Wan-Alpha:
High-Quality Text-to-Video Generation with Alpha Channel

Dong, H., Wang, W., Li, C., & Lin, D
Tianjin University, China 25.09

Abstract

- **RGBA 비디오 생성은** 일반적인 RGB 색상 채널에 투명도를 나타내는 alpha 채널을 추가한 형태임.
- Wan-Alpha 는 RGB 와 alpha 채널을 **jointly 하게 학습하여** 비디오를 생성하는 새로운 프레임워크를 제안함.
- **VAE 가 alpha 채널을 인코딩해서 RGB latent space 에 임베딩**하도록 설계함.
- 고품질의 다양한 RGBA 비디오 데이터셋을 직접 구축함.

Figure 1: Qualitative results of video generation using Wan-Alpha. Our model successfully generates various scenes with accurate and clearly rendered transparency. Notably, it can synthesize diverse semi-transparent objects, glowing effects, and fine-grained details such as hair.

1. Introduction

- T2V(텍스트-비디오 생성)는 다양한 산업(영화, TV 제작, 게임, 가상 현실, 소셜미디어, 광고)에서 상당한 잠재력을 입증함.
    - RGBA 비디오는 다양한 어플리케이션(비디오 편집, 게임 발전, 소셜 미디어 콘텐츠)에서 유용하지만, 정작 RGBA 비디오 생성은 많은 주목을 받지 못했음.

- 가장 큰 이유는 **RGBA 데이터가 RGB 데이터보다 훨씬 희귀하고 수집하기 어려워** 대규모 RGBA 비디오 생성 모델을 학습하는 것이 챌린지가 되었음.
    - 연구자들은 pretrained RGB 생성 모델을 활용하는 방법을 시도함.
    - **LayerDiffuse**: 첫 번째 RGBA 이미지 생성 모델로, VAE를 사용해 투명도 정보를 RGB latent space 에 임베딩했음.(LoRA 학습 방식 적용)
    - **AlphaVAE**: RGBA VAE를 개선하면서 학습 데이터 수를 100만 장에서 8천 장으로 줄였음.
    - **Alfie**: training-free 방법 제안함.(attention maps 에서 alpha 값을 직접 추론했음.)
        - 하지만, 작은 객체는 잘 생성하지 못한 한계가 있었음.

- 그 뒤에도 multi-layer 생성 방식이나 inpainting 방식으로 투명도를 생성하는 연구가 지속됨.
    - **TransPixeler**: 최신 SOTA 모델로 alpha tokens 개념을 도입하고 백본 네트워크를 복제했으며 cross-RGBA attention 방식을 적용했음.
        - **하지만, inference cost가 두 배로 증가했고, 결과도 만족스럽지 못했음.**

1. Introduction

- Wan-Alpha
  - **고품질 데이터셋 구축**
    - Wan 모델을 베이스 모델로 선정하고, visual quality 향상에 초점을 맞췄음.
    - 고해상도, 부드러운 움직임, 다양한 콘텐츠를 보장하기 위해 여러 출처에서 데이터를 수집했음.
    - **semantic consistency** 를 유지하기 위해 비디오 콘텐츠랑 프롬프트가 정확히 일치하는지 확인함.
  - **간단하고 효율적인 학습 및 추론 과정**
    - **LayerDiffuse** 전략을 채택했음.(VAE 를 학습하여 alpha 정보를 RGB latent space 에 임베딩함.)
    - **Feature merge block** 을 causal 3D 모듈과 함께 사용해 **RGB 와 alpha latent feature를 효과적으로 결합함**.
    - RGB/alpha decoder는 **LoRA** 방식을, DiT(diffusion transformer)는 **DoRA** 모듈을 적용하여 학습함.
    - inference 에서는 모델을 두 개의 LoRA 모듈(VAE decoders) 및 DoRA 모듈(DiT)을 로드함.
    - TransPixeler 와 비교해 추가 연산 비용 없이**, 4-step 에서도 CFG 없이 inference 가능함**.

2.1 VAE

- Wan-Alpha 는 LayerDiffuse 의 접근 방식을 따라, **RGB-alpha VAE 가 alpha 채널을 RGB VAE 와 동일한 latent space 로 인코딩하도록** 설계했음.
  - LayerDiffuse 에서는 별도의 encoder 와 decoder 를 위한 UNet 구조를 사용하는데, 이는 비디오 도메인에서 더 학습하기 어려운 문제가 있음.
    - 이를 해결하기 위해 **pretrained VAE를 복제하여** 사용하는 전략을 채택함.
  - encoder 에는 RGB-alpha feature merging block를 도입했고, RGB decoder 및 alpha decoder를 효율적으로 조정하기 위해 **LoRA**를 사용했음.
  - 또한 VAE 학습을 더욱 효과적으로 수행하기 위해 렌더링 기법(soft/hard rendering) 및 loss function 을 도입했음.

