/14 with a text embedding size of 768 and a maximum context length of 77. Hyperparameters for the architecture of the main CrossDiT part of our models are shown in Table 3.

[19] https://huggingface.co/sentence-transformers/clip-ViT-L-14

Table 3: CrossDiT hyperparameters for the Kandinsky 5.0 family of models.

Model	Number of CrossDiT blocks	Number of LTF blocks	Linear layer dimension	Model embedding dimension	Time embedding dimension
Image Lite	50	2	10240	2560	512
Video Lite	32	2	7168	1792	512
Video Pro	60	4	16384	4096	1024

5.3 CrossDiT Block Architecture

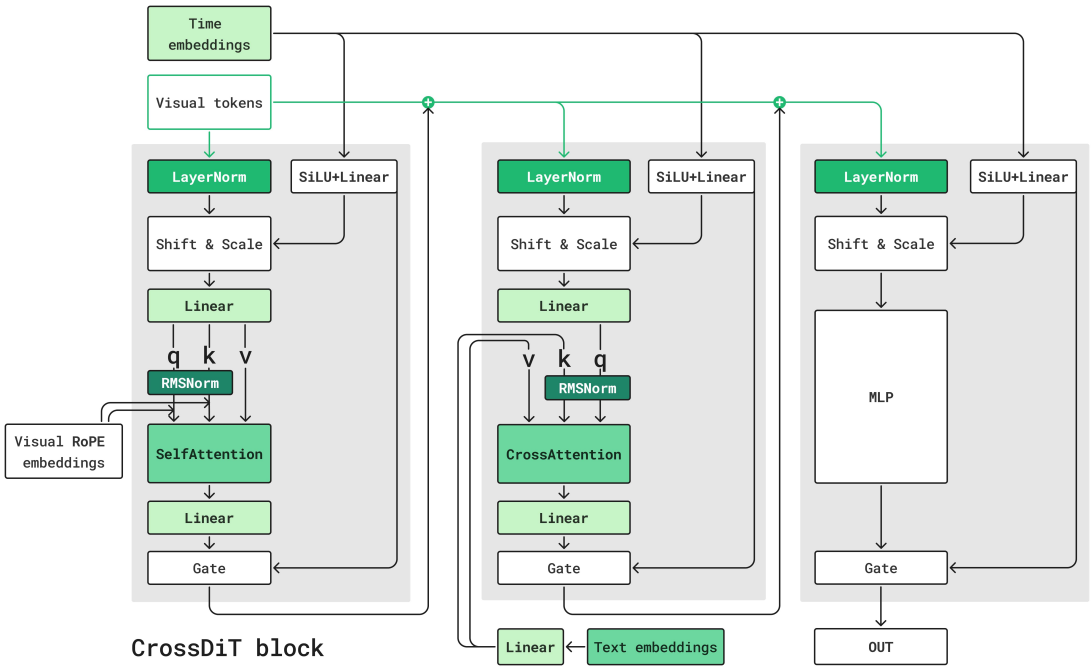


Figure 11: CrossDiT block architecture. From left to right: a Self-Attention Block, a Cross-Attention Block, and a MLP Block.

At the heart of the **CrossDiT** block are classic residual connections: three sequential sub-blocks handling: **Self-attention**, **Cross-attention** and **Multi-Layer Perceptron (MLP)** (Figure 11). The sum of outputs from the Self-attention and Cross-attention attention with the input visual latents is shown on the diagram with a "+" symbol.

The advantages of this cross-attention architecture lie in its better compatibility with the **sparse attention mechanisms** required for processing videos of varying lengths within a single batch. In contrast, the MMDiT-like architecture [8] used in Kandinsky 4.0 required a concatenation operation that significantly slowed down training. In Kandinsky 5.0, we have successfully eliminated this need.

5.4 Neighborhood Adaptive Block-Level Attention

To reduce the computational complexity of attention layers and accelerate the training process for high-resolution (up to 1024px) or long-duration (up to 10 seconds) video generation, we employ **NABLA** (Neighborhood-Adaptive Block-Level Attention) - a sparse attention mechanism that dynamically constructs content-aware masks for efficient video diffusion transformers.

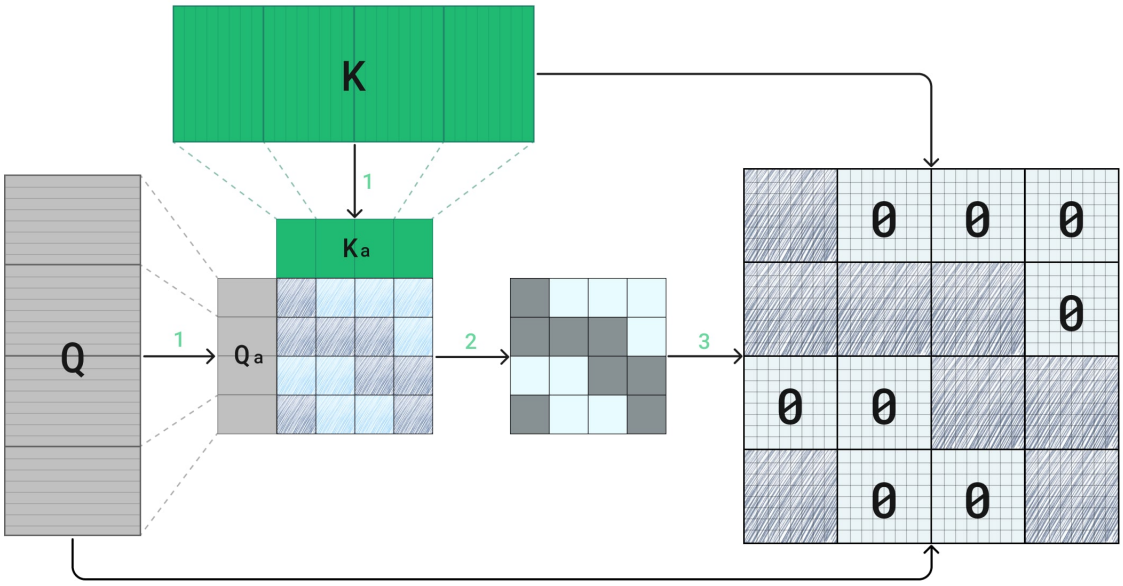


Figure 12: The block-sparse attention mask is computed by (1) reducing the dimensionality of queries (Q) and keys (K), (2) sparsifying the softmax distribution via a cumulative density function (CDF) threshold and binarizing the result, and (3) mapping the sparse mask back to the original input blocks.

As shown in Figure 12, NABLA constructs content-aware sparse attention masks through a three-stage process:

1. **Block-wise dimensionality reduction**: Queries and keys are average-pooled by groups of $N = 64$ elements, reducing the attention map computation complexity by a factor of $N^2 = 4096$ while maintaining the structural relationships essential for video coherence.
2. **Adaptive sparsification via CDF thresholding**: For each attention head, we compute the cumulative distribution function of the reduced attention map and dynamically select the most relevant blocks by thresholding at $1 - thr$, where thr controls the sparsity level. This ensures that each head preserves its unique attention pattern tailored to the input content.
3. **Border artifact suppression**: The resulting adaptive mask is optionally combined with Sliding-Tile Attention patterns through union operation to maintain local continuity and suppress potential border artifacts that can occur in high-resolution generation.

NABLA employs token reordering through fractal flattening with spatial patches of size $P \times P$ ($P = 8$), grouping all tokens within each patch into contiguous sequences of $P^2 = N = 64$

tokens while preserving the original temporal ordering as illustrated in Figure 15. This reorganization optimizes memory access patterns by ensuring spatially adjacent tokens remain contiguous in memory, significantly improving computational efficiency during attention computation. The reordering operation is applied at the DiT network input with its inverse at the output, maintaining proper spatial relationships throughout the processing pipeline.

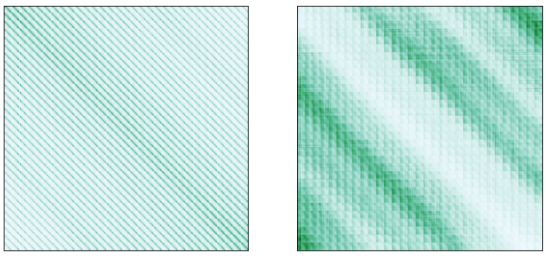


Figure 13: Real attention maps.

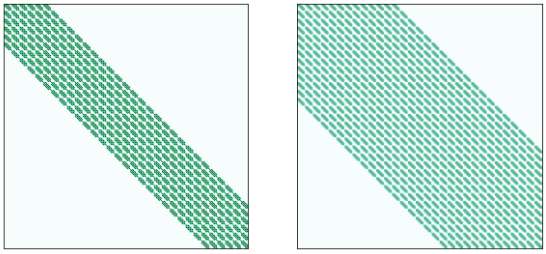


Figure 14: STA masks with different window sizes.

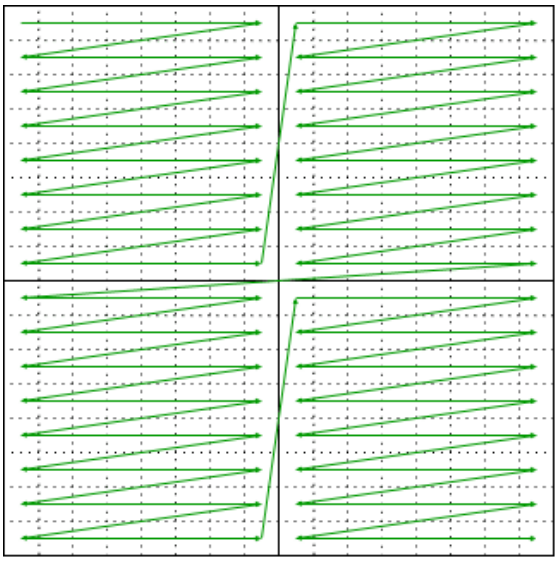


Figure 15: Token reordering illustration for a latent image with height 16, width 16, and patch size 8. The diagram shows how spatial tokens are reorganized into fractal-flattened sequences while preserving their semantic relationships.

NABLA generates head-specific sparsity patterns that adapt to varying attention maps observed in real transformer layers (Figure 13), overcoming limitations of fixed sparse patterns like STA (Figure 14) that may not capture complex long-range dependencies required for coherent video generation. Extensive evaluation demonstrates that NABLA achieves up to **2.7× speedup** in training and inference while maintaining equivalent generation quality to full attention, as validated by both quantitative metrics (CLIP, VBench) and human evaluations.

NABLA integrates seamlessly with PyTorch's FlexAttention framework without requiring custom CUDA kernels or additional loss functions, making it practical for both training and inference of large-scale video generation models. For complete implementation details and algorithmic specifications, see [36].

6 Training Stages

6.1 Training Infrastructure

We pre-train our models on a standard NVIDIA multi-node cluster. Each node contains 8 GPUs connected by NVLink. InfiniBand was used for inter-node connection. We used S3 instead of NFS to store dataset because the weight of the dataset was O(10) Pb and NFS is too expensive for this case.

6.1.1 Data Storage

All data were stored in an S3-compatible object storage and streamed during training over a 100 Gbit/s link. The storage system allowed only $\mathcal{O}(10^3)$ concurrent connections, so the data pipeline was designed to keep the number of opened objects small and the throughput per connection high.

Latent pre-encoding and storage layout. To reduce I/O and avoid repeated encoder calls on the fly, all images and videos were pre-encoded with the VAE, and *VAE latents* were stored instead of raw pixels. These latents were then packed into `.tar` archives so that each archive had approximately the same size. The number of samples per archive was chosen as a function of resolution and modality:

- **Images**
  - Low resolution: 1024 latents per tar
  - Medium resolution: 256 latents per tar
  - High resolution: 64 latents per tar
- **Videos** (one "latent" here denotes the full VAE-latent sequence for a video)
  - Low resolution: 16 video latents per tar
  - Medium resolution: 4 video latents per tar
  - High resolution: 1 video latent per tar

This scheme keeps archive sizes roughly uniform across resolutions while minimizing the total number of S3 objects that need to be opened during training.

Text embeddings. Caption text embeddings were computed on the fly during training rather than stored in S3. A single text embedding is approximately 50$\times$ larger than a low resolution (e.g., 256$\times$256) image latent; precomputing and storing all text embeddings would therefore significantly increase both the storage footprint and the per-object size, putting additional pressure on the S3 bandwidth and connection limits. Computing them online keeps the S3 load dominated by compact image/video latents and simplifies the storage layout.

6.1.2 DataLoader Design

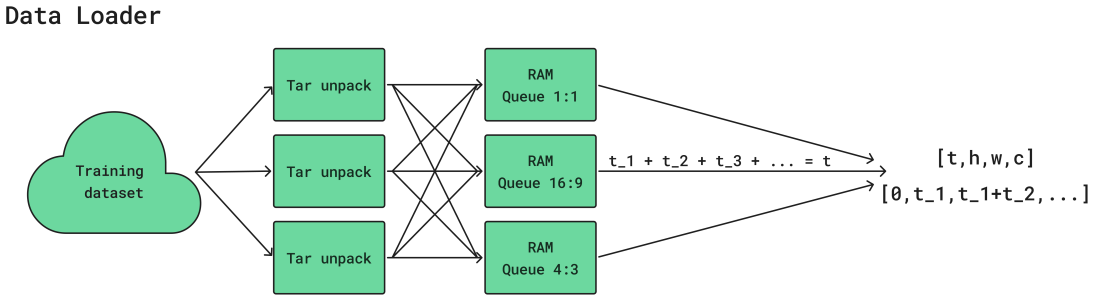
The data loader was implemented as a two–stage pipeline. A dedicated worker process streamed `.tar` archives from S3, unpacked them on the fly, and pushed individual VAE latents into in-memory queues. Each queue corresponded to a particular aspect ratio (e.g., 1:1, 16:9, 4:3), so that the main training process could sample shape-compatible clips without additional padding or resizing (see Figure 16a).

The main process constructed batches by repeatedly popping video latents from a selected aspect-ratio queue until the *sum of their temporal lengths* reached a predefined maximum:

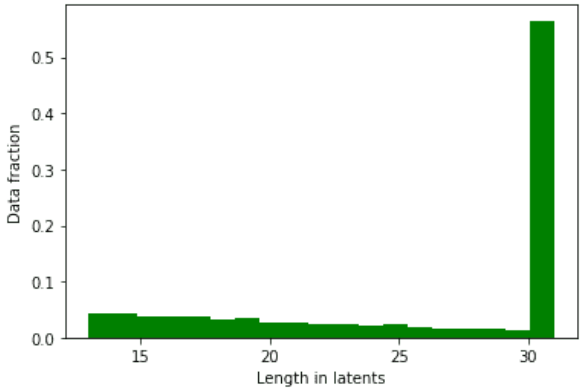
$$\sum_i t_i \approx t_{\max}.$$

Images were treated as videos of length 1, but they were always sampled in *image-only batches*. This allowed us to explicitly control the global fraction of image steps during training: a high proportion of images in mixed batches was empirically found to degrade convergence on video tasks.

Concatenating videos of different lengths into a single batch in this way significantly reduced idle time and improved GPU utilization. For large temporal contexts we additionally



(a) Data loader scheme with S3 streaming, tar unpacking, and aspect-ratio queues.



(b) Histogram of video latent temporal lengths t in the dataset.

Figure 16: Data streaming pipeline and distribution of video temporal lengths.

employed *adaptive attention*, so the cost of processing one long video was comparable to processing several shorter clips with the same total number of tokens. The distribution of temporal lengths in the dataset (Figure 16b) is highly skewed: most videos are close to the maximum allowed length, with a relatively small but non-negligible portion of shorter clips, which makes this dynamic batching strategy particularly beneficial.

6.1.3 Distributed Training and Memory Optimization

We used *HSDP* [93] for distributed training at all stages, from low resolutions up to the final high resolution setting. Starting from the medium resolution stage, we additionally enabled *Sequence Parallel* [94]. The weights of the Diffusion Transformer were partitioned across 64 GPUs, and the text encoder weights were partitioned across 32 GPUs. This sharding scheme provided full overlap of computation and communication at all training stages and resulted in very low per-GPU memory usage for both model parameters and optimizer states.

We chose Sequence Parallel instead of Tensor Parallel because it requires only two collective operations per transformer block (before and after the self-attention module). Sharding the weights inside a single block was unnecessary, as the block itself was relatively lightweight. For HD video training with 10-second sequences, we used the maximum sequence-parallel sharding over 8 GPUs, which ensured that all collectives stayed within a single NVLink island.

Checkpointing was performed in a non-blocking manner. Each of the 64 processes wrote its parameter shard directly to storage, without reconstructing the full `state_dict` on any node.

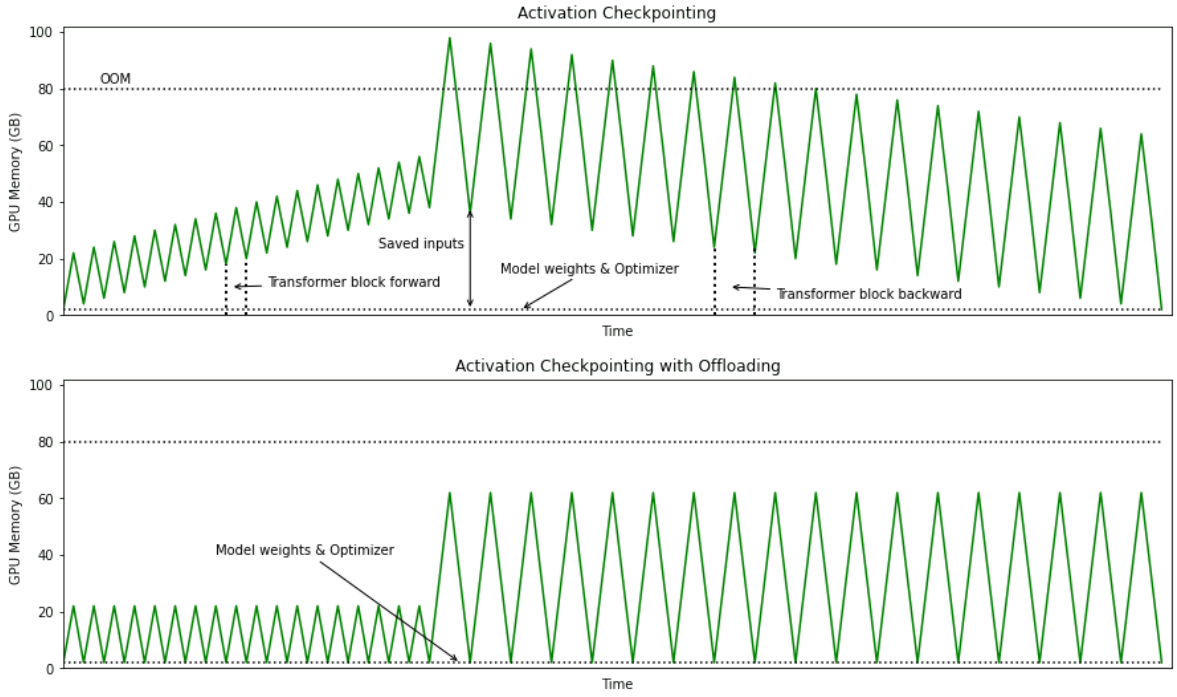


Figure 17: Activation memory traces for standard activation checkpointing (top) and activation checkpointing with host offloading (bottom).

Activation checkpointing and offloading. To further reduce memory usage we applied activation checkpointing [95] the granularity of transformer blocks. During pre-training we used the classical scheme, storing only a subset of activations and recomputing the rest during the backward pass. For the RL fine-tuning stage, where we had to keep activations for the entire generation trajectory, we used an extended variant with offloading: the inputs to each transformer block were temporarily moved from GPU memory to host RAM between the forward and backward passes.

Offloading was implemented in a non-blocking fashion: device–host transfers were issued asynchronously and overlapped with computation. As a result, the additional data movement did not noticeably slow down training while still providing a substantial reduction in peak activation memory.

