AInimotion

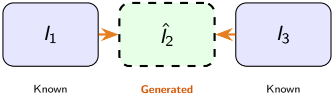
Layer-Separated Deformable Interpolation for Anime Video

Travis Whitney

February 16, 2026

Video Frame Interpolation for Anime

Goal: Generate intermediate frames to increase frame rate (24fps $\rightarrow$ 60fps).



Why anime is uniquely challenging:

1. **Flat color regions** — no texture for optical flow
2. **Non-linear motion** — characters "pop" between poses
3. **Ink line preservation** — thin lines blur easily
4. **Scene cuts** — abrupt shot changes
5. **Layer separation** — backgrounds pan rigidly, characters move independently

Why Deep Learning?

Frames have **multi-scale spatial structure** (edges, textures, objects) and **temporal structure** (motion patterns across frames). Anime adds **layer structure**: rigid backgrounds vs. deformable characters. My model encodes these structural priors directly.

Prior Work & My Contribution

Prior Approaches:

Method	Idea	Gap
DAIN [7]	Depth-aware flow	Not anime-specific
AdaCoF [2]	Deform. kernels	Single motion path
AnimeInterp [1]	Segment match	Ext. segmentation
RIFE [3]	Lightweight flow	Blurs heavy motion
LDMVFI [8]	Latent diffusion	100$\times$ slower

Layer decomposition dates to Wang & Adelson (1994) [6], but **no prior VFI method** builds anime's layer structure into the architecture.

What makes AInimotion novel:

1. **Dual-path architecture** — dedicated BG (affine grid) + FG (AdaCoF kernels)
2. **Domain-informed** — encodes how anime is produced (cel layers)
3. **Learned compositor** — soft α mask, no external segmentation
4. **Edge-aware loss** — Sobel-weighted L1 protects ink lines
5. **Practical** — real-time inference ($\sim$50ms) vs. diffusion ($\sim$5–30s)

ATD-12K Dataset

Source: ATD-12K [1] (CVPR 2021)

- **12,000** frame triplets (I_1, I_2, I_3) from **30 films**
- Motion categories: small, medium, **large**
- Established anime VFI benchmark

Training setup:

- Random 384×384 crops
- Horizontal/vertical/temporal flips
- 100K samples per epoch

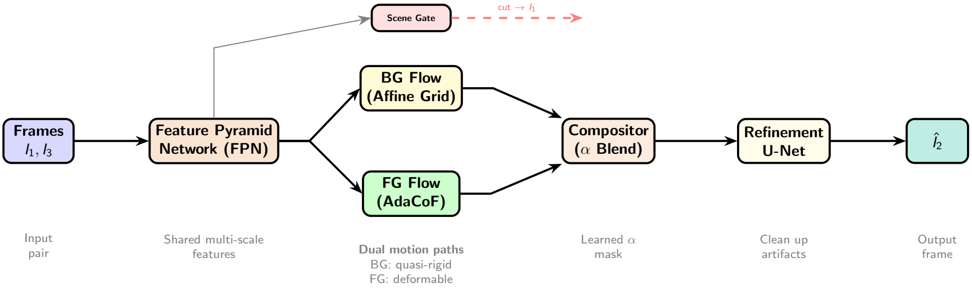
Why This Dataset?

- Purpose-built for anime VFI
- Publicly available
- Rich motion diversity
- Includes difficulty labels

Triplet Format

(I_1, I_2, I_3): I_2 is ground truth used as supervision during training.

Architecture Overview: LayeredInterpolator



Key Components & Loss Functions

Components:

- **FPN:** Shared encoder, 4 scales ($\frac{1}{2}$ to $\frac{1}{16}$) + **correlation volumes** (per-pixel dot-product similarity over a 9×9 search window between frames — tells the network "where did each pixel move?")
- **Scene Gate:** Detects hard cuts via correlation statistics, bypasses interpolation
- **Background:** 8×8 quasi-rigid affine grid for camera pans/zooms
- **Foreground (AdaCoF):** 9×9 deformable sampling kernels per pixel (81 weights + 162 offsets = 243 params/pixel)
- **Compositor:** Learned soft α mask + refinement U-Net

Generator Loss:

$$\mathcal{L}_G = \underbrace{1.5 \cdot \mathcal{L}_{L1}}_{\text{Reconstruction}} + \underbrace{0.1 \cdot \mathcal{L}_{\text{perc}}}_{\text{VGG19}} + \underbrace{1.0 \cdot \mathcal{L}_{\text{edge}}}_{\text{Ink Lines}} + \underbrace{0.005 \cdot \mathcal{L}_{\text{GAN}}}_{\text{Phase 2}}$$

- $\mathcal{L}_{L1}$: Pixel accuracy, drives PSNR
- $\mathcal{L}_{\text{perc}}$: VGG19 feature similarity, preserves style
- $\mathcal{L}_{\text{edge}}$: Sobel-weighted L1, 20× multiplier on edges
- $\mathcal{L}_{\text{GAN}}$: LSGAN patch discriminator with label smoothing (0.1), Phase 2 only

Two-Phase Training Strategy

Phase 1: Reconstruction (Ep 0–34)

- Generator only — no adversarial loss
- $\mathrm{lr}_G = 3 \times 10^{-4}$
- Loss: $\mathcal{L}_{L1} + \mathcal{L}_{\text{perc}} + \mathcal{L}_{\text{edge}}$
- **Goal:** PSNR $\geq$ 28 dB

$\Downarrow$ *D warmup (500 batches)*

Phase 2: GAN Fine-Tuning (Ep 35–49)

- Adversarial loss activated ($\lambda_{\text{GAN}} = 0.005$)
- $\mathrm{lr}_G = 5 \times 10^{-5}$ (reduced)
- D trains every other batch (1:2 ratio)
- **Goal:** perceptual sharpness

Why two phases?