2.1 VAE

- **Training – 데이터 전처리**
  - soft render $\mathbb{R}^s$, hard render $\mathbb{R}^h$
    - **Soft rendering**: alpha 비디오 값이 [0,1] 범위에 속하는 과정을 의미하며, 이는 일반적으로 **RGBA 비디오의 배경을 자연스럽게 합성하는데** 사용됨.
    - **Hard rendering**: 반대로, **0이 아닌 모든 alpha 값을 1로 설정하고** 완전히 투명하지 않은 영역은 RGB 채널의 원본 색상을 그대로 유지함.

$$\mathbb{R}^s = V_{rgb} \cdot \alpha + c \cdot (1 - \alpha),$$

$$\mathbb{R}^h = V_{rgb} \cdot \alpha + c \cdot (1 - \alpha), \quad \alpha = \begin{cases} 1 & \text{if } \alpha > 0, \\ 0 & \text{otherwise.} \end{cases}$$

    - c: 사전에 정의된 colors set c 중 랜덤 색상임.
      - black, blue, green, cyan, red, magenta, yellow, white

- **VAE encoder가 RGB 배경색과 투명도를 혼동하는것을 방지하기 위해**,
  color set C 로 부터 랜덤하게 color $\bar{c}$ 를 선택하고,
  RGB 비디오 V_{rgb}를 **hard-render** 하여 rendered 비디오 $\bar{V}_{rgb}$ 를 생성함.
    - $\bar{V}_{rgb} = \mathbb{R}^h(V_{rgb}, V_\alpha, \bar{c})$

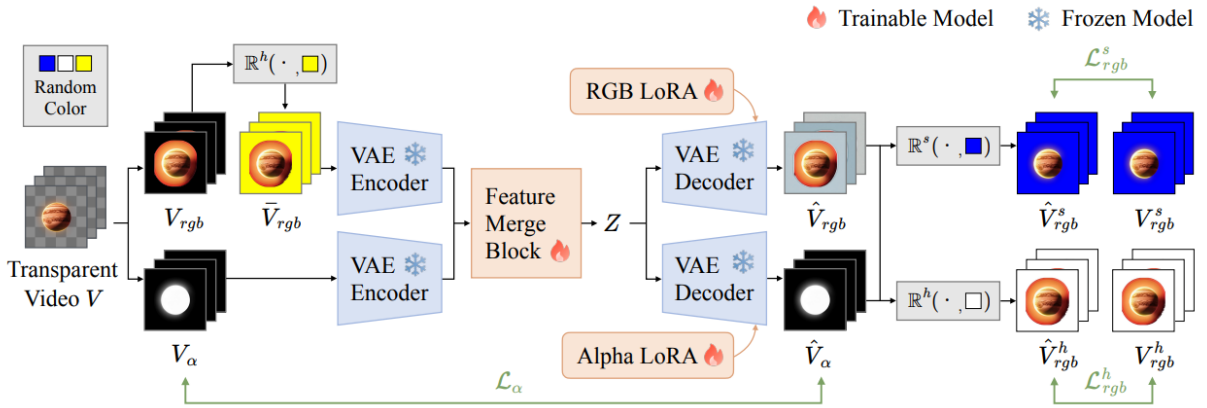
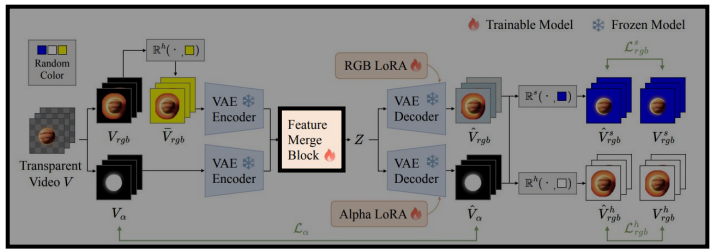


Figure 2: The overall architecture of VAE. The transparent video V is split into the RGB video V_{rgb} and the alpha video V_α. These are then fed into the frozen VAE encoder, where the random hard-rendered video $\bar{V}_{rgb}$ and V_α produce two distinct feature representations. Next, a feature merging block combines these features to generate the latent feature Z. Finally, Z is passed into both the RGB and alpha VAE decoders, which predict the RGB video $\hat{V}_{rgb}$ and the alpha video $\hat{V}_\alpha$.

2.1 VAE



- **Training – RGB ⊕ Alpha 통합 모듈 설계**
  - frozen Wan-VAE encoder $\mathcal{E}$ 사용하여 $\bar{V}_{rgb}, V_\alpha$ 를 인코딩 후 feature merging block $\mathcal{M}$ 에 입력함
  - feature merging block $\mathcal{M}$
    - 일반적인 3D convolution 과 비교했을 때, causal 3D convolution 은 연산 비용이 더 적고 더 긴 비디오를 인코딩할 수 있음.
      - **미래 프레임 정보를 사용하지 않고, 과거, 현재 프레임 정보만 활용함**
    - causal residual blocks와 attention layers 를 거쳐 latent z 를 생성함.