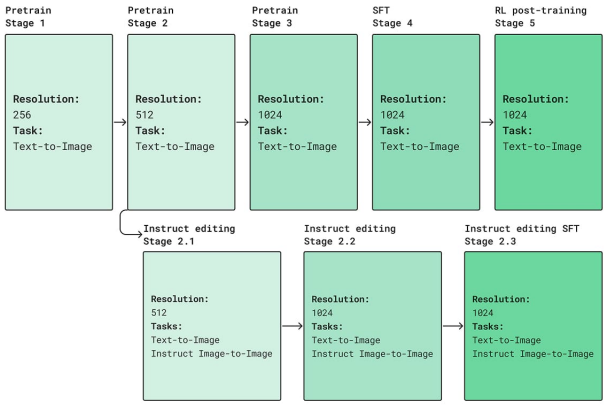
The effect of this modification is illustrated in Figure 17. Compared to standard activation checkpointing (top trace), activation checkpointing with offloading (bottom trace) significantly lowers the peak activation footprint while preserving a similar compute pattern.

6.2 Training Procedure Overview

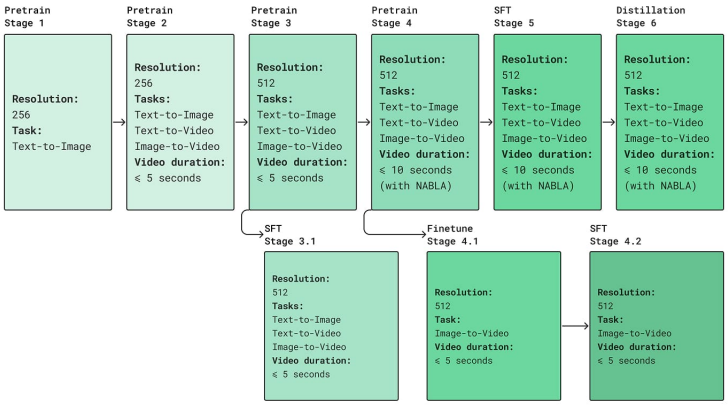
The training process follows a multi-stage training pipeline, which includes **pre-training** (Section 6.3), **supervised fine-tuning (SFT)** (Section 6.4) and **distillation** (Section 6.5) for video models. The image generation model undergoes an additional **Reinforcement Learning (RL)-based Post-Training** stage (Section 6.6) for visual quality enhancement. Figure 18 illustrates the overall scheme of training stages. More specifically:

Kandinsky 5.0 Image Lite 6B models. Both models share a common backbone from initial pre-training on text-to-image generation at **low** (192, 256, 320, 352, etc) and **medium** (384,

Kandinsky 5.0 Image & Image Editing 6B



Kandinsky 5.0 Video Lite 2B



Kandinsky 5.0 Video Pro 19B

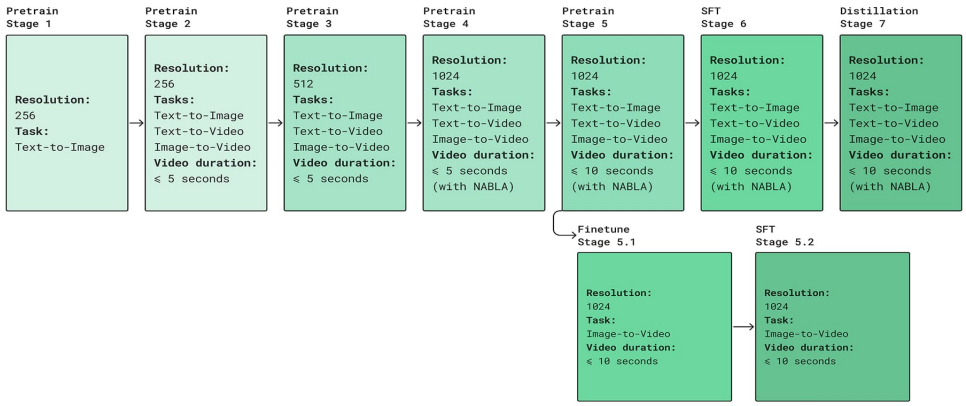


Figure 18: The training stages for models of the Kandinsky 5.0 family.

512, 640, etc) **resolutions**. Kandinsky 5.0 T2I Lite continues text-to-image pre-training at a **high resolution** (1024, 1280, 1408, etc) and then undergoes SFT and RL-based post-training for enhanced visual quality. Kandinsky 5.0 Image Editing inherits the checkpoint from the shared medium resolution pre-training stage and then undergoes two instructional editing fine-tuning stages at medium and high resolutions with subsequent SFT. During these three stages, the model performs instructional editing 80% and text-to-image generation 20% of cases.

Kandinsky 5.0 Video Lite 2B models. The pre-training begins with independent text-to-image generation stage with a low resolution. Subsequently, during most stages, the model is trained to solve three tasks simultaneously – text-to-image generation, text-to-video generation, and image-to-video generation with probabilities of 1%, 79%, and 20%, respectively. This approach is applied during the pre-training stages at low and high resolutions with a maximum video length of 5 seconds, after which an SFT stage for this video length follows. During the pre-training stage with a maximum video length of 10 seconds, we employ the NABLA mechanism to enhance the efficiency of the attention operation (see Section 5.4). After this stage, we conduct fine-tuning for the image-to-video generation task and an SFT stage with a video length of 5 seconds to obtain a specialized 5-second version of the model for this task. In parallel, we continue to train the model for all three tasks simultaneously with a video length of 10 seconds during the subsequent SFT and distillation stages, ultimately yielding the text-to-video generation model **Kandinsky 5.0 T2V Lite**, text-to-image generation model **Kandinsky 5.0 I2V Lite** and accelerated version, **Kandinsky 5.0 Video Lite Flash**.

Kandinsky 5.0 Video Pro 19B models. The pre-training for these models also begins with independent stage of text-to-image generation at a low resolution. We then conduct 4 stages of pre-training, as well as SFT, on the tasks of text-to-image, text-to-video, and image-to-video generation with probabilities of 2%, 77%, and 21%, respectively. During pre-training stages, we increase the maximum resolution to 1024 pixels and the maximum video length from 5 to 10 seconds. For the high resolution, we use the NABLA mechanism (Section 5.4). We also conduct specialized fine-tuning on the image-to-video generation task for **Kandinsky 5.0 I2V Lite** model, followed by an SFT stage. As a result of distillation of the model, which solves the three aforementioned tasks, we obtain the accelerated version, **Kandinsky 5.0 Video Pro Flash**.

6.3 Pre-training

6.3.1 Regimes

We follow four core schemes for models pre-training in text-to-image, text-to-video, image-to-video, and instruct editing regimes (Figure 19), each employing different weights of our CrossDiT architecture (Section 5.2). Below, we elaborate on each scheme in detail.

Text-to-Image Generation. This scheme is applied for training all models, including the initial pre-training stages for video generation (see the training schemes in Figure 18). The input to CrossDiT consists of a noisy image latent, the timestep in the diffusion process, and a text representation of the image description from the Qwen2.5-VL encoder [73] using the following template *caption prompt*:

```
"<|im_start|>system\nYou are a promt engineer. Describe the image by detailing the
color, shape, size, texture, quantity, text, spatial relationships of the objects and
background:<|im_end|>",
"<|im_start|>user\n{}<|im_end|>".
```

An image description is inserted into `user\n{}`. This format of text input is more suitable for a VLM text encoder, which is trained on instructions [90]. The model also receives a CLIP [51] text

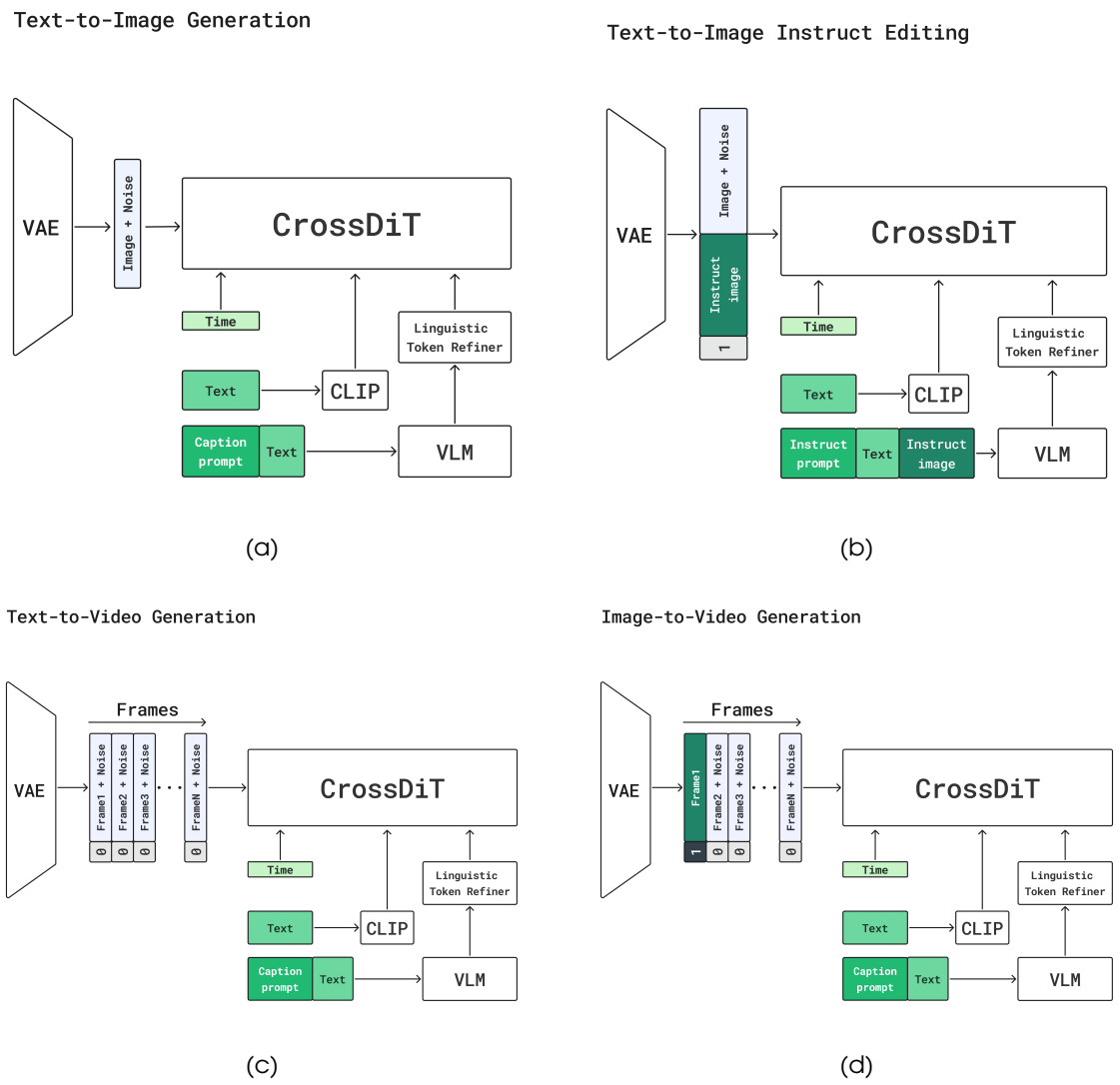


Figure 19: Pre-training setups for text-to-image generation (a), instruct image editing (b), text-to-video (c) and image-to-video regimes (d) for Kandinsky 5.0 models.

embedding corresponding to the full image description. For pre-training the video model, a single-channel mask of zeros is added to the image.

Text-to-Image Instruct Editing. This scheme is used for training Kandinsky 5.0 Image Editing. It differs from the previous one in that the input noisy image is channel-wise concatenated with an instruct image, which serves as a reference for the editing task, and a single-channel mask of ones. The instruct image is also fed into Qwen2.5-VL along with the textual image description, supplemented by the following *instruct prompt*:

```
"<|im_start|>system\nYou are a promt engineer. Based on the provided source image
(first image) and target image (second image), create an interesting text prompt that
can be used together with the source image to create the target image:<|im_end|>",
```

`"<|im_start|>user\n{}"`.

Text-to-Video Generation. Here, the input to the video transformer is a sequence of noisy frames with a zero-valued single-channel mask. For 1024 resolution or for videos up to 10 seconds long at 512 resolution, CrossDiT incorporates the NABLA mechanism (Section 5.4); in other cases, full attention is applied to the video sequence. For this task, the corresponding caption prompt is also used to describe the video.

Image-to-Video Generation. The first frame of the video sequence remains unnoised and, unlike the other frames, is accompanied by a single-channel mask of ones.

6.3.2 Training details

We train all our models at the pre-training stage with the AdamW optimizer [96] with `betas=(0.9, 0.95)` and `eps=1.0e-08`, changing the learning rate and `weight_decay` depending on the training stage (see Table 4). We use scheduler warmup for all pre-training stages, limit the gradient norm to one, and also apply an exponential moving average with a parameter of 0.9999. During training, we use unconditional data examples with a probability of 0.1. Other parameters that depend on the models and training stages, such as the batch size, the number of optimizer steps, and the probabilities of spatial resolutions used in the trained data are presented in Table 4.

6.4 Supervised Fine-tuning

This section outlines the general concept of supervised fine-tuning (SFT) for our models, following the established training protocol (see Figure 18). The approaches described below are applicable for all SFT stages in the training procedure.

6.4.1 Image Generation

Following the pretraining stage, we perform supervised fine-tuning on a dataset of 153 thousand high-quality, realistic images with detailed text descriptions. Data selection was conducted by annotators based on a range of criteria, including technical quality, composition, perspective, lighting, colour palette, and overall aesthetics. For more details, please refer to Section 4.6.

We found that direct full fine-tuning of the pretrained model on the entire SFT dataset yielded unsatisfactory results; namely, it maintained a non-photorealistic style, degraded text alignment, and reduced the quality of text rendering within images. Consequently, we employ a **model souping** technique [86, 97] with **hierarchical clustering**. The source dataset is clustered into 9 thematic domains (e.g., "people", "nature") using a VLM. To further enhance composition, realism, and text rendering, each domain is subsequently segmented into 2-9 semantically homogeneous subdomains (Figure 9).

We perform independent full fine-tuning on each subdomain with a batch size of 64 and a peak learning rate of 1e-5. Training on a subdomain is halted upon the emergence of generation artifacts on the validation set, such as geometrical distortions or colour aberrations. The resulting checkpoints within a single main domain are then aggregated via a weighted summation of their model parameters, with weights proportional to the square root of the size of each subdomain.

The final model is obtained by performing a weighted averaging of the nine domain-specific models. This technique leads to significant improvements across all target metrics in side-by-side human evaluation, enabling the model to achieve a high level of realism, text alignment, and compositional quality.

Table 4: Parameters for different stages of Kandinsky 5.0 models pre-training. Stages: LR – low resolution, MR – medium resolution, HR – high resolution.

Model & Training stage	**Training steps**	**Batch size**	**Learning rate**	**Weight decay**	**Resolution probabilities**
Image Lite (LR)	400k	8000	1e-4	0.0	256×256: 0.224, 192×320: 0.11, 320×192: 0.332, 160×352: 0.005, 352×160: 0.012, 224×288: 0.144, 288×224: 0.173
Image Lite (MR)	200k	4000	4.0e-05	0.0	512×512: 0.3152, 640×384: 0.5301, 384×640: 0.1547
Image Lite (HR)	200k	2000	3.0e-05	0.0	1024×1024: 0.183, 1408×640: 0.017, 640×1408: 0.010, 1280×768: 0.315, 768×1280: 0.115, 1152×896: 0.175, 896×1152: 0.185
Image Editing (MR)	200k	4000	4.0e-05	0.001	512×512: 0.479, 384×640: 0.4, 640×384: 0.121
Image Editing (HR)	110k	2000	2.0e-05	0.001	1024×1024: 0.297, 640×1408: 0.012, 1408×640: 0.005, 768×1280: 0.258, 1280×768: 0.093, 896×1152: 0.136, 1152×896: 0.199
Video Lite (T2I, LR)	220k	8192	1e-4	0.0	256×256: 0.25, 256×384: 0.22, 384×256: 0.53
Video Lite (T2V + I2V, LR)	10k	2662	6.0e-05	0.001	256×256: 0.25, 256×384: 0.22, 384×256: 0.53
Video Lite (MR, 5 seconds)	50k	1331	3.0e-05	0.001	512×512: 0.28, 512×768: 0.25, 768×512: 0.47
Video Lite (MR, 10 seconds)	10k	665	3.0e-05	0.001	512×512: 0.28, 512×768: 0.25, 768×512: 0.47
Video Pro (T2I, LR)	200k	16384	1e-4	0.0	256×256: 0.25, 256×384: 0.22, 384×256: 0.53
Video Pro (LR)	200k	2662	1e-4	0.0	256×256: 0.25, 256×384: 0.22, 384×256: 0.53
Video Pro (MR)	45k	1331	3.0e-05	0.001	512×512: 0.28, 512×768: 0.25, 768×512: 0.47
Video Pro (HR, 5 seconds)	22.5k	665	2.0e-05	0.001	1024×1024: 0.183, 1408×640: 0.017, 640×1408: 0.01, 1280×768: 0.315, 768×1280: 0.115, 1152×896: 0.175, 896×1152: 0.185
Video Pro (HR, 10 seconds)	10k	333	2.0e-05	0.001	1024×1024: 0.183, 1408×640: 0.017, 640×1408: 0.01, 1280×768: 0.315, 768×1280: 0.115, 1152×896: 0.175, 896×1152: 0.185

6.4.2 Video Generation

For video model, we investigated two approaches: standard fine-tuning and model souping. The video SFT dataset consists of approximately 2.8 thousand videos and 45 thousand images that underwent a strict manual selection process based on multiple criteria. For more details on the video data collection for SFT refer to Section 4.6.

During standard fine-tuning of the Kandinsky 5.0 Video Lite model, we observed overfitting after approximately 10 thousand steps. The best results for the 5-second model were achieved using an Exponential Moving Average (EMA) checkpoint after 10 thousand iterations.

We implement the souping method for video similarly to the approach for images: the data is partitioned into nine thematic domains using a VLM. We train a separate model on each domain and then perform uniform averaging of their weights. Although the individual domain models are prone to overfitting and generating artifacts, their averaged version demonstrates high visual quality and generation stability. In our internal side-by-side evaluations, the model obtained through souping outperformed the standard SFT model in visual quality, motion consistency, and absence of artifacts, while maintaining equivalent text alignment. Examples of comparing the generations before and after the SFT stage are presented in Figure 27.

6.5 Distillation

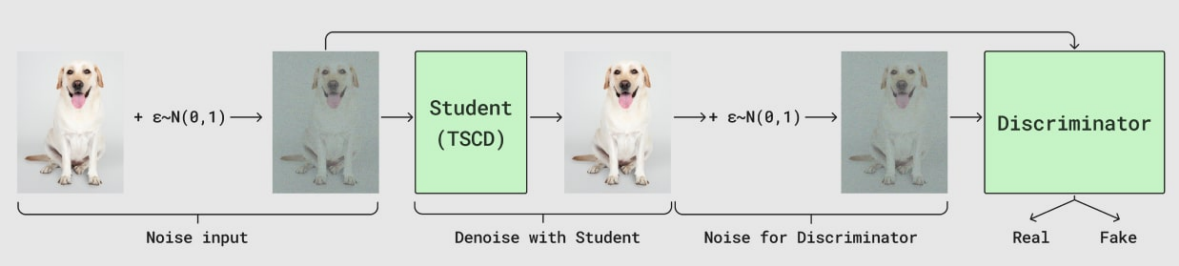


Figure 20: Adversarial post-training for diffusion distilled model.

We present two distinct classes of distilled models for video generation, each optimized for a different balance of sampling speed and visual fidelity:

1. **Guidance Distilled Model.** The first model variant is produced via Classifier-Free Guidance (CFG) Distillation [42]. This process directly distills the sampling trajectory of a base model, conditioned on a CFG scale, into a model that requires fewer number of function evaluations (NFEs). Starting from a base model requiring 100 NFEs, we applied CFG distillation with a null prompt and an optimal CFG scale of 5.0. This yielded a distilled model capable of generating samples of comparable quality in just 50 NFEs – a 2x speedup.

   According to our internal side-by-side human evaluation, the CFG Distilled model preserves the semantic composition, texture detail, motion coherence, and prompt alignment of the original model without perceptible degradation.

2. **Diffusion Distilled Model.** The second variant is a more aggressively distilled model, produced through a two-stage pipeline designed to maximize inference speed. This version is the basis for the **Kandinsky 5.0 Video Lite Flash** and **Kandinsky 5.0 Video Pro Flash** models.

   (a) **Initial CFG Distillation:** We first applied Trajectory Segmented Consistency Distillation (TSCD) [43] to the CFG-checkpoint, distilling it into a model requiring only 16 NFEs. This stage prioritizes a significant reduction in inference cost.

