

TL;DR UniAnimate targets consistent human image animation by unifying modalities inside a single video diffusion model, supporting both short and long videos. It reduces identity-alignment overhead, introduces a unified noise input for better long-term behavior, and considers state-space temporal modeling to replace heavy temporal Transformers.

What is UniAnimate?

UniAnimate focuses on human image animation: given a reference image (identity) and a target pose sequence, synthesize a coherent video following the poses while preserving the subject's appearance. The core idea is to avoid maintaining a separate identity/reference branch by embedding reference image, pose guidance, and noise video into a shared feature space within one unified video diffusion model.

Links:

- Project: <https://unianimate.github.io/>
 - Repo: <https://github.com/ali-vilab/UniAnimate>
 - Related (new model): UniAnimate-DiT based on Wan2.1 - <https://github.com/ali-vilab/UniAnimate-DiT>
-

Key ideas

- Unified feature space: Reference image, posture guidance, and noise are mapped into a common space inside a single video diffusion model to reduce optimization complexity and improve temporal coherence.
- Unified noise input: Supports random-noise starts and first-frame-conditioned inputs, helping extend sequence length and stabilize identity across longer videos.
- Efficient temporal modeling: Explores replacing temporal Transformer with a state-space model (SSM) for long-sequence efficiency.
- Practical engineering tips (from repo):
 - CPU offload for CLIP/VAE (set `CPU_CLIP_VAE: True`) to cut GPU memory (reported ~12 GB for 32×768×512).
 - Multi-segment parallel denoising on large-VRAM GPUs via `context_batch_size > 1`.

- Noise prior option that can improve background/appearance preservation in long videos.
-

Quickstart (inference)

Generate a short clip (32 frames, e.g., 512×768), then adjust settings:

```
# Install dependencies per repo instructions (PyTorch + deps)
# ...

# Short video generation
python inference.py --cfg configs/UniAnimate_infer.yaml

# Increase resolution (e.g., 768×1216) in configs/UniAnimate_infer.yaml:
# resolution: [768, 1216]
# Then re-run the same command.
```

Generate long videos (sliding window with temporal overlap):

```
python inference.py --cfg configs/UniAnimate_infer_long.yaml

# In this config, test_list entries contain:
# [frame_interval, reference_image_path, driving_pose_sequence_path]
# frame_interval=1 → use every pose frame; 2 → sample every two frames.
```

Config knobs and tips

- Length vs memory: Lower `max_frames` (e.g., 24/16/8) to run on smaller VRAM; larger values extend clips if memory allows.
 - Resolution: Trained at 512×768 but often generalizes to 768×1216; if appearance becomes inconsistent, try a different seed or revert resolution.
 - Long video stitching: For very long takes, you can render segments and feed the last frame of one segment as the first-frame condition for the next.
 - Appearance preservation: Try the repo's noise-prior option for backgrounds, and tweak `context_overlap` (e.g., 8→16) when using the long config.
 - Throughput: A100-class GPUs can set `context_batch_size > 1` to parallelize denoising for long clips.
-

References

- Project: <https://unianimate.github.io/>
- Code: <https://github.com/ali-vilab/UniAnimate>
- Paper (preprint): arXiv:2406.01188 (UniAnimate)
- Follow-up: UniAnimate-DiT (training/inference code) - <https://github.com/ali-vilab/UniAnimate-DiT>

Notes: Configs, memory figures, and tips reflect the public README at publish time. Evaluate on your own hardware and assets; adjust overlap, noise prior, and resolution for long-form runs.