1. Generator learns stable reconstructions *before* seeing adversarial gradients
2. GAN adds sharpness without destabilizing a converged generator

Key stabilization choices:

- **D warmup** — 500 batches before G sees GAN loss
- **Low GAN weight** (0.005) prevents mode collapse
- **Adaptive lr_D** — auto-adjusts based on d_{acc}
- **Label smoothing** (0.1) on discriminator

GAN Stabilization & Training Infrastructure

Discriminator Control:

- **D Warmup** (500 batches) — D learns before G sees adversarial loss
- **Patch Discriminator** — 70×70 patches, not full images
- **Label Smoothing** (0.1) — prevents overconfident D
- **D Update Ratio** (1:2) — D trains every other batch
- **Adaptive lr$_D$** — auto-adjusts based on d_{acc}

Evaluation Metrics:

- **PSNR** (dB): $-10 \log_{10}(\text{MSE})$. Higher $=$ better pixel accuracy. Log scale: $+3$ dB $\approx$ halving MSE.
- **SSIM**: Measures luminance, contrast, and structural similarity (0–1). Catches distortions PSNR misses.
- **Visual comparison**: Side-by-side with ground truth, especially on ink lines and motion.

Infrastructure

Mixed precision, grad_clip=1.0, OOM recovery, W&B logging. RTX 5090 (32 GB VRAM).

Design Evolution & Training Progress

Component	Initial	Final
Base Channels	32	96 (3×)
Kernel Size	7	9 (sweep winner)
GAN Training	From epoch 0	Two-phase (ep 35)
L1 Weight	1.0	1.5 (sweep-optimized)
GAN Weight	0.01	0.005 (gentler)
Edge Multiplier	10×	20× (sharper)
Disc. LR	10^{-4}	2×10^{-5}
D Update	1:1	1:2 (G trains more)
D Warmup	None	500 batches

Lesson: GAN training is highly sensitive in anime VFI. The discriminator easily dominates.

Evaluation Baselines (ATD-12K):

Method	PSNR	SSIM
RIFE	∼25–27 dB	∼0.85
AnimeInterp	∼27–29 dB	∼0.88
AInimotion (target)	**≥28 dB**	**≥0.90**

Sweep Results (13 experiments):

- Best: **21.80 dB** (K=9, crop 384, perc=0.1)
- Crop 384 alone: +0.74 dB over 256
- Gradients stable ($\sim$ 0.2–0.5)

Targets

Phase 1: PSNR $\geq$ 28 dB by epoch 34
Phase 2: + perceptual sharpness via GAN

Summary & Next Steps

What I built:

- **LayeredInterpolator** — novel dual-path architecture separating BG/FG motion
- **Edge-aware losses** for ink line preservation
- **Two-phase training** with D warmup for stable GAN integration
- **Robust infrastructure** — OOM recovery, adaptive GAN balancing, mixed precision, W&B logging

Next steps:

- Complete full 50-epoch training run
- Evaluate Phase 2 GAN fine-tuning
- Qualitative evaluation on test set
- Compare with RIFE, AnimeInterp baselines

Project Repository

`github.com/TWhit229/AInimotion`

References I

- L. Siyao, S. Zhao, W. Yu, W. Sun, D. Metaxas, C. Loy, Z. Liu.
  "Deep Animation Video Interpolation in the Wild."
  CVPR, 2021.
- H. Lee, T. Kim, T. Chung, D. Pak, Y. Ban, S. Lee.
  "AdaCoF: Adaptive Collaboration of Flows for Video Frame Interpolation."
  CVPR, 2020.
- Z. Huang, T. Zhang, W. Heng, B. Shi, S. Zhou.
  "Real-Time Intermediate Flow Estimation for Video Frame Interpolation."
  ECCV, 2022.
- I. Goodfellow et al.
  "Generative Adversarial Nets."
  NeurIPS, 2014.

References II

X. Mao, Q. Li, H. Xie, R. Lau, Z. Wang, S. Smolley.
"Least Squares Generative Adversarial Networks."
ICCV, 2017.

J. Wang, E. Adelson.
"Representing Moving Images with Layers."
IEEE Trans. Image Processing, 1994.

W. Bao, W. Lai, C. Ma, X. Zhang, Z. Gao, M. Yang.
"Depth-Aware Video Frame Interpolation."
CVPR, 2019.

D. Danier, F. Zhang, D. Bull.
"LDMVFI: Video Frame Interpolation with Latent Diffusion Models."
AAAI, 2023.