(b) **Adversarial Post-Training:** The model resulting from the first stage achieves high speed but exhibits a deficit in visual fidelity. To address this, we performed a second-stage refinement using an adversarial training framework inspired by LADD [98] and (Figure 20).

The adversarial training was conducted using a Hinge loss objective. We used the RMSprop optimizer [99] for both the generator (student model) and the discriminator, with learning rates of 1e-6 and 1e-4, respectively. Gradient clipping with a maximum norm of 1.0 was applied to stabilize training.

A critical component for stabilizing our adversarial training was the application of a stochastic perturbation to images before feeding them to the discriminator. Specifically, both real and generated images are perturbed with Gaussian noise, with the noise level sampled from a Logit-Normal(-4, 1) timestep distribution. We found that this "re-noising" strategy, as also explored in [100], significantly improves training stability compared to feeding clean images to the discriminator. Furthermore, this approach eliminates the need for an R1 gradient penalty, simplifying the training objective and reducing computational overhead.

6.6 RL-based Post-training for Image Generation

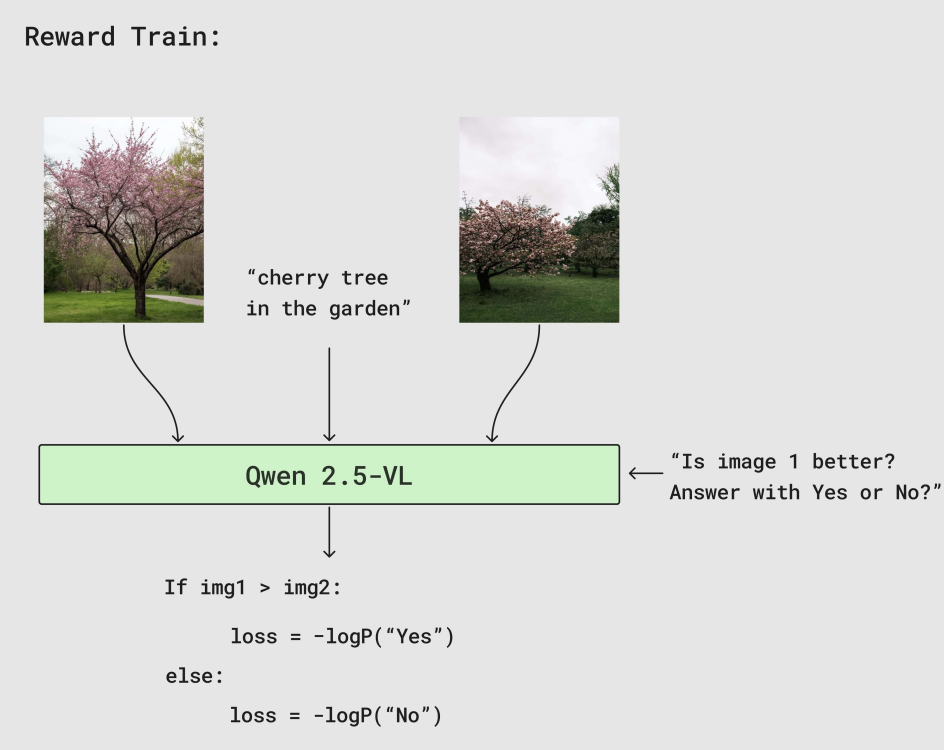


Figure 21: Reward model training.

To further enhance the visual quality and realism of generated images, we apply RL-based post-training techniques, which consist of two main parts – training a **reward model** (Figure 21) and **RL-based** fine-tuning stage of the image generation model with feedback from trained reward model (Figure 24).

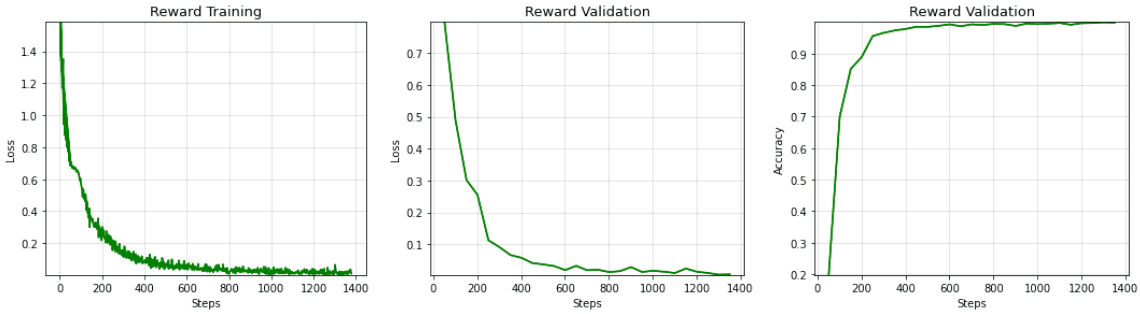


Figure 22: Training loss, validation loss and validation accuracy for reward model.

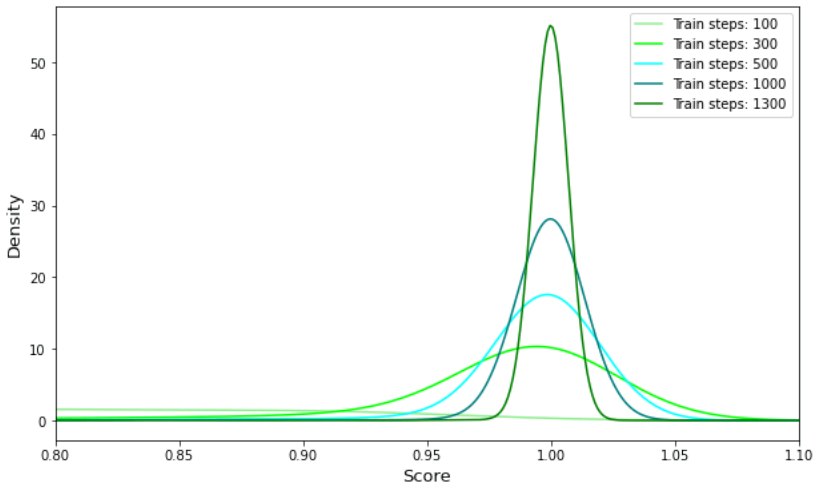


Figure 23: The scores distribution of the reward model on the validation set. As the number of steps increases, the distribution looks more like a delta function.

Reward Model Training

Data. We conduct experiments to train the reward model using real data collected for the supervised fine-tuning (SFT) stage (see Section 4.6), as well as generations from the Kandinsky 5.0 Image model after the pre-training and SFT stages. To form the dataset for reward model training, we used heuristic, that allowed us to skip the usual for RLHF [101] (Reinforcement Learning on Human Feedback) data annotation process and still get very good results from RL-based post-training stage. We supposed that by design image generated from pre-train checkpoint is worse, than image generated from SFT checkpoint, which is worse, than the real, not generated, image from SFT set. This way, we collected our (x_w, x_l, y) samples for reward model training.

Relative Reward Training. Our approach is based on the Reward Dance method [102]. For the reward, we are training a visual-language model, which we initialize from Qwen 2.5VL-7B [73]. The reward model takes two annotated images with a text description as input, and then returns whether the first image is better or not. The model is trained to return "Yes", when the first image is better and "No", when the first image is worse. The value, that reward returns, that will be used as feedback for the generative model is the probability of the token "Yes" as the answer for whether the image on position 1 is better, i.e.: $R(x_1, x_2, y) = P(\text{"Yes"}|x_1, x_2, y)$.

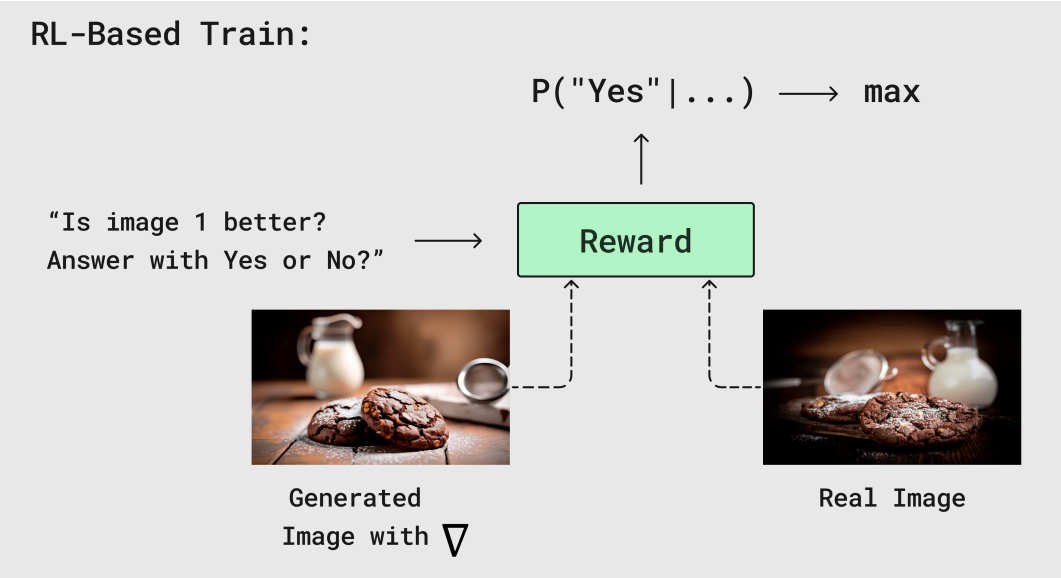


Figure 24: Fine-tuning of the image generation model with feedback from Reward Model.

Figure 22 shows the training loss for the reward model, as well as the loss and accuracy values on the validation set for the reward outputs. Figure 23 shows a plot of the reward model scores for different numbers of training steps on the validation set. It can be seen that as the number of steps increases, overfitting occurs and the plot increasingly resembles a delta function. During training, we monitor this plot and the standard deviation value σ for Gaussian kernel density estimation to avoid overfitting. As a result, **we selected the reward model trained for 1300 steps**.

RL-based post-training

For RL-based tuning we utilize Direct Reward Fine-Tuning (DRaFT) method [103], specifically we use the DRaFT-K version (Algorithm 1 from the original paper). Specifically, we generate an image using our model that has undergone the SFT stage and only backpropagate gradients through the last few generation steps. For the second image, we take a real image from the SFT dataset and maximize the probability that the generated image will be chosen as better by the reward model. We also use the Kullback–Leibler divergence between the distributions of images generated by the SFT-stage model p_{SFT} and the model being trained p_{RL}. In the case of flow-matching models [3], the KL-divergence takes the following form:

$$\mathrm{KL}(p_{RL} \parallel p_{SFT}) = \sum_t \|v_{RL}(x_t, t) - v_{SFT}(x_t, t)\|^2,$$

where v_{RL}, v_{SFT} are the outputs of the image generation model in the trainable (p_{RL}) and frozen post-SFT (p_{SFT}) versions respectively, x_t is the denoising result after the first $T - t$ steps of the p_{RL} model, and the sum is taken only over those final generation steps t for which gradients are backpropagated through the p_{RL} model. The final loss looks as follows:

$$L = L_{RL} + \beta_{KL} \cdot \mathrm{KL}(p_{RL} \parallel p_{SFT}),$$

where $L_{RL} = 1 - R(x_\nabla, x_{\text{real}}, y)$ is RL loss term, x_∇ is the image generated with gradients for the last K steps, as proposed by DRaFT-K algorithm from paper [103], x_{real} is the real image from SFT set, that corresponds to the text prompt y. For text-to-image model, optimal β_{KL} equals to

2e-2 and $K = 10$ for DRaFT-K. The training plots for the generation model during the RL-based training are shown in Figure 25.

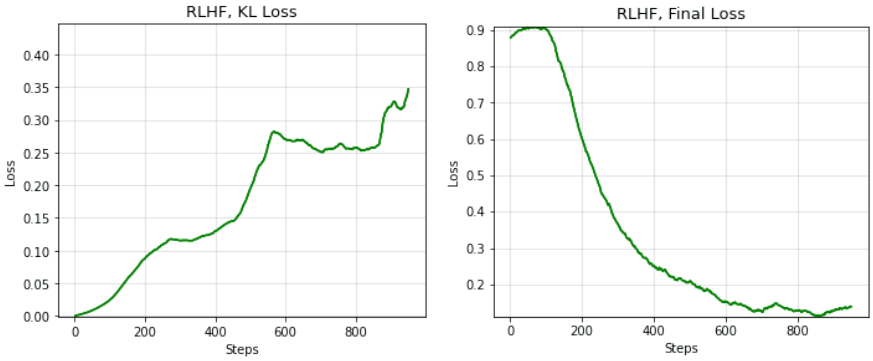


Figure 25: KL loss and final loss for RL-based post-training of image model.

Other Experiments. We also investigated other approaches, including the use of a differentiable absolute reward [103, 104, 105], as well as using two generated images for relative reward – with and without gradient backpropagation on the final steps, selecting the best image from N generated ones as the reference image as proposed in [102]. We tried using two generations as input to the reward model, but eventually in our experiments we found that using a relative reward with a real image from the SFT dataset as a reference sample results in the best quality.

7 Optimizations

7.1 VAE Encoder Acceleration

We have significantly accelerated the Hunyuan Video VAE encoder [26] through code optimization and by replacing several operations with more efficient equivalents. This resulted in an average 2.5× speed-up in the encoding process without requiring any additional training. Alongside the performance gains, we also improved video reconstruction quality: we found that temporal blending of tiles reduces the quality, fewer tiles means less quality loss.

Key enhancements include:

- **Optimized Tiling.** Since Kandinsky 5 supports a limited set of input spatial resolutions, we determined optimal tile sizes. This approach efficiently utilizes GPU memory, improves GPU utilization, and minimizes artifacts at tile boundaries blending;
- **Integration of `torch.compile`.** We added support for `torch.compile` and identified a configuration that provides the optimal balance between compilation time and inference speed.

7.2 CrossDiT Optimization

The primary computational bottleneck in image and video generation is the diffusion transformer, making its optimization especially critical. In addition to diffusion step distillation, described in Section 6.5, we apply the following optimization techniques:

1. Detailed performance profiling and careful refactoring of the inference code to eliminate GPU idle time and achieve maximum efficiency with `torch.compile`.

2. Caching diffusion steps using the MagCache [106] method. In our experiments, this technique delivered a 46% speedup with no visible quality drop.
3. For SD resolution and generation durations up to 5 seconds, we employ either Flash Attention [107, 108] or Sage Attention [109], depending on the GPU hardware.
4. For longer generation times (over 5 seconds) or HD resolution, we use our custom NABLA method (see Section 5.4), which accelerates inference by a factor of 2.7 compared to full attention.

Table 5 summarizes the computational performance measurements for all models in the Kandinsky 5.0 family. Memory requirements can be decreased by using quantized text encoder.

Model	Frame Number	Resolution	NFE	Generation Time, s	GPU Memory, GB (with offloading)
Video Lite 5s	121	512×768	100	139	21
Video Lite 10s	241	512×768	100	224	21
Video Lite 5s Flash	121	512×768	16	35	21
Video Lite 10s Flash	241	512×768	16	61	21
Video Pro 5s	121	512×768	100	560	47
Video Pro 10s	241	512×768	100	1158	51
Video Pro 5s Flash	121	512×768	16	123	47
Video Pro 10s Flash	241	512×768	16	242	51
Video Pro 5s	121	768×1280	100	1241	53
Video Pro 10s	241	768×1280	100	3218*	68
Video Pro 5s Flash	121	768×1280	16	235	53
Video Pro 10s Flash	241	768×1280	16	576*	68
Image Lite	1	1024×1024	100	13	17

Table 5: Kandinsky 5.0 computational performance evaluation. All measurements are performed using single 80 GB H100 GPU.
*Evaluation is performed with offloading enabled.

7.3 Training

To estimate the training step duration and GPU memory consumption during the model training, we developed a mathematical model that relates these quantities to the main parameters of the training and model configuration.

7.3.1 Training Step Estimation.

The training step time is estimated using the following expression:

$$Step = \frac{d}{d_0} \cdot \frac{S}{S_0} \cdot \left(9 + 14 \cdot \frac{S}{S_0} + 6 \cdot \frac{d}{d_0}\right) \cdot L \cdot B, \tag{1}$$

where:

- d_0 and S_0 are constants: $d_0 = 1792$ (hidden dimension for the 2B model) and $S_0 = 256 \times 384 \times 31$ (reference video resolution);
- d is the hidden dimension of the model being analyzed;

- S is the video resolution;
- L is the number of transformer blocks;
- B is the batch size.

7.3.2 GPU Memory Consumption.

The GPU memory consumption is estimated as:

$$Memory = 12L\frac{(9d_td + 8d^2 + 2d_fd)}{N} + \max\left(4L\frac{(9d_td + 8d^2 + 2d_fd)}{N},\ 2S(Ldo + 18d + 2d_f)\right), \quad (2)$$

where:

- N is the number of GPUs;
- S is the sequence length;
- L is the number of blocks;
- d is the hidden dimension ($hidden\_dim$);
- d_t is the time dimension ($time\_dim$);
- d_f is the feed-forward dimension (ff_dim);
- $o = 0$ if activations are offloaded, otherwise $o = 1$.

This model enables fast analytical estimation of the scaling behaviour of the training pipeline with respect to model size, resolution, and hardware configuration and according to our experiments well fit real training step time and memory consumption. We used theoretical estimation for experiments planning, correct batch and parallel setup selection.

8 Results

8.1 Quality Progress

We have seen significant qualitative improvements in terms of visual quality, realism, detail, and video dynamics following our multi-step training procedure. Figures 26 and 27 show a side-by-side comparison of the generation results for images (after pre-training, SFT and RL stages) and videos (after pre-training and SFT stages), respectively.

8.2 Human Evaluation

The Side-by-Side (SBS) evaluation is conducted on the **Elementary platform**[20]. Typically, a project is annotated by approximately 20 annotators who are not novices. The evaluation is split into two main phases:

1. **Visual Quality Assessment**: Annotators evaluate generated visuals (images or videos) without access to the prompt to prevent bias.
2. **Prompt Following Assessment**: After completing the visual assessment, annotators evaluate how well the generations adhere to the given prompt.

[20] https://elementary.center

(a) Pre-training

(b) SFT

(c) RL post-training

A graceful red fox with a fluffy tail stands in a forest clearing early in the morning. The rays of the rising sun gently illuminate its silky fur with golden hues. The fox looks attentively into the distance, ears alertly raised up. In the background, dense trees covered with morning dew create an atmosphere of mystery and tranquility. Detailed rendering of fur, eyes, and paws emphasizes the natural beauty of the animal. Realistic image with high resolution, focus on light and shadow transmission. Photographic accuracy, natural surroundings, no extraneous objects.

(d) Pre-training

(e) SFT

(f) RL post-training

Close-up portrait of James Bond: a rugged face, confident gaze from cold blue eyes, perfectly styled dark hair, three-day stubble adds brutality to his image. He wears a classic black tuxedo with a white shirt and bow tie, an elegant tie clip, and premium wristwatch. Behind him is a night cityscape with skyscraper silhouettes and reflections of lights on wet pavement. The composition emphasizes an atmosphere of mystery and danger. Realistic depiction with deep shadows and sharp light accents, cinematic post-processing, high-quality photography.

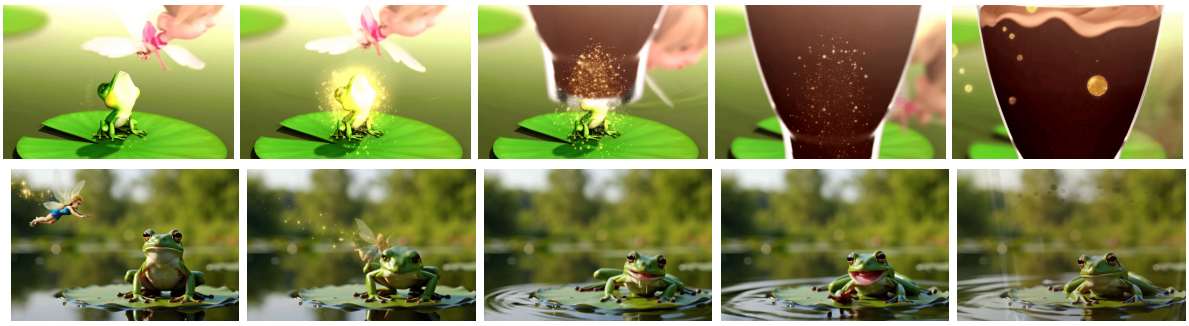
(g) Pre-training

(h) SFT

(i) RL post-training

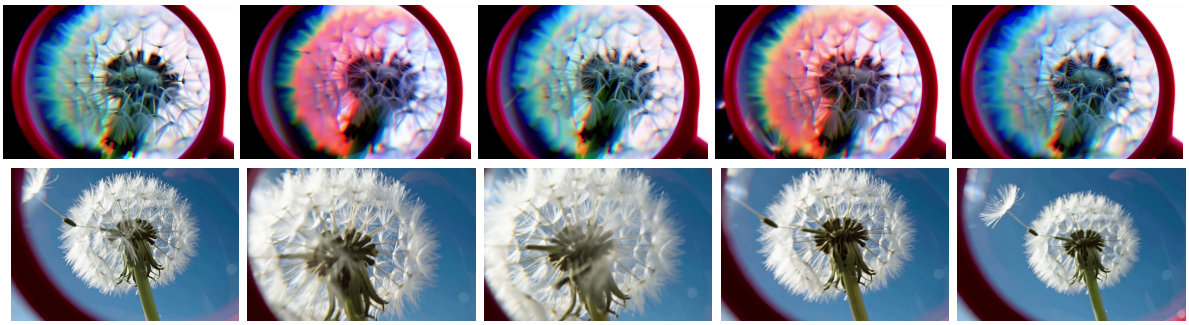
Frame in dynamic motion: a small bear cub confidently stands on a brightly colored skateboard, skillfully maneuvering among autumn leaves. The bear wears a colorful helmet and protective accessories that reflect safety and fun. Behind him is a picturesque park landscape with multicolored trees illuminated by soft sunlight. The animal's fur, board texture, and leaves are detailed to create a sense of realism. The artistic style combines elements of hyperrealistic photography and cute cartoon accents, emphasizing both naturalness and playfulness. The frame is filled with positive mood and carefree childhood atmosphere.

Figure 26: Visual quality progress for text-to-image generation after different training stages. From left to right: pretraining, SFT, RL-based fine-tuning.



(a) A small green frog sits on a lily pad in a calm pond, its eyes wide and curious. A gentle kiss lands on its forehead from a passing fairy, sending a shimmering glow across its body. The frog begins to shimmer and dissolve, its form transforming into a rich swirl of chocolate and milk. Bubbles rise as the once-living creature fully becomes a frothy, delicious chocolate milkshake in a glass. The transformation is magical and fluid, with golden sparkles floating around as the final product glistens under soft sunlight.

(b) A vibrant bird crafted entirely from fresh, juicy oranges suddenly bursts forth from a large, ripe orange. Its wings, made of plump citrus slices, flap rapidly as it soars into the air, leaving behind a trail of sparkling orange juice droplets. Sunlight glimmers off the bird's glossy orange feathers, casting a warm, golden glow around it. The bird twists and turns gracefully, its beak pecking playfully at floating citrus peels. As it flies higher, the background shifts to a bright orange sunset, completing the surreal and lively scene.



(c) A dynamic, extremely macro closeup view of a delicate white dandelion, its feathery seeds gently swaying in a soft breeze. The scene is seen through the curved, highly reflective surface of a large red magnifying glass, distorting and refracting the light in colorful patterns. The dandelion's tiny details—each seed, petal, and dewdrop—are magnified and illuminated dramatically. A slight movement of the magnifying glass causes the image to shift and shimmer, creating a mesmerizing visual effect. As the wind picks up, the dandelion's seeds begin to lift and drift, captured in slow motion through the magnifying lens.

Figure 27: Visual quality progress for text-to-video generation. **The results after pre-training stage (the top example from the pair) and after SFT (the bottom example).**

Generations from different models are mixed and displayed in random order (left/right). An overlap of 5 is used for annotation consistency.

For video generation, most SBS evaluations are performed on the **Moviegen**[31] (1003 prompts).

8.2.1 Prompt Following

Evaluators assessed prompt adherence using the criteria below. For each criterion, annotators selected one of the following judgments:

Judgment Options:

Choice	Description
Model 1 Better	Model 1 better satisfies the prompt requirement
Model 2 Better	Model 2 better satisfies the prompt requirement
Both Fully Correct	Both models fully satisfy the prompt requirement
Both Fully Incorrect	Neither model satisfies the prompt requirement
Equally	Both models partially satisfy the requirement, with no clear advantage

Evaluation Criteria:

Criterion	Description
Object Presence (Count)	Number of unique objects from the prompt correctly generated
Object Quantity	Whether the correct number of each object is present
Object Properties	Accuracy of object attributes (e.g., color, size, shape)
Object Placement	Correct spatial relationships and relative positioning of objects
Action Presence (Count)	Number of unique actions from the prompt successfully realized
Action Properties	Accuracy in execution, timing, and dynamics of described actions

Aggregation Formula:

- Model 1 Score = Model 1 Better + Both Fully Correct + Equally / 2
- Model 2 Score = Model 2 Better + Both Fully Correct + Equally / 2

Scores are normalized and averaged across prompts to compute an overall **Prompt Following** score, with detailed results visualized per criterion.

8.2.2 Visual Quality

Annotators evaluate the following aspects:

No.	Description
1	Composition: Framing, balance, and visual structure
2	Lighting: Realism and consistency of illumination
3	Color and Contrast: Accuracy and harmony of color palette and contrast levels
4	Object and Background Distinctness: Clarity of foreground/background separation
5	Frame Transition Smoothness: Temporal coherence between consecutive frames
6	Dynamism: Energy, motion intensity, and scene activity
7	Realism of Object Motion: Physical plausibility and naturalness of movement
8	Face Generation Consistency: Temporal stability of facial features (if applicable)
9	Overall Impression: Holistic aesthetic quality
10	Number of Artifacts: Visual defects (e.g., distortions, glitches, blur)
11	Number of Semantic Breaks: Unintended transformations or content shifts (per KandiVideoPrompts)

Judgment Options (Criteria 1–9):
Model 1 Better
Model 2 Better
Equally
Model 1 Better (Unconfident)
Model 2 Better (Unconfident)

Judgment Options (Criteria 10–11):
Model 1 Better
Model 2 Better
Both Fully Correct
Equally

Aggregation Formula (Criteria 1-9):

- Model 1 Score = Model 1 Better + Model 1 Better (Unconfident) / 2 + Equally / 2
- Model 2 Score = Model 2 Better + Model 2 Better (Unconfident) / 2 + Equally / 2

Aggregation Formula (Criteria 10-11):

- Model 1 Score = Model 1 Better + Both Fully Correct + Equally / 2
- Model 2 Score = Model 2 Better + Both Fully Correct + Equally / 2

Results are averaged into:

- Overall **Visual Quality** score (Criteria 1-5, 9-10).
- Overall **Dynamism and Motion Quality** score (Criteria 6-8, 11).

Averaging is done by percentage due to uneven distribution of ratings across criteria.

8.2.3 Kandinsky 5.0 Video Lite vs. Sora

We conducted a side-by-side (SBS) human evaluation study comparing **Kandinsky 5.0 Video Lite** and **Sora** across six key dimensions of video generation quality on the full MovieGen benchmark. For each criterion, raters were presented with paired outputs and asked to select the better-performing model or indicate a tie. The results are aggregated over a representative sample of prompts and visualized as stacked bar charts, where each segment reflects the proportion of judgments favoring one model, both, or neither. See Figure 28 for details.

8.2.4 Kandinsky 5.0 Video Lite vs. Wan Models

We conducted a simplified side-by-side (SBS) evaluation of our **Kandinsky 5.0 Video Lite** in Text-to-Video mode against three models of Wan series — **Wan 2.1 14B**, **Wan 2.2 5B**, and **Wan 2.2 A14B** — on the MovieGen benchmark, comparing performance across three key dimensions: Prompt Following, Visual Quality, and Motion Dynamics. Expert raters assessed paired video outputs per prompt, with results aggregated into preference scores per criterion.

Results, visualized in Figure 29 reveal a consistent pattern:

- **Visual Quality** and **Motion Dynamics** consistently favor **Kandinsky 5.0 Video Lite** across all comparisons, with clear advantages in aesthetic coherence, object realism, and temporal fluidity.
- **Prompt Following** shows stronger performance from Wan models, particularly **Wan 2.2 A14B** and **Wan 2.1 14B**, which better capture fine-grained semantic details and action specifications.

While Wan variants demonstrate superior alignment with textual prompts, Kandinsky 5.0 Video Lite maintains a decisive edge in perceptual quality and motion naturalness — suggesting a clear trade-off between semantic fidelity and visual fluency. The gap in prompt adherence is moderate and varies by scenario, indicating that Kandinsky 5.0 Video Lite remains a compelling choice for applications prioritizing visual output over precise instruction following.

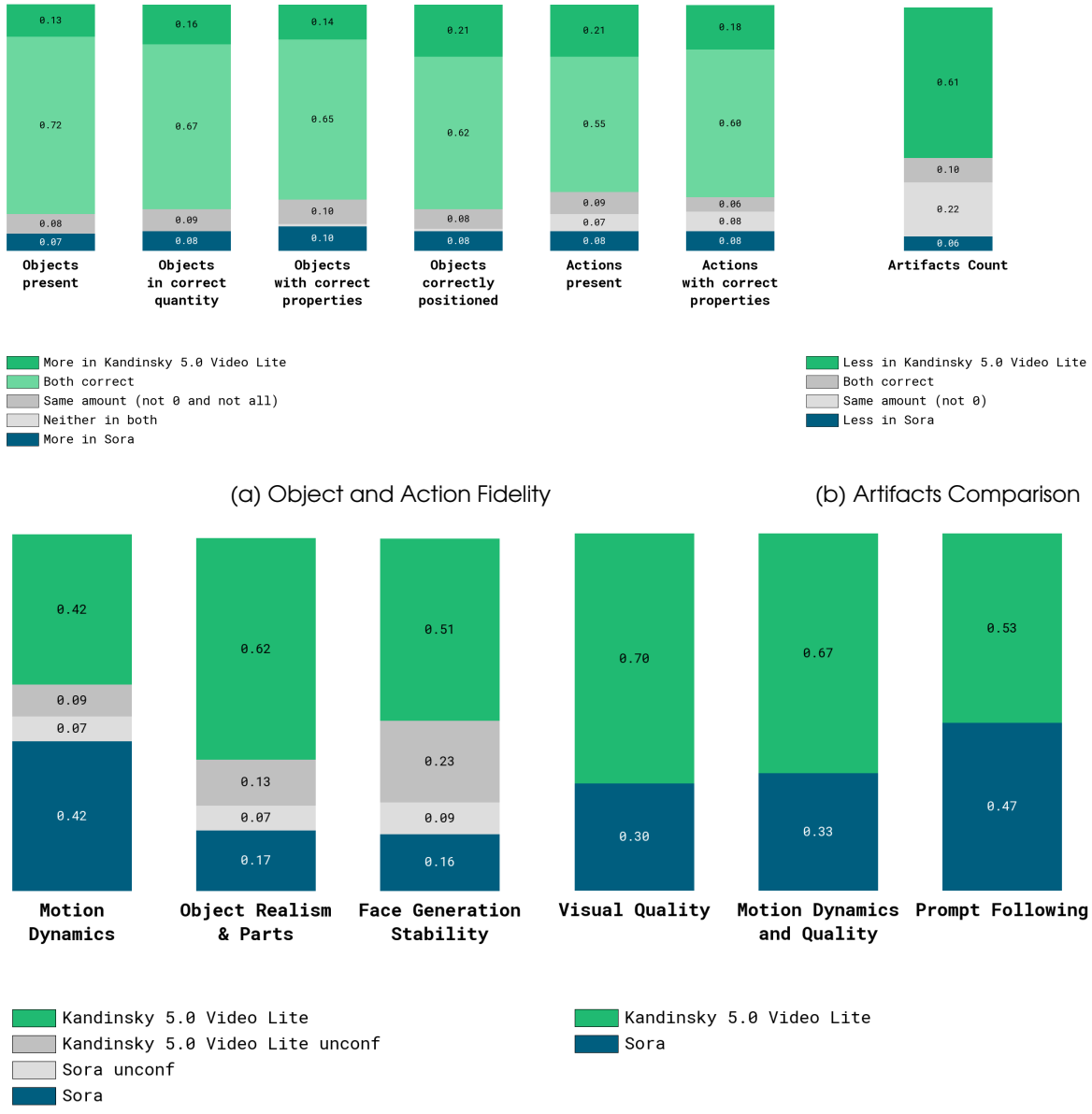
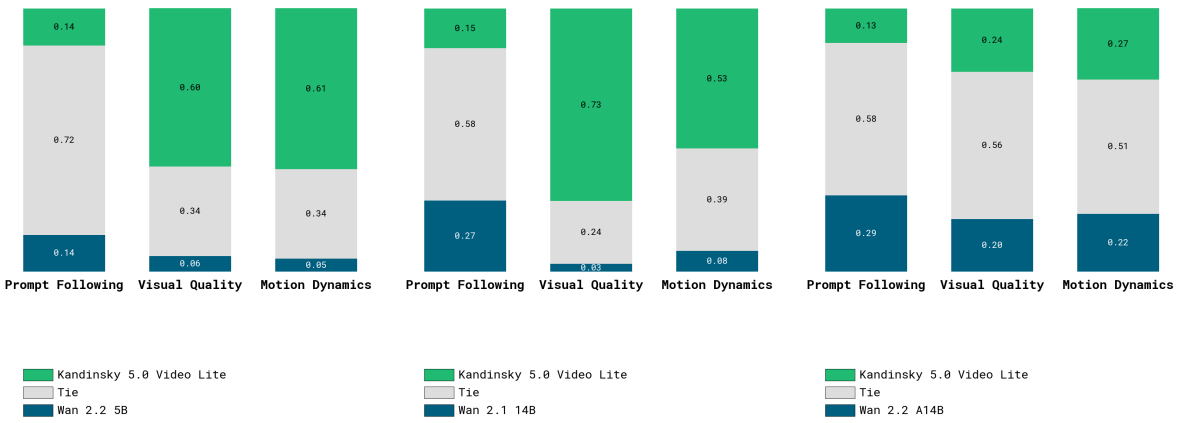


Figure 28: Side-by-side (SBS) human evaluation of **Kandinsky 5.0 Video Lite** versus **Sora** on the full MovieGen benchmark. We collected ~65K pairwise judgments from 44 trained raters (239 person-hours) across 1,002 prompt-video pairs (with 5-way overlap per item). Each subplot shows the distribution of preferences: green segments indicate cases where **Kandinsky 5.0 Video Lite** was rated higher, blue — where Sora was preferred, and intermediate shades represent ties or neutral outcomes. Inter-rater agreement is approximately 71%.



(a) Comparison with Wan 2.2 5B (b) Comparison with Wan 2.1 14B (c) Comparison with Wan 2.2 A14B

Figure 29: Kandinsky 5.0 Video Lite outperforms Wan models in Visual Quality and Motion Dynamics.

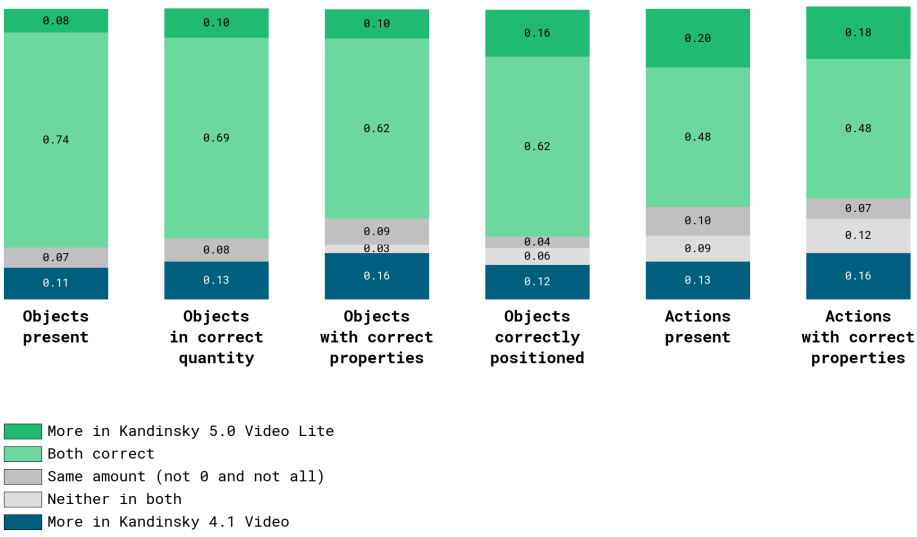
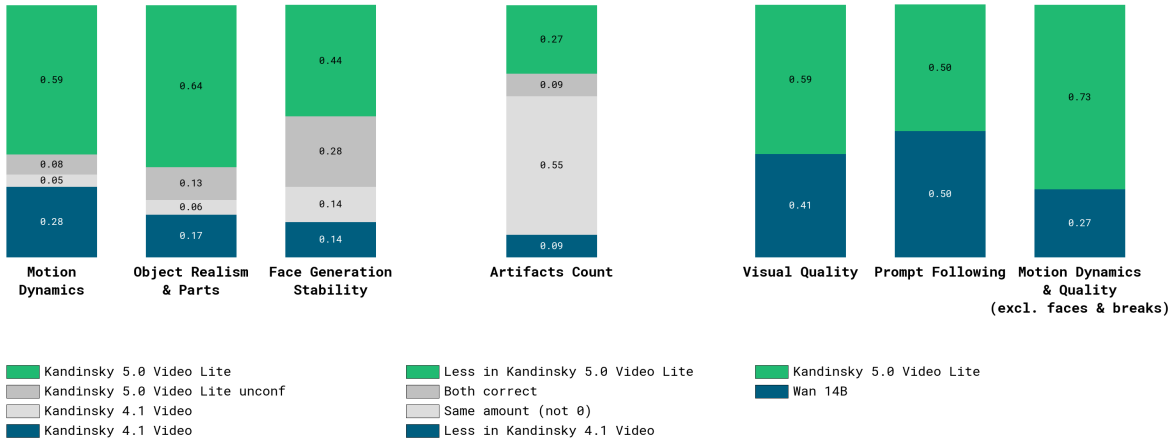


Figure 30: Object and Action Presence, Quantity, Properties, and Positioning. **Kandinsky 5.0 Video Lite** outperforms **Kandinsky 4.1 Video** in action-related metrics — both in the presence of actions and their alignment with prompt semantics — and shows better object positioning. Conversely, **Kandinsky 4.1 Video** is slightly preferred in basic object presence, quantity, and attribute fidelity, though both models in most cases produce correct outputs in these categories.



(a) Dynamics and object motion realism. (b) Artifact comparison (c) Key Video Quality Dimensions

Figure 31: (a) Motion Dynamics: **Kandinsky 5.0 Video Lite** is preferred in 59% of cases, **Kandinsky 4.1 Video** in 28%, with 13% undecided — indicating substantial improvement in temporal coherence and fluidity. (b) Artifacts: In 55% of comparisons, both models exhibit similar artifact levels; **Kandinsky 5.0 Video Lite** has fewer artifacts in 27% of cases, while Kandinsky 4.1 Video does so in only 9%. This confirms significant artifact reduction in the newer version. (c) Overall Quality Dimensions: **Kandinsky 5.0 Video Lite** leads decisively in Visual Quality (0.59) and Motion Dynamics (0.73), while matching **Kandinsky 4.1 Video** in Prompt Following (0.50). The upgrade delivers consistent gains across all core metrics, especially in motion and aesthetics.

8.2.5 Kandinsky 5.0 Video Lite vs. Kandinsky 4.1 Video

We also conducted a comprehensive side-by-side (SBS) human evaluation comparing **Kandinsky 5.0 Video Lite** and our previous model **Kandinsky 4.1 Video** across key dimensions of video generation quality on the full MovieGen benchmark. Evaluations were performed by trained raters using paired outputs per prompt, with judgments aggregated over a representative sample of video generations. Results are visualized in stacked bar charts, where green segments indicate preference for **Kandinsky 5.0 Video Lite**, blue for **Kandinsky 4.1 Video**, and intermediate shades denote ties or neutral outcomes. See Figures 30–31 for detailed breakdowns.

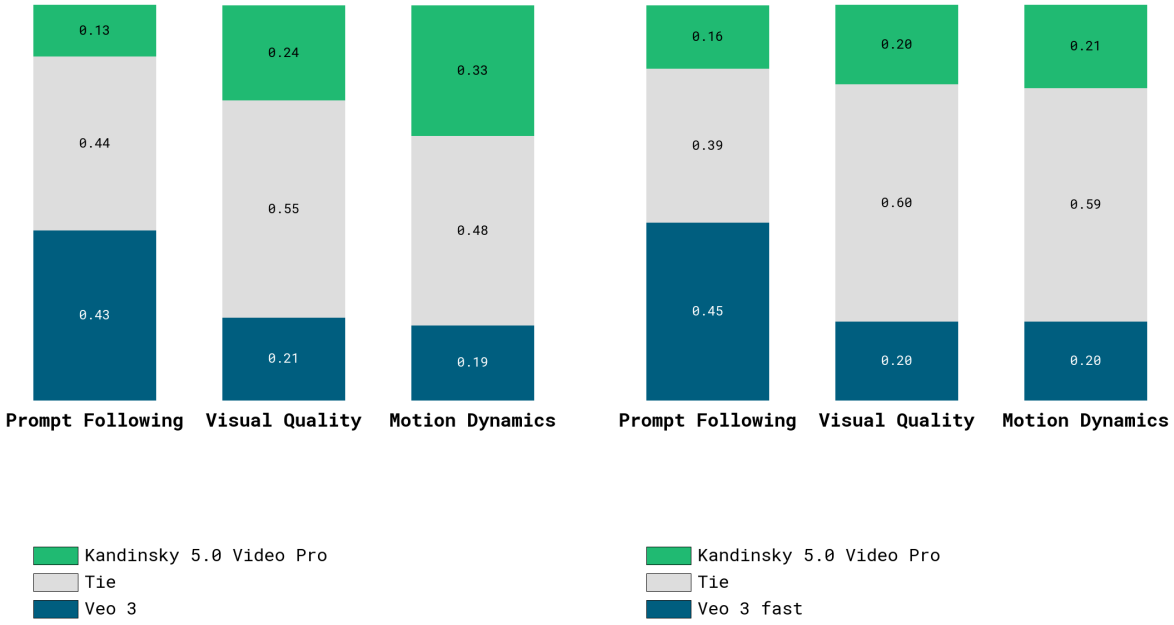
Across all metrics, **Kandinsky 5.0 Video Lite** demonstrates marked improvements over **Kandinsky 4.1 Video**, particularly in motion dynamics, object realism, artifact suppression, and semantic accuracy. The new version excels in generating coherent, visually rich sequences with higher fidelity to prompts, while maintaining strong performance in face stability and component-level realism. These results confirm that **Kandinsky 5.0 Video Lite** model represents a substantial leap forward in video generation capability within the Kandinsky family.

8.2.6 Kandinsky 5.0 Video Pro vs Veo 3 and Veo 3 fast

We conducted a side-by-side (SBS) comparison of our Kandinsky 5.0 Video Pro text-to-video model with leading video generation models Veo 3 and Veo 3 Fast, using the MovieGen benchmark dataset. The evaluation focused on three key aspects: Prompt Following (how

accurately the generated video aligns with the input text description), Video Quality (including aesthetic appeal, visual coherence, and technical execution), and Motion Dynamics (the naturalness, smoothness, and realism of motion over time). Expert evaluators assessed the outputs based on these criteria. Results show that Veo 3 and Veo 3 Fast significantly outperform Kandinsky 5.0 Video Pro in Prompt Following, demonstrating superior understanding and fidelity to complex textual instructions. However, Kandinsky 5.0 Video Pro achieves higher scores in Video Quality and Motion Dynamics, delivering more visually compelling and dynamically coherent sequences.

We recognize the importance of precise prompt adherence and will prioritize further improvements in this area to close the gap, while continuing to leverage our strengths in visual and temporal realism. The results are presented in Figure 32.



(a) Comparison with Veo 3

(b) Comparison with Veo 3 fast

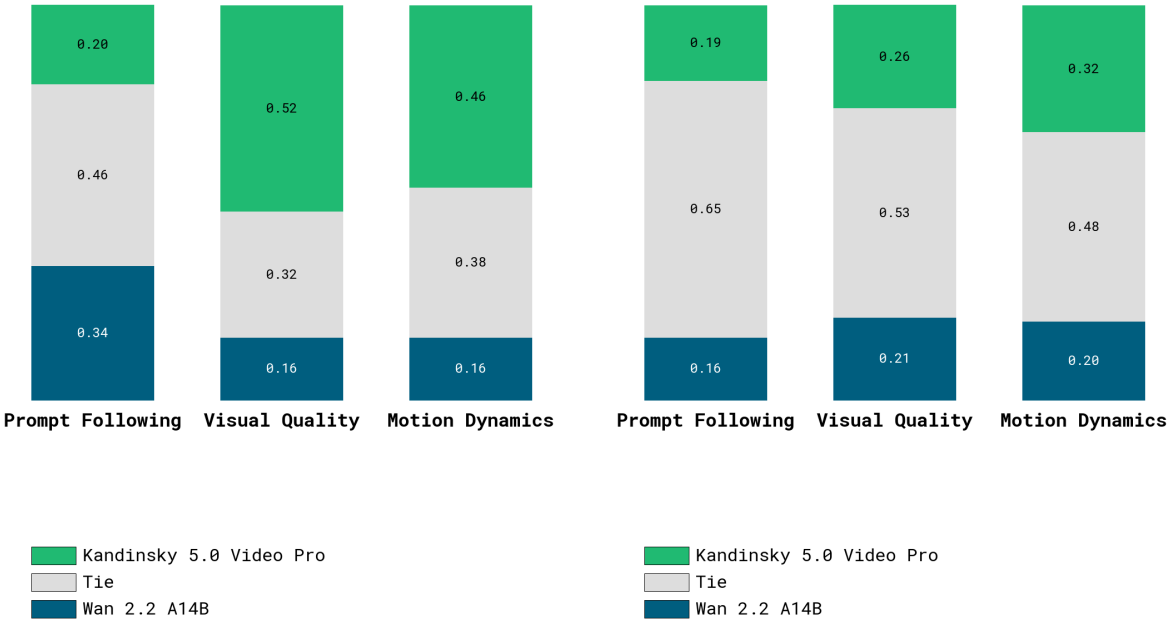
Figure 32: Kandinsky 5.0 Video Pro excelled in Visual Quality and Motion Dynamics, whereas Prompt Following remained a relative weakness compared to Veo 3 variants.

8.2.7 Kandinsky 5.0 Video Pro vs Wan 2.2 A14B

We also conducted a simplified side-by-side (SBS) comparison of Kandinsky 5.0 Video Pro with Wan 2.2 A14B in both text-to-video and image-to-video modes, evaluating performance on the same three criteria: Prompt Following, Visual Quality, and Motion Dynamics. The results are presented in Figure 33.

8.2.8 Kandinsky 5.0 Image Lite and Image Editing

We conducted an internal simplified side-by-side (SBS) comparison of our **Kandinsky 5.0 Image Lite** text-to-image model with popular models, specifically evaluating **FLUX.1 [dev]** and **Qwen-Image**.



(a) Text-to-Video mode

(b) Image-to-Video mode

Figure 33: Kandinsky 5.0 Video Pro outperforms Wan 2.2 A14B models in Visual Quality and Motion Dynamics.

The test was performed using a custom prompt dataset based on the PartiPrompts (P2) dataset [21], which was further expanded using the Giga Max large language model. Expert evaluators then compared the generated images from the models based on only two key parameters: **prompt following** (how accurately the image reflects the given text description) and **visual quality** (encompassing aesthetics, coherence, and technical execution). The results of this comparison are presented in Figure 34.

We also conducted an internal simplified side-by-side (SBS) comparison of our **Kandinsky 5.0 Image Editing** capabilities with popular image editing models, specifically evaluating **FLUX.1 Kontext [dev]** and **Qwen-Image-Edit-2509**.

The test utilized a Kontext Bench dataset [22] of image-instruction pairs. Expert evaluators compared the edited images from all models based on two key parameters: **instruction following** (how accurately the edit reflects the given instruction) and **visual quality** (assessing the coherence, realism, and aesthetic quality of the edited regions within the final image).

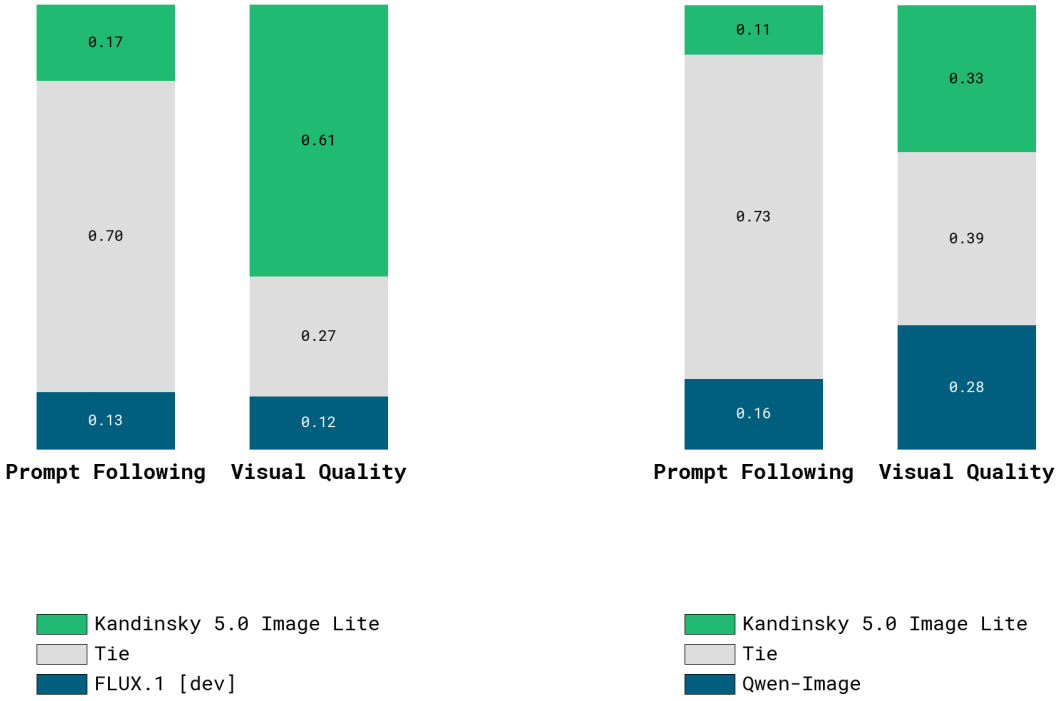
The results of this comparison are presented in Figure 35.

8.2.9 Kandinsky 5.0 Video Lite Flash

We conducted a simplified side-by-side (SBS) comparison between Kandinsky 5.0 Video Lite and Kandinsky 5.0 Video Lite Flash to assess the quality trade-offs associated with model distillation and optimization for speed. The evaluation covered both 5-second and 10-second generation lengths, using the same criteria as in prior comparisons: Prompt Following, Visual Quality, and Motion Dynamics.

[21] Original dataset available at: https://huggingface.co/datasets/nateraw/parti-prompts

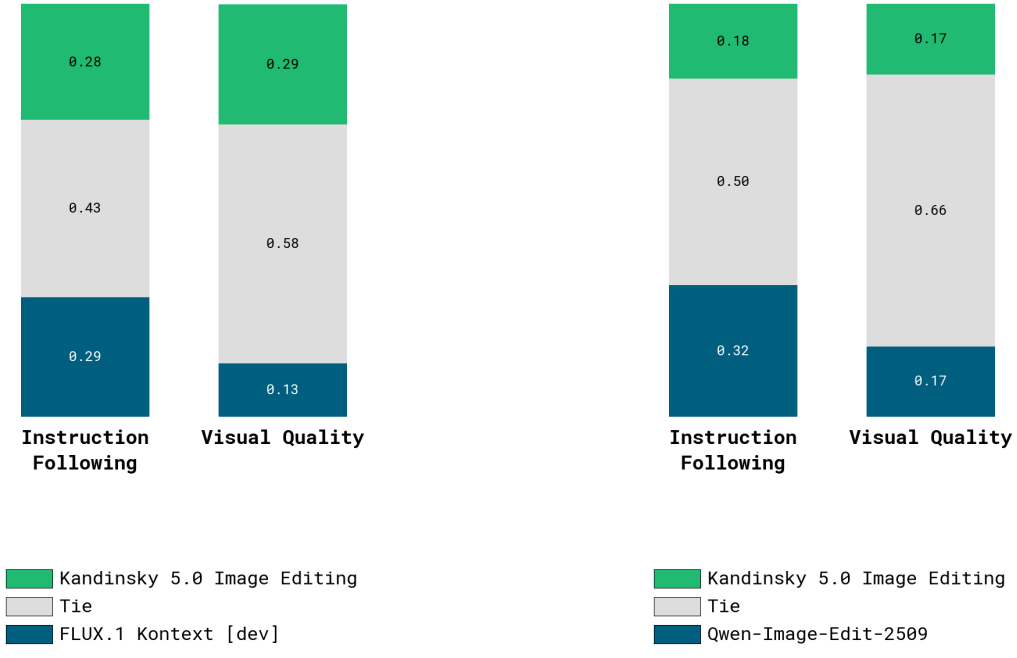
[22] https://huggingface.co/datasets/black-forest-labs/kontext-bench



(a) Comparison with FLUX.1 dev

(b) Comparison with Qwen-Image

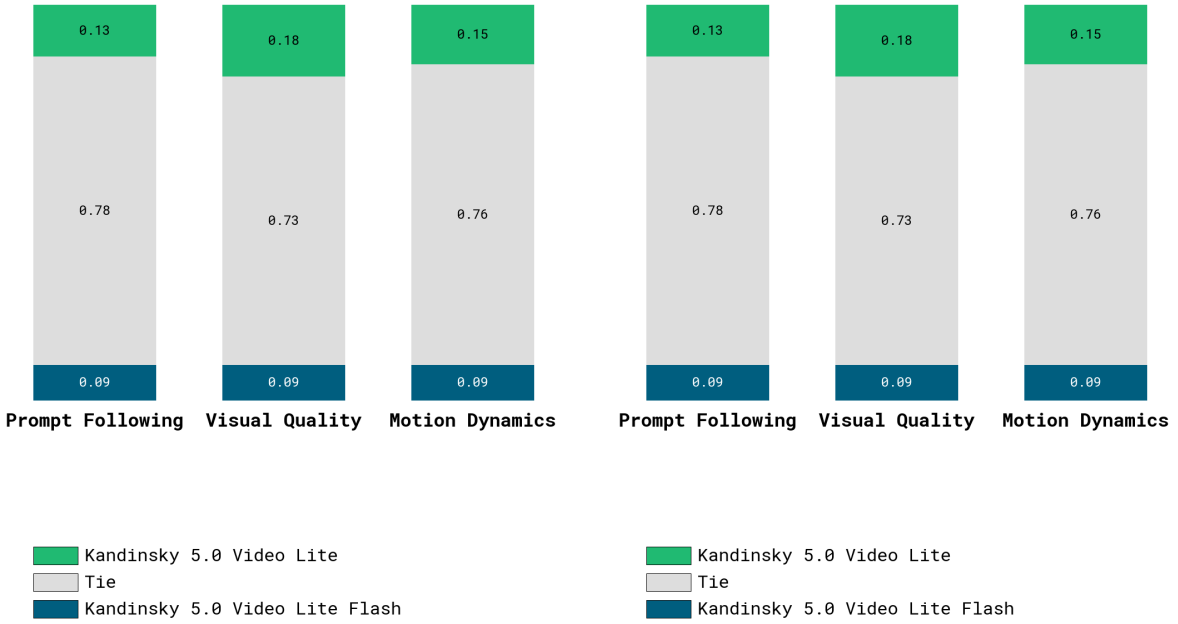
Figure 34: **Kandinsky 5.0 Image Lite** demonstrated stronger performance in Visual Quality while remaining competitive in Prompt Following.



(a) Comparison with FLUX.1 Kontext [dev]

(b) Comparison with Qwen-Image-Edit-2509

Figure 35: **Kandinsky 5.0 Image Editing** demonstrated competitive performance in against the evaluated models.



(a) 5-second model

(b) 10-second model

Figure 36: Evaluation Kandinsky 5.0 Video Lite Flash model against not distilled Kandinsky 5.0 Video Lite.

Results (Figure 36)indicate a measurable but generally moderate drop in performance for the Flash variant, particularly in fine detail rendering, temporal coherence, and handling of complex prompt semantics. This reflects the inherent trade-off between reduced computational cost, faster inference, and output fidelity.

However, the degradation is not critical and varies depending on the use case — lighter prompts and shorter durations show minimal perceptible difference. These findings suggest that Kandinsky 5.0 Video Lite Flash remains a viable option for applications where speed and efficiency are prioritized over maximum visual precision. The results support flexible deployment across the model family based on scenario-specific requirements.

9 Use cases

9.1 Text-to-Image

Kandinsky 5.0 Image Lite model line-up contains a powerful text-to-image generative model capable of producing highly diverse visual content with strong semantic alignment to input prompts. The model excels in photorealistic image synthesis, accurately rendering lighting, textures, and fine details that closely match real-world appearances. Beyond realism, it supports a wide range of artistic styles and media simulations — including oil and acrylic paintings, watercolor washes, pencil and charcoal sketches, and wax crayon drawings — allowing users to generate images that mimic specific traditional or digital art techniques. Additionally, the model can generate custom logos, typographic compositions, and even render legible text within images when explicitly prompted. Representative examples of these capabilities are illustrated in Figures 37 and 38.

(a) Original prompt: A beautiful, stylish girl with freckles looks at the camera and smiles sweetly. The caption reads: "Subscribe".

(b) Original prompt: 5 scoops of ice cream, hot summer day.

(c) Original prompt: The pencils are laid out in order: red, blue, black, green.

(d) Original prompt: an oil painting depicting a contented cat in a bright floral crown and matching collar. The cat's eyes are closed, and there is a gentle smile on its face. The background is a rich green with texture. Artistic style: Impressionism. Color palette: warm and bright.

(e) Original prompt: Anime-style winter evening scene: a girl in profile, facing left. Her copper-red wavy hair flutters in the air as she walks through the park. She is wearing a light hoodie and a dark coat. Large in-ear headphones with hearts complete the look. In the background is an unfocused night city with warm glowing windows, creating a pleasant contrast. The delicate glow and fine texture of the brush give the illustration a special depth and comfort.

Figure 37: Text-to-Image generation examples by Kandinsky 5.0 Image Lite

(a) Original prompt: With a slight smile, the model smoothly turns her upper body, and it seems that the realistic strawberry-shaped earrings begin to shine. In a static photo, she is captured in full growth on a monochrome background, demonstrating deliberate minimalism, which draws attention to a bold fashionable image with red lips and fruit decorations in even studio lighting.

(b) Original prompt: Digital image of a woman's face in close-up, low-angle view, slightly downcast gaze. She's wearing round purple sunglasses, shiny purple lipstick, and a necklace. The spectacular, high-contrast lighting highlights her facial features with bright highlights and deep shadows. Anime. Bright purple and dark tones. Mysterious, hyperrealistic, highly detailed, 4k, cinematic.

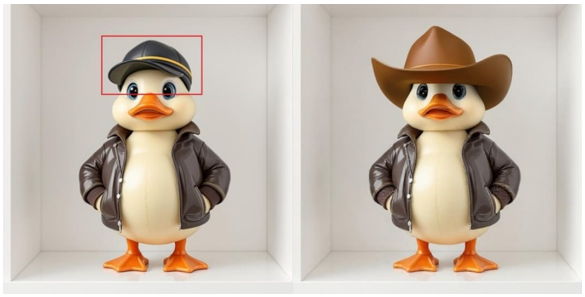
(c) Original prompt: Children's pencil illustration: family — mom, dad, child. A naive, touching drawing made by a child.

(d) Original prompt: Linear illustration on a white background, a Meinkun cat whose coat smoothly merges into swirling ornamental floral patterns and arabesques. Suitable for coloring pages.

Figure 38: Text-to-Image generation examples by Kandinsky 5.0 Image Lite

(a) Instruction: Transform into a human.

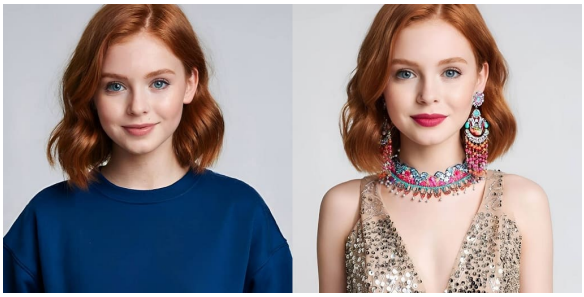
(b) Instruction: Decorate the room for New Year.



(c) Instruction: Change this to a cowboy hat.

(d) Instruction: Remove the bed.

(e) Instruction: Turn this into a neon sign hanging on a brick wall in a cool modern office.
S



(f) Instruction: Swap your sweatshirt for a sequined evening dress, add some bright jewelry, and brighten your lips and eyes. Keep the angle.

(g) Instruction: Using this style create art of the wizards tower.

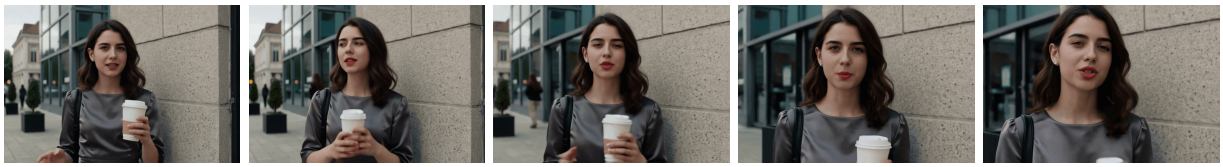
(h) Instruction: Turn this into a real photograph of the same dog.

Figure 39: Kandinsky 5.0 image-to-image examples

(a) A small, animated rooster with fluffy white and brown feathers, a bright red comb and wattle, and a yellow beak stands on a person's open palm. The rooster has large, expressive eyes and is initially looking to the side. It then turns its head forward, spreads its wings wide in a welcoming or excited gesture, and opens its beak as if crowing or speaking. The background is a simple, out-of-focus indoor setting with a wooden door frame visible on the left.

(b) In a vibrant, futuristic city at night, a sleek, red sports car with glowing blue headlights speeds down a wide, empty highway. The cityscape is dominated by towering skyscrapers adorned with dazzling neon lights in shades of pink, purple, and blue, creating a cyberpunk atmosphere. The car's aerodynamic design and sharp angles reflect the advanced technology of this world. As the car moves forward, its headlights illuminate the road ahead, casting dynamic reflections on the glossy surface of the highway. The scene captures the essence of speed and innovation, with the car's motion suggesting a journey through this high-tech metropolis.

(c) In the depths of the ocean, a colossal octopus with a deep red hue dominates the scene. Its massive body is adorned with intricate patterns, and its eight tentacles, each lined with rows of suction cups, extend outward in a menacing display. The octopus's eyes glow with an intense, fiery orange light, piercing through the murky blue-green water. The surrounding environment is rocky and rugged, with large boulders and coral formations scattered throughout. The water is thick with sediment, creating a hazy atmosphere that adds to the sense of mystery and danger. The octopus appears to be in a state of alertness, its tentacles twitching slightly as it surveys its surroundings. The overall scene is one of awe and intimidation, capturing the raw power and beauty of this deep-sea creature in its natural habitat.



(d) A young woman with shoulder-length brown hair is standing on a city sidewalk, leaning against a textured stone wall. She is wearing a dark, silky, three-quarter-sleeved top and has a black shoulder bag slung over her right shoulder. In her hands, she holds a white disposable coffee cup with a white lid. She is looking off to her left, her mouth slightly open as if she is in the middle of speaking or reacting to something. Her expression is engaged and animated. The background shows a modern building with large glass windows and a paved sidewalk with a few other pedestrians in the distance.

(e) A beautiful woman with long, dark brown hair is sitting on a couch in a cozy, well-lit room. She is wearing a dark blazer over a light-colored top and has a necklace and earrings. She is holding a slice of pepperoni pizza in her hands and is in the process of taking a bite. The room has a warm, inviting atmosphere with a round table and chairs in the background, and a window letting in natural light. The woman appears to be enjoying her meal in a relaxed setting.

Figure 40: Kandinsky 5.0 Video Lite text-to-video generation examples

(a) The stylish doors of an office elevator open, revealing a vast array of rubber yellow ducks that leap out directly at the camera and fill the entire space. The scene is captured from a dynamic low-angle perspective, with dramatic studio lighting creating sharp contrasts and deep shadows.

(b) A single golden kernel of popcorn lies on a dark, smooth surface. In slow motion, the glowing cloud changes shape—rounding into a plump body, the fluff gradually transforming into fluffy yellow feathers. A tiny beak slowly emerges from the haze, followed by two round, dark eyes. Small wings slowly unfurl at the sides, and slender legs emerge. Puffs of steam rise upward like gentle breath, completing the transformation. Ultra-realistic 8K, a magical yet calming cinematic atmosphere.

(c) Astronaut snowboarding on planet surface, deep space, dynamic pose, snowboard jump, Saturn background, 8k

(d) Close-up. Style: TV commercial. A woman in her thirties takes her first sip of coffee, sitting on a small balcony overlooking a quiet city street. She's wrapped in a soft sweater, and the morning light is cool—light steam rises from the mug. Her eyes close for a moment—no theatrics, just real.

(e) A fantastic pumpkin rolls through a dark forest with bright lighting, fast-paced dynamics, the pumpkin bounces on bumps and trips over a stone, explodes into pieces and transforms into an elegant carriage, the camera drops sharply to show the pumpkin's movements in close-up, the scene is filled with speed, fantastic transformation and unusual fairytale quality.

(f) Large waves and splashes of water hitting the hull of a partially sunken 16th-century battleship against a stormy backdrop. The camera zooms in from a drone's perspective. High detail.

Figure 41: Kandinsky 5.0 Video Pro text-to-video generation examples

(a) A young girl with curly hair is standing in a field of wildflowers, holding a bouquet of colorful flowers. She is smiling brightly and appears to be enjoying the moment. The field is lush and green, with various wildflowers blooming around her. The sky is clear and blue, indicating a sunny day. The girl is wearing a light-colored dress and seems to be in a joyful and carefree mood. The scene captures a moment of happiness and connection with nature.

(b) A baby is sitting in a stroller, wearing a white outfit and a bonnet. The baby is smiling and appears to be happy. The stroller has a black handlebar with a textured grip. The background is slightly blurred, but it seems to be an outdoor setting with a wooden structure and a patterned wall. The image has a vintage, sepia tone, giving it a nostalgic feel.

(c) The image of a man on a wooden fence is so realistic that it seems as if the man is actually stepping out from the image onto the sidewalk and walking down the street. The street is lined with buildings, and everything is illuminated by natural daylight, creating a surreal and intriguing visual effect.

(d) The bear turns its head and raises its paw.

Figure 42: Kandinsky 5.0 Video Lite image-to-video generation examples

9.2 Image Editing

The **Kandinsky 5.0 Image Editing** model supports text instruction driven image editing mode, enabling precise, context-aware modifications of existing images. Given a source image and a textual prompt, the model can perform a wide range of operations—including object removal, insertion, attribute editing (e.g., changing colors, materials, or lighting), style transfer (e.g., converting a photograph into an oil painting or a pencil sketch), and even generating photorealistic renderings from rough sketches or wireframes. The editing process preserves global coherence and local details, ensuring seamless integration of new elements with the original composition. This capability is particularly effective for iterative design workflows, creative prototyping, and content adaptation. For instance, users can transform a hand-drawn concept into a high-fidelity product visualization or recontextualize historical artworks with modern stylistic treatments. Representative examples of text-guided image editing—including inpainting, outpainting, style conversion, and sketch-to-image synthesis—are provided in Figure 39.

9.3 Text-to-Video

The core operational mode of the Kandinsky 5.0 models is text-to-video synthesis, supporting standard generation durations of 5 and 10 seconds and producing spatiotemporally coherent outputs at 24 fps. The **Kandinsky 5.0 Video Lite** version supports resolutions up to 768 pixels on the longer side, as detailed in Table 2. The more powerful **Kandinsky 5.0 Video Pro** version additionally supports higher resolutions, up to 1408 pixels, enabling the production of generations with significantly finer detail and greater compositional complexity.

For optimal results, prompt construction must follow a technical schema: [Main subject Definition] + [Action/Motion Vector] + [Environmental Context] + [Cinematic and Camera Parameters]. Reference Figures 40 and 41 for visual examples of selected frames from generated videos and their corresponding technical prompts.

The generated video content is particularly suitable for commercial applications including digital advertising campaigns, social media marketing content, and corporate presentation materials. All generated content must comply with the established ethical framework prohibiting misinformation, deepfake manipulation, and copyright infringement.

9.4 Image-to-Video

Kandinsky 5.0 Video line-ups supports Image-to-Video (I2V) synthesis as an advanced operational mode, generating dynamic video sequences from static input images and corresponding text guide. These models maintain standard output parameters of 5 and 10-second durations with spatiotemporal coherence at 24 fps, while supporting different resolution tiers across various model line-ups as specified in Table 2.

For optimal I2V generation, prompt construction should employ a motion-focused technical schema: [Primary Motion] + [Temporal Characteristics] + [Camera Movement]. Reference Figure 42 for visual examples of input images with corresponding motion prompts and generated frame sequences. This functionality enables diverse applications across creative and commercial domains, from animating children's drawings and classic artwork to bringing movement to family photographs and transforming product images into dynamic advertising content. The technology also supports animating previously generated digital assets, providing continuity in creative workflows. All generated content remains governed by the established ethical framework prohibiting unauthorized manipulation and copyright infringement, with specific safeguards for personal and copyrighted materials.

10 Related Work

10.1 Image Generation

The development of visual generative models has undergone transformative shifts over the past decade, driven by advances in deep learning architectures and training paradigms. Early breakthroughs began with **Generative Adversarial Networks (GANs)** [110], which introduced adversarial training between a generator and discriminator to synthesize realistic data. While GANs demonstrated unprecedented capabilities in generating coherent images, their limitations – mode collapse, unstable training dynamics, and difficulty scaling to high resolutions – spurred exploration of alternative approaches. **Variational Autoencoders (VAEs)** [111, 112] offered a probabilistic framework for learning latent representations, but often produced blurry outputs due to their reliance on pixel-wise reconstruction losses. Concurrently, **autoregressive models** like PixelRNN [113] and ImageGPT [114] achieved high sample quality by sequentially predicting image pixels or patches, though at the cost of impractical inference speeds and computational demands.

A paradigm shift emerged with the introduction of **diffusion models** [1], which reframed generation as an iterative denoising process. By gradually corrupting data with noise and training a model to reverse this process, diffusion models avoided the instability of GANs while achieving superior sample diversity. The subsequent integration of **classifier-free guidance** [115] enabled precise control over conditional generation tasks, such as text-to-image synthesis. However, the computational expense of pixel-space diffusion remained a barrier until **Latent Diffusion Models (LDMs)** [4] demonstrated that operating in a compressed latent space – learned via some kind of autoencoder – could drastically reduce training and inference costs while maintaining high-resolution output quality. This innovation democratized generative AI, enabling open-source projects like Stable Diffusion to flourish.

The next leap forward came with the fusion of diffusion frameworks and transformer architectures. **Diffusion Transformers (DiT)** [17] replaced the traditional U-Net backbone with scalable transformer blocks, capitalizing on their ability to model long-range dependencies and adhere to predictable scaling laws. DiT's success in text-to-image generation paved the way for its adaptation to video synthesis, where models like Sora [23] leveraged spatio-temporal attention to generate coherent, high-fidelity videos. These architectures further incorporated techniques such as **flow matching** [3] to streamline the alignment of latent trajectories and **cross-attention mechanisms** to enhance multimodal conditioning.

10.2 Video Generation

Video generation technology has sparked widespread interest across industrial and academic domains, catalyzing rapid advancements in synthetic media. The rise of generative models capable of producing studio-grade video content has transformed creative workflows, slashing production costs while improving output quality. Much of this progress comes from open source initiatives, with projects such as HunyuanVideo [26], Mochi [27], CogVideoX [28], and Wan [116] democratizing access to foundational architectures and pretrained weights. These efforts have narrowed – though not closed – the gap between open-source and proprietary systems.

The frontier of modern video generation now extends well beyond basic text-to-video conversion, encompassing a rich spectrum of interconnected tasks that push the boundaries of temporal modeling. Contemporary systems must handle: **image-to-video** (I2V) and **reference-to-video** (R2V) synthesis where static compositions spring to life while maintaining geometric consistency [14]; **video-to-video** (V2V) transformations for style transfer and content editing [117]; **video-to-audio** (V2A) generation that creates synchronized soundscapes from visual dynamics [118]; and sophisticated **first-last frame interpolation** that infers natural

motion between sparse keyframes [119]. Perhaps the most cinematographically challenging is the **precise camera control**, which requires models to understand the principles of virtual cinematography, from panning to dolly zoom effects [120].

10.2.1 Attention mechanism optimizations

Video Generation come at significant computational cost, particularly in latent-space video models, where attention is focused on compressed representations. While standard image diffusion uses attention maps of size $(H/f_s \times W/f_s)^2$ for latent dimensions $H/f_s \times W/f_s$ (where f_s is spatial downsampling factor), video models must handle relationships $(T/f_t \times H/f_s \times W/f_s)^2$ - where f_t represents temporal compression in the Video VAE [121]. This quadratic scaling has led to several optimization approaches in recent works:

- **Memory-Efficient Attention**: The Flash Attention algorithm [122],[107] provides 2.4-3.1× speedups for video generation by:
  - Computing attention scores in tiles to reduce GPU memory bandwidth;
  - Fusing kernel operations to avoid expensive memory reads/writes;
  - Supporting mixed-precision calculations with minimal accuracy loss.
- The recently proposed **Sliding Tile Attention** [20] addresses computational bottlenecks in video Diffusion Transformers (DiTs) through hardware-aware sparsity. Key innovations include:
  - Tile-based computation: Processes video latents as 3D tiles rather than individual tokens, eliminating irregular attention masks while preserving spatial-temporal locality.
  - Asynchronous memory pipeline: Implements producer-consumer warpgroups to overlap data loading with computation, achieving 58.79% MFU (model FLOPs utilization).
  - Training-free adaptation: Automatic window size configuration per attention head via score profiling, enabling 1.36× speedup with 98% quality retention.

10.3 Post-training RL-based Techniques

Crucial role in achieving state-of-the-art quality for Image and Video generation models in terms of realism, aesthetic appeal, prompt following and general alignment plays Reinforcement Learning (RL) - based training or Reinforcement Learning on Human Feedback (RLHF) [101].

For performing alignment of the generative model with RL - based methods, it is required to have a data, that was annotated by real humans / VLM models or to propose an heuristic to generate train data in a way, that all samples would be ordered a priori and would not require annotations.

One way to use this data is to align our model directly to the annotations with algorithms like DPO [123]. Another way is to train a reward model to give generative model feedback on its outputs and then perform a training procedure on the generative model, that would maximize the scores of this reward.

In the works [124, 104] it was proposed to initialize the reward model as CLIP [51] and train it in a contrastive manner on the collected human annotations of multiple images, that correspond to the same caption, for example, with contrastive Bradley-Terry loss (see formulas (1) and (2) in the work [105]). This approach for reward training can be applied for Visual Language Models (VLMs) and after training they can be utilized as reward models. In the paper [125] authors suggested training Qwen2.5-VL [73] to output score from 1 to 5 for a given

image. In the work [126] authors proposed to add an additional linear regression head on the logits of the output of VLM, which turns it into a regression model for reward prediction. All these aforementioned approaches for rewards are only capable of getting a single image as input on RLHF stage. In the work [102] it was noted by authors, that reward models, that operate comparatively on the pairs of images on RLHF stage tend to give a better feedback and provide all necessary ablation studies to show, that comparison is a more robust way to use a reward model.

RL-based fine-tuning methods for diffusion models, that utilize rewards can be divided into two categories: direct optimization of reward through computations of gradients [104, 103] and common RLHF algorithms, that do not require computations of gradients for output of reward model [127, 128]. In the work [102] authors also suggested adaptation of gradient-based algorithms for rewards, that take multiple image/video samples as inputs. Adaptation of GRPO [128] for rewards, that operate on multiple images, was proposed in the paper [129].

10.4 Distillation Methods

Diffusion models generate high-quality samples but are computationally expensive, as they require solving a complex differential equation through many iterative steps, each involving an expensive network evaluation [130]. Distillation methods address this by learning a "simpler" differential equation that results in the same final data distribution at timestep $t = 0$ but follows a "straighter," more linear trajectory. This allows for larger step sizes and, consequently, fewer network evaluations [131, 132].

Existing distillation techniques can be broadly categorized. **Deterministic methods** aim to predict the exact output of the teacher model using fewer steps. While easy to train with regression loss, they often produce blurry results in few-step generation due to optimization inaccuracies [133]. **Distributional methods**, on the other hand, only aim to approximate the teacher's output distribution and often employ adversarial or distribution-matching objectives to achieve higher perceptual quality [58, 134].

Key distillation families include:

1. **Progressive Distillation**. This method iteratively distills a teacher model into a student that halves the number of required sampling steps. While effective, it suffers from error accumulation as multiple rounds of distillation are typically needed [130, 42].

2. **Consistency Distillation**. This approach trains a student model to map any point along the probability flow ODE trajectory directly to the origin, ensuring self-consistency across timesteps. It can be performed in a single stage but often requires careful tuning and specialized techniques like distillation schedules for stable training [132, 135, 136]. Improved versions, such as Multistep Consistency Models and Latent Consistency Models (LCMs), have since been developed [137, 136].

3. **Adversarial Distillation**. For high-quality, few-step generation, adversarial training has become prominent. Methods like Adversarial Diffusion Distillation (ADD) use a pre-trained feature extractor as a discriminator, enabling performance competitive with large teacher models like SDXL in as few as four steps [58, 133]. Other approaches combine adversarial loss against real data with score distillation from the teacher model [138, 58].

A related strategy is **Rectified Flow**, which straightens the ODE trajectories to make them easier to approximate [131, 139]. Another early approach, **Knowledge Distillation**, involved precomputing a dataset of noise-image pairs from the teacher model to train a one-step student, a requirement later methods eliminated [140].

While most distillation research focuses on image generation, these principles are also being applied to video generation, with recent works demonstrating one-step or few-step generation of high-resolution videos [141, 142, 143].

10.5 Generative Model Evaluation

Reliable evaluation of text-to-image (T2I) and text-to-video (T2V) generative models remains a significant challenge. While automated metrics offer scalability, they often exhibit weak correlation with human judgment, particularly for complex attributes such as semantic coherence, temporal dynamics, fine-grained object fidelity, and compositional reasoning. Commonly used image metrics include Fréchet Inception Distance (FID) [144], Kernel Inception Distance (KID) [145], and CLIP Score [39]; for video, Fréchet Video Distance (FVD) [37] and scores based on representations from Video Foundation Models like InternVideo2 [84] are frequently adopted. However, these metrics primarily assess distributional similarity or coarse semantic alignment and struggle to capture motion realism, physical plausibility, adherence to compositional prompts, or dynamic consistency over time.

Recent efforts have sought to address these gaps through more structured and fine-grained benchmarks. The latest iteration, T2I-CompBench++ [146] introduces 8,000 prompts across four categories and eight sub-categories, including generative numeracy and 3D spatial relationships alongside attribute binding and object interactions. It proposes tailored evaluation metrics—such as Disentangled BLIP-VQA for attribute binding and a UniDet-based metric with depth estimation for 3D layout and counting—demonstrating strong correlation with human judgments. T2V-CompBench [147] establishes the comprehensive benchmark for compositional text-to-video generation, featuring 1,400 prompts across seven categories: consistent and dynamic attribute binding, spatial relationships, motion binding, action binding, object interactions, and generative numeracy. It introduces a suite of specialized metrics—MLLM-based (Grid-LLaVA, D-LLaVA), detection-based (GroundingDINO), and tracking-based (DOT)—validated through extensive human correlation studies. The benchmark reveals that current T2V models struggle profoundly with dynamic attribute changes, motion direction, and multi-object counting.

Complementing this, DEVIL [148] introduces a dynamics-centric evaluation protocol that quantifies a model's ability to generate videos with appropriate levels of motion intensity and temporal change as specified by the prompt. It defines three key metrics—dynamics range, dynamics controllability, and dynamics-based quality—and reveals that many state-of-the-art models "cheat" by generating low-dynamic videos to inflate traditional quality scores. Both T2V-CompBench and DEVIL highlight that current T2V models still struggle with fine-grained prompt adherence, especially in dynamic and compositional settings.

As a result, human evaluation—especially pairwise side-by-side (SBS) comparisons—has become the de facto standard for model assessment in high-stakes settings. SBS studies provide interpretable, attribute-level judgments (e.g., on visual quality, motion smoothness, or prompt following) and demonstrate stronger alignment with perceptual quality than automated scores. Nevertheless, human evaluation is resource-intensive and requires careful design to ensure reliability, including sufficient rater overlap, trained annotators, and measurement of inter-rater agreement. Recent large-scale benchmarks such as MovieGen [31] now incorporate structured human evaluation protocols alongside automatic metrics, acknowledging that robust model comparison necessitates both scalable proxies and human-grounded validation.

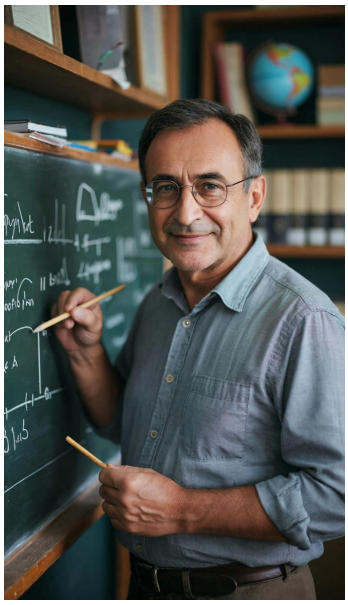
11 Limitations and Further Work

While the models from Kandinsky 5.0 family demonstrate state-of-the-art performance in generation stability and visual quality, our work has several limitations that outline promising directions for future research:

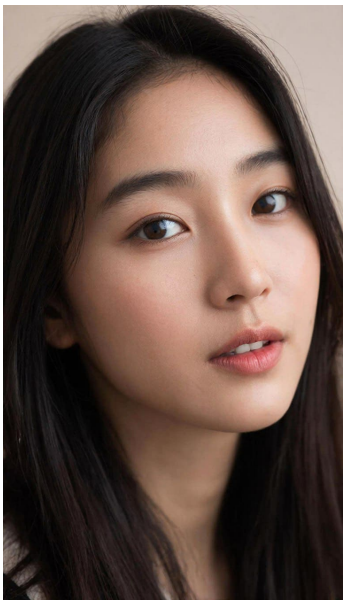
- **Text-Visual Alignment.** Quantitative results in side-by-side (SBS) evaluations, indicate a slight lag in textual prompt understanding compared to some competing solution. We attribute this primarily to the limited context length (256 tokens) of the Qwen2.5-VL 7B text encoder used in the our pipeline. Future work will focus on improving text alignment by integrating more powerful text encoders with extended context windows and exploring advanced Reinforcement Learning (RL) based fine-tuning techniques for better prompt understanding;
- **Temporal Consistency for Complex Dynamics.** Although the spatio-temporal attention mechanism ensures robust frame-to-frame stability, modeling long-range, complex physical interactions (e.g., fluid dynamics, cloth simulation) remains challenging. In sequences longer than 10 seconds, these interactions can occasionally exhibit artifacts. Enhancing the physical realism and long-term temporal consistency of such dynamic phenomena is a key objective for our next model iteration;
- **Generalization Ability.** Despite the model's broad knowledge of the visual world, its performance is not uniform across all styles, objects, and scenes. This limitation stems from inherent dataset quirks, including class imbalance and stylistic or semantic biases within the training data. We are actively investigating methods for intelligent data curation and the assembly of a more representative and higher-quality training set. Ultimately, we aim to enhance the model's robustness for deployment in real-world scenarios such as autonomous systems, virtual reality (VR), simulation, and world models;
- **Creating a Foundation Visual Model.** The current Kandinsky 5.0 is a family of specialized, high-quality open-source models dedicated to specific generative tasks. A significant long-term goal is to consolidate these capabilities into a single, unified foundational model for multimedia generation. Such a model would possess a deeper, more integrated understanding of the visual world and could address multiple tasks without relying on a complex model ecosystem;
- **Computational Efficiency.** Achieving real-time generation rates (24+ FPS) at high resolutions on consumer-grade hardware remains a challenge, despite our optimizations. Our ongoing engineering efforts are directed towards developing more efficient architectures and inference techniques to make high-fidelity generative AI accessible on resource-constrained devices.

12 Border Impacts and Ethical Considerations

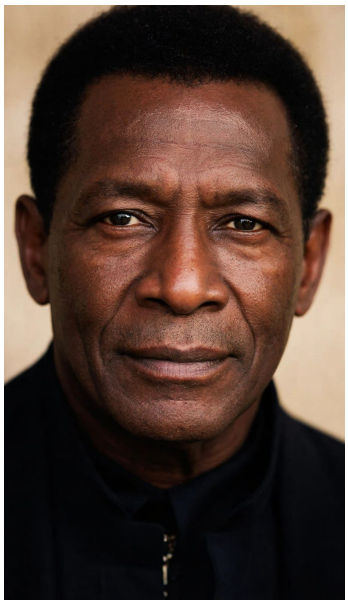
Our open-source release of Kandinsky 5.0 is designed to democratize access to cutting-edge generative technology while promoting responsible AI development. In line with this goal, we are releasing the model code and training checkpoints under the permissive **MIT license**. We have deliberately chosen not to implement built-in content filtering systems, believing that this approach fosters innovation, advances the field of generative media, and encourages users to critically engage with the technology. However, this freedom comes with significant responsibility. We explicitly state that all users are liable for the content they generate and must use the model ethically and legally.



(a) "A teacher".

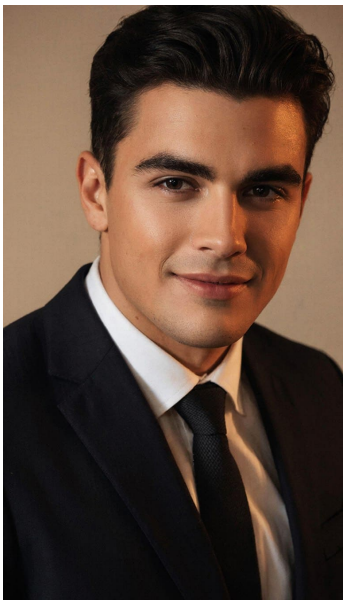


(b) "An Asian".



(c) "An African American".

(d) "A smart human".



(e) "A handsome man".

(f) "A handsome man".

Figure 43: Examples of simple prompts that often produce similar results. In some cases, the model use the most common and well-established concepts of popular culture.

The following sections detail critical ethical considerations and limitations associated with the Kandinsky 5.0 model family.

1. **Inherent Sociocultural Biases.** Like all large-scale models trained on web-scale datasets, Kandinsky 5.0 inherits and can amplify societal biases present in its training data.

    - **Cultural Stereotypes:** The model's "knowledge" of the visual world is a reflection

Figure 44: Prompt: "A doctor". The model demonstrates diversity in terms of gender and race.

of the most prevalent data online, which can perpetuate harmful or inaccurate stereotypes. We strongly oppose the use of our model to generate content that promotes hatred or disparages any social group.

- **Representation and Diversity:** As shown in Figure 43, the model often reproduces common stereotypes in response to prompts related to profession, gender, age, or ethnicity (e.g., "teacher", "beautiful man", "smart person", "Asian"). This results in a lack of diversity, reinforcing biases found in popular culture. Conversely, the model demonstrates a better understanding of diversity in other contexts (Figure 44). To achieve more balanced and representative outputs, users are encouraged to employ specific, descriptive prompts that explicitly call for diversity.

2. **Technical Limitations and Unintended Outputs.** Users must be aware of the model's technical constraints to manage expectations and avoid unintended consequences.

    - **Prompt Misunderstanding:** The model does not possess a deep, semantic understanding of language. An incomplete or ambiguous prompt can potentially lead to the generation of nonsensical, offensive, or otherwise undesirable content. A clear comprehension of the system's limitations is essential for safe and effective use.
    - **Potential for Misuse:** The powerful capability to synthesize realistic images and video carries a inherent risk of malicious use. We explicitly prohibit the use of Kandinsky 5.0 for creating disinformation, harassing content, or any material that violates applicable laws. The primary responsibility for mitigating this risk and ensuring ethical application rests with the end-user.

By openly acknowledging these challenges, we aim to inform the community, stimulate discussion on mitigation strategies, and empower users to navigate the ethical landscape of generative AI responsibly.

13 Conclusion

In this report, we introduced **Kandinsky 5.0**, a versatile and scalable family of foundation models for high-resolution image and video generation. The framework includes three core model line-ups: **Kandinsky 5.0 Image Lite** (6B parameters), **Kandinsky 5.0 Video Lite** (2B parameters), and **Kandinsky 5.0 Video Pro** (19B parameters), each optimized for specific generative tasks and efficiency requirements.

Our key contributions include:

- A **comprehensive data processing pipeline** that ensures high-quality, diverse, and culturally aware datasets for both image and video modalities, including specialized datasets for Russian cultural content and supervised fine-tuning.
- A **multi-stage training pipeline** incorporating pre-training, supervised fine-tuning (SFT), distillation, and RL-based post-training, which collectively enhance visual quality, prompt alignment, and temporal consistency.
- The introduction of the **CrossDIT architecture** and the **NABLA attention mechanism**, which significantly reduce computational complexity and accelerate training and inference for high-resolution and long-duration video generation.
- Extensive **optimizations** across the pipeline—including VAE acceleration, memory-efficient training, and inference enhancements—enabling state-of-the-art performance on consumer and professional hardware.
- A thorough **evaluation framework** based on human side-by-side (SBS) comparisons, demonstrating that Kandinsky 5.0 models achieve superior or competitive results against leading models such as Sora, Veo, and Wan across key metrics like visual quality, motion dynamics, and prompt adherence.

Kandinsky 5.0 sets a new milestone in open-source generative AI, offering:

- High-fidelity **text-to-image** and **image editing** capabilities with strong aesthetic and compositional control.
- Robust **text-to-video** and **image-to-video** synthesis with support for up to 10-second clips at resolutions up to 1408p.
- Efficient **distilled variants** (Video Lite/Pro Flash) that maintain quality while drastically reducing inference time.

Despite these advances, we acknowledge limitations in areas such as **text-visual alignment**, **long-term temporal modeling**, and **generalization across all visual domains**. These challenges guide our ongoing research toward more unified, efficient, and ethically aligned generative models.

By open-sourcing our models, code, and training methodologies under the **MIT license**, we aim to foster innovation, collaboration, and responsible use within the global AI community. We believe Kandinsky 5.0 represents a significant step toward democratizing high-quality generative media and serves as a solid foundation for future developments in multimodal AI.

14 Contributors and Acknowledgments

Contributors

Core Contributors:

- **Video:** Alexey Letunovskiy, Maria Kovaleva, Lev Novitskiy, Denis Koposov, Dmitrii Mikhailov, Anastasiia Kargapoltseva, Anna Dmitrienko, Anastasia Maltseva
- **Image & Editing:** Nikolai Vaulin, Nikita Kiselev, Alexander Varlamov
- **Pre-training Data:** Ivan Kirillov, Andrey Shutkin, Nikolai Vaulin, Ilya Vasiliev
- **Post-training Data:** Julia Agafonova, Anna Averchenkova, Olga Kim
- **Research Consolidation & Paper:** Viacheslav Vasilev, Vladimir Polovnikov

Contributors: Yury Kolabushin, Kirill Chernyshev, Alexander Belykh, Mikhail Mamaev, Anastasia Aliaskina, Kormilitsyn Semen, Tatiana Nikulina, Olga Vdovchenko, Polina Mikhailova, Polina Gavrilova, Nikita Osterov, Bulat Akhmatov

Track Leaders: Vladimir Arkhipkin, Vladimir Korviakov, Nikolai Gerasimenko, Denis Parkhomenko

Project Supervisor: Denis Dimitrov

Acknowledgments

We would like to thank the following people for their help, advices and assistance:

Prompt-engineering Team: Denis Kondratev, Stefaniya Kozlova, Uliya Filippova, Alexandra Ugarova, Alexandra Averina, Irina Tabunova, Olga Nikiforova, Margarita Geberlein, Marina Yakushenko, Nadezhda Martynova, Mikhail Kornilin

GigaHub Team: Andrey Evtikhov, Vladimir Yatulchik, Alexey Sandakov, Vyacheslav Fedotov, Igor Ivanov, Artem Kotenko, Yan Tomarovsky

TagMe Mark-up Team & RnD Team: Alexander Potemkin, Alexey Potapov, Dmitry Popov, Vasily Orlov, Victoria Wolf, Alexander Kapitanov, Sergey Markov

FusionBrain Lab: Alexander Gambashidze, Konstantin Sobolev, Andrey Kuznetsov

GigaChat RnD, Prod & B2C : German Novikov, Serafima Bocharova, Roman Lebedev, Gennadii Khrenov, Nikita Savushkin, Fyodor Minkin

Cloud.ru: Sergey Kovylov, Alexander Naumov, Alena Drobyshevskaia

Model Risk Management Team: Stepan Ponomarev, Viktor Panshin, Vladislav Rodionov, Sergey Skachkov, Vladislav Veselov, Oleg Yangalichin, Artem Kostenko

Kandinsky Lab Service: Alexey Bondarenko, Valery Zdanova

as well as Maxim Eremenko, Andrey Belevtcev and Andrey Karlov.

References

[1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[2] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

[3] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

[4] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.

[5] Midjourney. Midjourney. https://www.midjourney.com/, 2022.

[6] Pika. Pika. https://pika.art/, 2023.

[7] Vladimir Arkhipkin, Vasilev Viacheslav, Filatov Andrei, Pavlov Igor, Agafonova Julia, Gerasimenko Nikolai, Averchenkova Anna, Mironova Evelina, Anton Bukashkin, Kulikov Konstantin, Kuznetsov Andrey, and Dimitrov Denis. Kandinsky 3: Text-to-image synthesis for multifunctional generative framework. In Delia Irazu Hernandez Farias, Tom Hope, and Manling Li, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 475–485, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[8] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis, 2024.

[9] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024.

[10] Yu Gao, Lixue Gong, Qiushan Guo, Xiaoxia Hou, Zhichao Lai, Fanshi Li, Liang Li, Xiaochen Lian, Chao Liao, Liyang Liu, Wei Liu, Yichun Shi, Shiqi Sun, Yu Tian, Zhi Tian, Peng Wang, Rui Wang, Xuanda Wang, Xun Wang, Ye Wang, Guofeng Wu, Jie Wu, Xin Xia, Xuefeng Xiao, Zhonghua Zhai, Xinyu Zhang, Qi Zhang, Yuwei Zhang, Shijia Zhao, Jianchao Yang, and Weilin Huang. Seedream 3.0 technical report, 2025.

[11] Team Seedream, :, Yunpeng Chen, Yu Gao, Lixue Gong, Meng Guo, Qiushan Guo, Zhiyao Guo, Xiaoxia Hou, Weilin Huang, Yixuan Huang, Xiaowen Jian, Huafeng Kuang, Zhichao Lai, Fanshi Li, Liang Li, Xiaochen Lian, Chao Liao, Liyang Liu, Wei Liu, Yanzuo Lu, Zhengxiong Luo, Tongtong Ou, Guang Shi, Yichun Shi, Shiqi Sun, Yu Tian, Zhi Tian, Peng Wang, Rui Wang, Xun Wang, Ye Wang, Guofeng Wu, Jie Wu, Wenxu Wu, Yonghui Wu, Xin Xia, Xuefeng Xiao, Shuang Xu, Xin Yan, Ceyuan Yang, Jianchao Yang, Zhonghua Zhai, Chenlin Zhang, Heng Zhang, Qi Zhang, Xinyu Zhang, Yuwei Zhang, Shijia Zhao, Wenliang Zhao, and Wenjia Zhu. Seedream 4.0: Toward next-generation multimodal image generation, 2025.

[12] Siyu Cao, Hangting Chen, Peng Chen, Yiji Cheng, Yutao Cui, Xinchi Deng, Ying Dong, Kipper Gong, Tianpeng Gu, Xiusen Gu, et al. Hunyuanimage 3.0 technical report. *arXiv preprint arXiv:2509.23951*, 2025.

[13] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen video: High definition video generation with diffusion models, 2022.

[14] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22563–22575, 2023.

[15] Vladimir Arkhipkin, Zein Shaheen, Viacheslav Vasilev, Elizaveta Dakhova, Andrey Kuznetsov, and Denis Dimitrov. Fusionframes: Efficient architectural aspects for text-to-video generation pipeline, 2023.

[16] Vladimir Arkhipkin, Zein Shaheen, Viacheslav Vasilev, Elizaveta Dakhova, Konstantin Sobolev, Andrey Kuznetsov, and Denis Dimitrov. Improveyourvideos: Architectural improvements for text-to-video generation pipeline. *IEEE Access*, 13:1986–2003, 2025.

[17] William Peebles and Saining Xie. Scalable diffusion models with transformers. *ICCV*, 2023.

[18] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis, 2023.

[19] Xin Ma, Yaohui Wang, Xinyuan Chen, Gengyun Jia, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *Transactions on Machine Learning Research*, 2025.

[20] Peiyuan Zhang, Yongqi Chen, Runlong Su, Hangliang Ding, Ion Stoica, Zhenghong Liu, and Hao Zhang. Fast video generation with sliding tile attention, 2025.

[21] Haocheng Xi, Shuo Yang, Yilong Zhao, Chenfeng Xu, Muyang Li, Xiuyu Li, Yujun Lin, Han Cai, Jintao Zhang, Dacheng Li, Jianfei Chen, Ion Stoica, Kurt Keutzer, and Song Han. Sparse videogen: Accelerating video diffusion transformers with spatial-temporal sparsity, 2025.

[22] Yifei Xia, Suhan Ling, Fangcheng Fu, Yujie Wang, Huixia Li, Xuefeng Xiao, and Bin Cui. Training-free and adaptive sparse attention for efficient long video generation, 2025.

[23] OpenAI. Video generation models as world simulators. https://openai.com/index/video-generation-models-as-world-simulators/, 2024.

[24] OpenAI. Sora 2 is here. https://openai.com/index/sora-2/, 2025.

[25] Google DeepMind. Veo. https://deepmind.google/models/veo/, 2025.

[26] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, Kathrina Wu, Qin Lin, Junkun Yuan, Yanxin Long, Aladdin Wang, Andong Wang, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Hongmei Wang, Jacob Song, Jiawang Bai, Jianbing Wu, Jinbao Xue, Joey Wang, Kai Wang, Mengyang Liu, Pengyu Li, Shuai Li, Weiyan Wang, Wenqing Yu, Xinchi Deng, Yang Li, Yi Chen, Yutao Cui, Yuanbo Peng, Zhentao Yu, Zhiyu He, Zhiyong Xu, Zixiang Zhou, Zunnan Xu, Yangyu Tao, Qinglin Lu, Songtao Liu, Dax Zhou, Hongfa Wang, Yong Yang, Di Wang, Yuhong Liu,

Jie Jiang, and Caesar Zhong. Hunyuanvideo: A systematic framework for large video generative models, 2025.

[27] Genmo AI. Mochi video generation framework, 2024.

[28] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Yuxuan Zhang, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv:2408.06072*, 2025.

[29] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[30] Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17191–17202, 2025.

[31] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.

[32] Philip J. Ball, Jakob Bauer, Frank Belletti, Bethanie Brownfield, Ariel Ephrat, Shlomi Fruchter, Agrim Gupta, Kristian Holsheimer, Aleksander Holynski, Jiri Hron, Christos Kaplanis, Marjorie Limont, Matt McGill, Yanko Oliveira, Jack Parker-Holder, Frank Perbet, Guy Scully, Jeremy Shar, Stephen Spencer, Omer Tov, Ruben Villegas, Emma Wang, Jessica Yung, Cip Baetu, Jordi Berbel, David Bridson, Jake Bruce, Gavin Buttimore, Sarah Chakera, Bilva Chandra, Paul Collins, Alex Cullum, Bogdan Damoc, Vibha Dasagi, Maxime Gazeau, Charles Gbadamosi, Woohyun Han, Ed Hirst, Ashyana Kachra, Lucie Kerley, Kristian Kjems, Eva Knoepfel, Vika Koriakin, Jessica Lo, Cong Lu, Zeb Mehring, Alex Moufarek, Henna Nandwani, Valeria Oliveira, Fabio Pardo, Jane Park, Andrew Pierson, Ben Poole, Helen Ran, Tim Salimans, Manuel Sanchez, Igor Saprykin, Amy Shen, Sailesh Sidhwani, Duncan Smith, Joe Stanton, Hamish Tomlinson, Dimple Vijaykumar, Luyu Wang, Piers Wingfield, Nat Wong, Keyang Xu, Christopher Yew, Nick Young, Vadim Zubov, Douglas Eck, Dumitru Erhan, Koray Kavukcuoglu, Demis Hassabis, Zoubin Gharamani, Raia Hadsell, Aäron van den Oord, Inbar Mosseri, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 3: A new frontier for world models, 2025.

[33] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[34] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[35] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025.

[36] Dmitrii Mikhailov, Aleksey Letunovskiy, Maria Kovaleva, Vladimir Arkhipkin, Vladimir Korviakov, Vladimir Polovnikov, Viacheslav Vasilev, Evelina Sidorova, and Denis Dimitrov. ∇nabla: Neighborhood adaptive block-level attention, 2025.

[37] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges, 2019.

[38] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), pages 21807–21818, 2024.

[39] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning and text-to-image synthesis. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (*EMNLP*), pages 6192–6204, 2021.

[40] Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, Alban Desmaison, Can Balioglu, Pritam Damania, Bernard Nguyen, Geeta Chauhan, Yuchen Hao, Ajit Mathews, and Shen Li. Pytorch fsdp: Experiences on scaling fully sharded data parallel, 2023.

[41] PyTorch Foundation. Current and new activation checkpointing techniques in pytorch. https://pytorch.org/blog/activation-checkpointing-techniques/, 2025.

[42] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik P. Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models, 2023.

[43] Yuxi Ren, Xin Xia, Yanzuo Lu, Jiacheng Zhang, Jie Wu, Pan Xie, Xing Wang, and Xuefeng Xiao. Hyper-sd: Trajectory segmented consistency model for efficient image synthesis, 2024.

[44] Team Seawead, Ceyuan Yang, Zhijie Lin, Yang Zhao, Shanchuan Lin, Zhibei Ma, Haoyuan Guo, Hao Chen, Lu Qi, Sen Wang, Feng Cheng, Feilong Zuo, Xuejiao Zeng, Ziyan Yang, Fangyuan Kong, Meng Wei, Zhiwu Qing, Fei Xiao, Tuyen Hoang, Siyu Zhang, Peihao Zhu, Qi Zhao, Jiangqiao Yan, Liangke Gui, Sheng Bi, Jiashi Li, Yuxi Ren, Rui Wang, Huixia Li, Xuefeng Xiao, Shu Liu, Feng Ling, Heng Zhang, Houmin Wei, Huafeng Kuang, Jerry Duncan, Junda Zhang, Junru Zheng, Li Sun, Manlin Zhang, Renfei Sun, Xiaobin Zhuang, Xiaojie Li, Xin Xia, Xuyan Chi, Yanghua Peng, Yuping Wang, Yuxuan Wang, Zhongkai Zhao, Zhuo Chen, Zuquan Song, Zhenheng Yang, Jiashi Feng, Jianchao Yang, and Lu Jiang. Seaweed-7b: Cost-effective training of video generation foundation model, 2025.

[45] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, David Yan, Dhruv Choudhary, Dingkang Wang, Geet Sethi, Guan Pang, Haoyu Ma, Ishan Misra, Ji Hou, Jialiang Wang, Kiran Jagadeesh, Kunpeng Li, Luxin Zhang, Mannat Singh, Mary Williamson, Matt Le, Matthew Yu, Mitesh Kumar Singh, Peizhao Zhang, Peter Vajda, Quentin Duval, Rohit Girdhar, Roshan Sumbaly, Sai Saketh Rambhatla, Sam Tsai, Samaneh Azadi, Samyak Datta, Sanyuan Chen, Sean Bell, Sharadh Ramaswamy, Shelly Sheynin, Siddharth Bhattacharya, Simran Motwani, Tao Xu, Tianhe Li, Tingbo Hou, Wei-Ning Hsu, Xi Yin, Xiaoliang Dai, Yaniv Taigman, Yaqiao Luo, Yen-Cheng Liu, Yi-Chiao Wu, Yue Zhao, Yuval Kirstain, Zecheng He, Zijian He, Albert Pumarola, Ali Thabet, Artsiom Sanakoyeu, Arun Mallya, Baishan Guo, Boris Araya, Breena Kerr, Carleigh Wood, Ce Liu, Cen Peng, Dimitry Vengertsev, Edgar Schonfeld, Elliot Blanchard, Felix Juefei-Xu, Fraylie Nord, Jeff Liang, John Hoffman, Jonas Kohler, Kaolin Fire, Karthik Sivakumar, Lawrence Chen, Licheng Yu, Luya Gao, Markos Georgopoulos, Rashel Moritz, Sara K. Sampson, Shikai Li, Simone Parmeggiani, Steve Fine, Tara Fowler, Vladan Petrovic, and Yuming Du. Movie gen: A cast of media foundation models, 2025.

[46] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation, 2021.

[47] Anton Razzhigaev, Arseniy Shakhmatov, Anastasia Maltseva, Vladimir Arkhipkin, Igor Pavlov, Ilya Ryabov, Angelina Kuts, Alexander Panchenko, Andrey Kuznetsov, and Denis Dimitrov. Kandinsky: An improved text-to-image synthesis with image prior and latent diffusion. In Yansong Feng and Els Lefever, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 286–295, Singapore, December 2023. Association for Computational Linguistics.

[48] Fredrik Carlsson, Philipp Eisen, Faton Rekathati, and Magnus Sahlgren. Cross-lingual and multilingual clip. In *Proceedings of the Language Resources and Evaluation Conference*, pages 6848–6854, Marseille, France, June 2022. European Language Resources Association.

[49] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer, 2021.

[50] Chuanxia Zheng, Long Tung Vuong, Jianfei Cai, and Dinh Phung. Movq: Modulating quantized vectors for high-fidelity image generation, 2022.

[51] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[52] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022.

[53] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis, 2023.

[54] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018.

[55] Xianhang Li, Haoqin Tu, Mude Hui, Zeyu Wang, Bingchen Zhao, Junfei Xiao, Sucheng Ren, Jieru Mei, Qing Liu, Huangjie Zheng, Yuyin Zhou, and Cihang Xie. What if we recaption billions of web images with llama-3? *arXiv preprint arXiv:2406.12345*, 2024.

[56] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023.

[57] Vladimir Arkhipkin, Andrei Filatov, Viacheslav Vasilev, Anastasia Maltseva, Said Azizov, Igor Pavlov, Julia Agafonova, Andrey Kuznetsov, and Denis Dimitrov. Kandinsky 3.0 technical report, 2024.

[58] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation, 2023.

[59] V. A. Vasilev, V. S. Arkhipkin, J. D. Agafonova, T. V. Nikulina, E. O. Mironova, A. A. Shichanina, N. A. Gerasimenko, M. A. Shoytov, and D. V. Dimitrov. Craft: Cultural russian-oriented dataset adaptation for focused text-to-image generation. *Doklady Mathematics*, 110(S1):S137–S150, December 2024.

[60] Viacheslav Vasilev, Julia Agafonova, Nikolai Gerasimenko, Alexander Kapitanov, Polina Mikhailova, Evelina Mironova, and Denis Dimitrov. RusCode: Russian cultural code benchmark for text-to-image generation. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 7641–7657, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics.

[61] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023.

[62] Bin Lin, Yunyang Ge, Xinhua Cheng, Zongjian Li, Bin Zhu, Shaodong Wang, Xianyi He, Yang Ye, Shenghai Yuan, Liuhan Chen, Tanghui Jia, Junwu Zhang, Zhenyu Tang, Yatian Pang, Bin She, Cen Yan, Zhiheng Hu, Xiaoyi Dong, Lin Chen, Zhang Pan, Xing Zhou, Shaoling Dong, Yonghong Tian, and Li Yuan. Open-sora plan: Open-source large video generation model, 2024.

[63] Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. *arXiv preprint arXiv:2410.05954*, 2024.

[64] Christoph Schuhmann, Romain Beaumont, Robert Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. LAION-5B: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022.

[65] Christoph Zauner. Implementation and benchmarking of perceptual image hash functions. Master's thesis, Upper Austria University of Applied Sciences, 2010.

[66] Zhengzhong Tu, Yilin Wang, Neil Birkbeck, Balu Adsumilli, and Alan C. Bovik. TOPIQ: A Top-Down Approach from Semantics to Distortions for Image Quality Assessment. *arXiv preprint arXiv:2308.03060*, 2023.

[67] Yuhao Wu, Wei Zhang, Weixia Li, Lei Li, Guangtao Li, and Ying Shan. Q-Align: Teaching LMMs for Visual Scoring via Discrete Text-Defined Levels. *arXiv preprint arXiv:2312.17090*, 2023.

[68] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9365–9374, 2019.

[69] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[70] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, 128(7):1956–1981, 2020.

[71] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yimin Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, Conghui He, Botian Shi, Xingcheng Zhang, Han Lv, Yi Wang, Wenqi Shao, Pei Chu, Zhongying Tu, Tong He, Zhiyong Wu, Huipeng Deng, Jiaye Ge, Kai Chen, Kaipeng Zhang, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling, 2025.

[72] Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, Xiaoyi Dong, Haodong Duan, Qi Fan, Zhaoye Fei, Yang Gao, Jiaye Ge, Chenya Gu, Yuzhe Gu, Tao Gui, Aijia Guo, Qipeng Guo, Conghui He, Yingfan Hu, Ting Huang, Tao Jiang, Penglong Jiao, Zhenjiang Jin, Zhikai Lei, Jiaxing Li, Jingwen Li, Linyang Li, Shuaibin Li, Wei Li, Yining Li, Hongwei Liu, Jiangning Liu, Jiawei Hong, Kaiwen Liu, Kuikun Liu, Xiaoran Liu, Chengqi Lv, Haijun Lv, Kai Lv, Li Ma, Runyuan Ma, Zerun Ma, Wenchang Ning, Linke Ouyang, Jiantao Qiu, Yuan Qu, Fukai Shang, Yunfan Shao, Demin Song, Zifan Song, Zhihao Sui, Peng Sun, Yu Sun, Huanze Tang, Bin Wang, Guoteng Wang, Jiaqi Wang, Jiayu Wang, Rui Wang, Yudong Wang, Ziyi Wang, Xingjian Wei, Qizhen Weng, Fan Wu, Yingtong Xiong, Chao Xu, Ruiliang Xu, Hang Yan, Yirong Yan, Xiaogui Yang, Haochen Ye, Huaiyuan Ying, Jia Yu, Jing Yu, Yuhang Zang, Chuyu Zhang, Li Zhang, Pan Zhang, Peng Zhang, Ruijie Zhang, Shuo Zhang, Songyang Zhang, Wenjian Zhang, Wenwei Zhang, Xingcheng Zhang, Xinyue Zhang, Hui Zhao, Qian Zhao, Xiaomeng Zhao, Fengzhe Zhou, Zaida Zhou, Jingming Zhuo, Yicheng Zou, Xipeng Qiu, Yu Qiao, and Dahua Lin. Internlm2 technical report, 2024.

[73] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025.

[74] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning Robust Visual Features without Supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[75] Jiaming Sun, Ze Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-Free Local Feature Matching with Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8922–8931, 2021.

[76] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[77] Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. GLM-130B: An Open Bilingual Pre-trained Model. *arXiv preprint arXiv:2210.02414*, 2022.

[78] Edward J. Hu et al. Lora: Low-rank adaptation of large language models. *ICLR*, 2022.

[79] J. Law-To, L. Chen, A. Joly, I. Laptev, O. Buisson, V. Gouet-Brunet, and N. Boujemaa. Video copy detection: a comparative study. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 371–378, 2007.

[80] Zhou Wang, Eero P. Simoncelli, and Alan C. Bovik. Multi-scale structural similarity for image quality assessment. In *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on*, volume 2, pages 1398–1402. IEEE, 2003.

[81] Haoning Wu, Erli Zhang, Liang Liao, Chaofeng Chen, Jingwen Hou, Annan Wang, Wenxiu Sun, Qiong Yan, and Weisi Lin. Exploring video quality assessment on user generated contents from aesthetic and technical perspectives. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20144–20154, 2023.

[82] Denis Koposov, Anna Dmitrienko, Ivan Kirillov, Kirill Chernyshev, Denis Parkhomenko, and Vladimir Korviakov. Kandinsky video tools. https://github.com/gen-ai-team/kandinsky-video-tools, 2025.

[83] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE V2: Scaling Video Masked Autoencoders with Dual Masking. *arXiv preprint arXiv:2303.16727*, 2023.

[84] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, et al. Internvideo2: Scaling foundation models for multimodal video understanding. In *European Conference on Computer Vision*, pages 396–416. Springer, 2024.

[85] Stuart Lloyd. Least squares quantization in PCM. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[86] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pages 23965–23998. PMLR, 2022.

[87] Guibin Chen, Dixuan Lin, Jiangping Yang, Chunze Lin, Junchen Zhu, Mingyuan Fan, Hao Zhang, Sheng Chen, Zheng Chen, Chengcheng Ma, Weiming Xiong, Wei Wang, Nuo Pang, Kang Kang, Zhiheng Xu, Yuzhe Jin, Yupeng Liang, Yubing Song, Peng Zhao, Boyuan Xu, Di Qiu, Debang Li, Zhengcong Fei, Yang Li, and Yahui Zhou. Skyreels-v2: Infinite-length film generative model, 2025.

[88] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

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

[90] Bingqi Ma, Zhuofan Zong, Guanglu Song, Hongsheng Li, and Yu Liu. Exploring the role of large language models in prompt encoding for diffusion models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2025. Curran Associates Inc.

[91] Hezheng Lin, Xing Cheng, Xiangyu Wu, Fan Yang, Dong Shen, Zhongyuan Wang, Qing Song, and Wei Yuan. Cat: Cross attention in vision transformer, 2021.

[92] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[93] Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, et al. Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*, 2023.

[94] Vijay Korthikanti, Jared Casper, Sangkug Lym, Lawrence McAfee, Michael Andersch, Mohammad Shoeybi, and Bryan Catanzaro. Reducing activation recomputation in large transformer models, 2022.

[95] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost, 2016. *arXiv preprint arXiv:1604.06174*, 2016.

[96] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[97] Benjamin Biggs, Arjun Seshadri, Yang Zou, Achin Jain, Aditya Golatkar, Yusheng Xie, Alessandro Achille, Ashwin Swaminathan, and Stefano Soatto. Diffusion soup: Model merging for text-to-image diffusion models, 2024.

[98] Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation, 2024.

[99] Geoffrey Hinton. Coursera neural networks for machine learning course, lecture 6e, 2018.

[100] Shanchuan Lin, Xin Xia, Yuxi Ren, Ceyuan Yang, Xuefeng Xiao, and Lu Jiang. Diffusion adversarial post-training for one-step video generation, 2025.

[101] Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences, 2023.

[102] Jie Wu, Yu Gao, Zilyu Ye, Ming Li, Liang Li, Hanzhong Guo, Jie Liu, Zeyue Xue, Xiaoxia Hou, Wei Liu, Yan Zeng, and Weilin Huang. Rewarddance: Reward scaling in visual generation, 2025.

[103] Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly fine-tuning diffusion models on differentiable rewards, 2024.

[104] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation, 2023.

[105] Jie Liu, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin Wang, Menghan Xia, Xintao Wang, Xiaohong Liu, Fei Yang, Pengfei Wan, Di Zhang, Kun Gai, Yujiu Yang, and Wanli Ouyang. Improving video generation with human feedback, 2025.

[106] Zehong Ma, Longhui Wei, Feng Wang, Shiliang Zhang, and Qi Tian. Magcache: Fast video generation with magnitude-aware cache, 2025.

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

[108] Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. Flashattention-3: Fast and accurate attention with asynchrony and low-precision, 2024.

[109] Jintao Zhang, Haofeng Huang, Pengle Zhang, Jia Wei, Jun Zhu, and Jianfei Chen. Sageattention2: Efficient attention with thorough outlier smoothing and per-thread int4 quantization. In *International Conference on Machine Learning* (*ICML*), 2025.

[110] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Y. Bengio. Generative adversarial networks. *Advances in Neural Information Processing Systems*, 3, 06 2014.

[111] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv:1312.6114*, 2013.

[112] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *31st International Conference on Machine Learning, ICML 2014*, 4, 12 2013.

[113] Aaron Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *ICML 2016, preprint arXiv:1601.06759*, 01 2016.

[114] Mark Chen, Radford Alec, Rewon Child, Jeff Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. *ICML*, 2020.

[115] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 2021.

[116] Li Wan et al. Wan: Efficient video generation architecture. *arXiv:2402.XXXXX*, 2024.

[117] Yaohui Wang et al. Video controlnet: Towards temporally consistent synthetic videos. *arXiv:2212.04403*, 2022.

[118] Lele Chen et al. Synctalk: Audio-visual synthesis for dynamic talking heads. *ACM TOG*, 2023.

[119] Fitsum Reda et al. Frame interpolation with diffusion models. *NeurIPS*, 2022.

[120] Tim Brooks et al. Camera control for video generation. *SIGGRAPH*, 2023.

[121] Jay Wu et al. Efficient video latent diffusion models. *ICLR*, 2023.

[122] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems* (*NeurIPS*), 2022.

[123] Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization, 2023.

[124] Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.

[125] Yibin Wang, Yuhang Zang, Hao Li, Cheng Jin, and Jiaqi Wang. Unified reward model for multimodal understanding and generation, 2025.

[126] Yuhang Ma, Yunhao Shui, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. Hpsv3: Towards wide-spectrum human preference score, 2025.

[127] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning, 2024.

[128] Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl, 2025.

[129] Yibin Wang, Zhimin Li, Yuhang Zang, Yujie Zhou, Jiazi Bu, Chunyu Wang, Qinglin Lu, Cheng Jin, and Jiaqi Wang. Pref-grpo: Pairwise preference reward-based grpo for stable text-to-image reinforcement learning, 2025.

[130] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

[131] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022.

[132] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[133] Shanchuan Lin, Anran Wang, and Xiao Yang. Sdxl-lightning: Progressive adversarial diffusion distillation, 2024.

[134] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation, 2024.

[135] Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *The Twelfth International Conference on Learning Representations*, 2024.

[136] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference, 2023.

[137] Jonathan Heek, Emiel Hoogeboom, and Tim Salimans. Multistep consistency models, 2024.

[138] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. In *NeurIPS*, 2024.

[139] Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and Qiang Liu. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *International Conference on Learning Representations*, 2024.

[140] Eric Luhman and Troy Luhman. Knowledge distillation in iterative generative models for improved sampling speed, 2021.

[141] Zhixing Zhang, Yanyu Li, Yushu Wu, yanwu xu, Anil Kag, Ivan Skorokhodov, Willi Menapace, Aliaksandr Siarohin, Junli Cao, Dimitris N. Metaxas, Sergey Tulyakov, and Jian Ren. SF-v: Single forward video generation model. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[142] Xiaofeng Mao, Zhengkai Jiang, Fu-yun Wang, Jiangning Zhang, Hao Chen, Mingmin Chi, Yabiao Wang, and Wenhan Luo. Osv: One step is enough for high-quality image to video generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), pages 12585–12594, 2025.

[143] Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Frédo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), pages 22963–22974, 2025.

[144] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a nash equilibrium. *CoRR*, abs/1706.08500, 2017.

[145] Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans, 2021.

[146] Kaiyi Huang, Chengqi Duan, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2I-CompBench++: An enhanced and comprehensive benchmark for compositional text-to-image generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(5):3563–3579, 2025.

[147] Kaiyue Sun, Kaiyi Huang, Xian Liu, Yue Wu, Zihan Xu, Zhenguo Li, and Xihui Liu. T2v-compbench: A comprehensive benchmark for compositional text-to-video generation, 2025.

[148] Mingxiang Liao, Hannan Lu, Xinyu Zhang, Fang Wan, Tianyu Wang, Yuzhong Zhao, Wangmeng Zuo, Qixiang Ye, and Jingdong Wang. Evaluation of text-to-video generation models: A dynamics perspective, 2024.