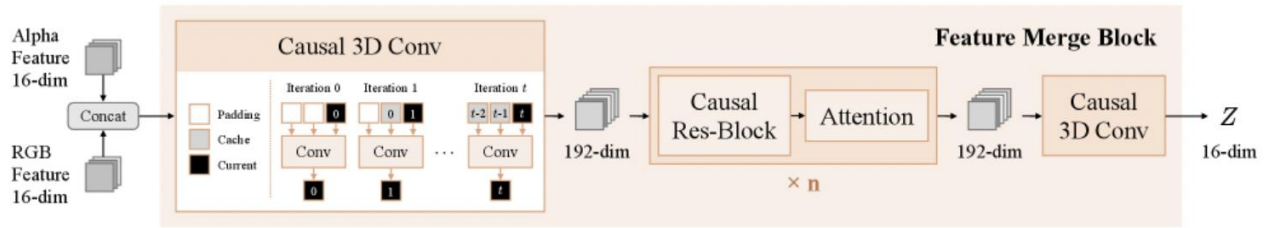


Figure 3: The detailed architecture for the feature merge block. RGB and alpha features are concatenated and fused by a causal 3D convolution. Then, we use several causal residual blocks and attention layers. Finally, a causal 3D convolution generates the merged latent Z.

- latent z 가 frozen Wan-VAE decoder 에 입력되어 RGB video $\hat{V}_{rgb}$, alpha video $\hat{V}_\alpha$ 를 생성함.

$$\hat{V}_{rgb} = \mathcal{D}_{\text{w/RGB LoRA}}(Z), \quad \hat{V}_\alpha = \mathcal{D}_{\text{w/ Alpha LoRA}}(Z).$$

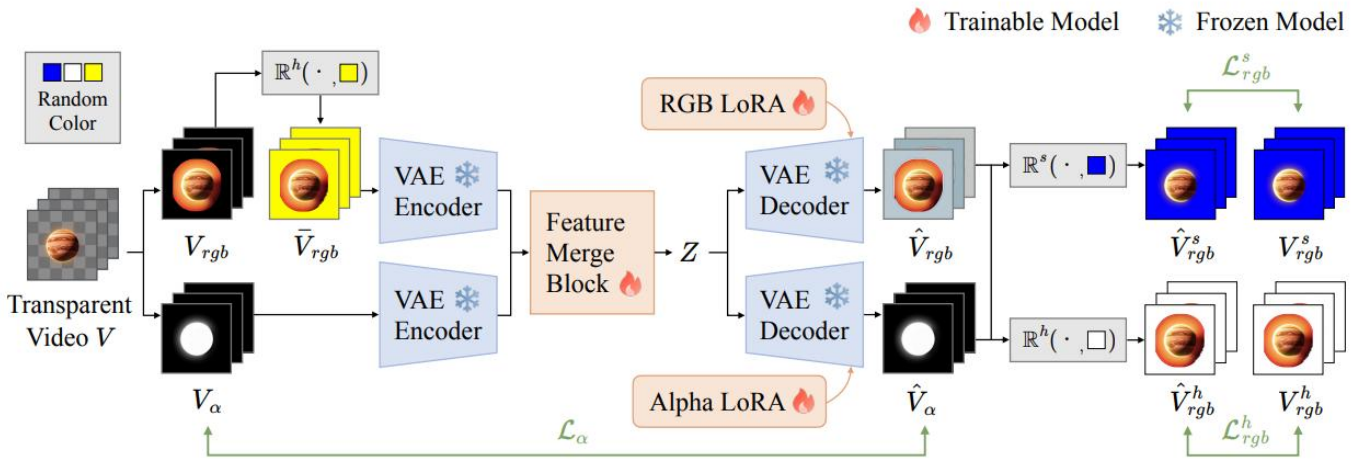
2.1 VAE

- Training Objectives

1. reconstruction loss $\mathcal{L}_{rec}\left(\hat{V}_\alpha, V_\alpha\right) = \left\|\hat{V}_\alpha - V_\alpha\right\|.$

2. perceptual loss $\mathcal{L}_{per}\left(\hat{V}_\alpha, V_\alpha\right) = \left\|\Phi\left(\hat{V}_\alpha\right) - \Phi\left(V_\alpha\right)\right\|_2$
   - VGG 네트워크 사용

3. edge loss $\mathcal{L}_{\text{edge}}\left(\hat{V}_\alpha, V_\alpha\right) = \left\|S\left(\hat{V}_\alpha\right) - S\left(V_\alpha\right)\right\|.$
   - sobel 필터 사용



$$\mathcal{L}_\alpha = \mathcal{L}_{rec}\left(\hat{V}_\alpha, V_\alpha\right) + \mathcal{L}_{per}\left(\hat{V}_\alpha, V_\alpha\right) + \mathcal{L}_{edge}\left(\hat{V}_\alpha, V_\alpha\right)$$

$$\mathcal{L}^s_{rgb} = \mathcal{L}_{rec}\left(\hat{V}^s_{rgb}, V^s_{rgb}\right) + \mathcal{L}_{per}\left(\hat{V}^s_{rgb}, V^s_{rgb}\right) + \mathcal{L}_{edge}\left(\hat{V}^s_{rgb}, V^s_{rgb}\right),$$
$$\mathcal{L}^h_{rgb} = \mathcal{L}_{rec}\left(\hat{V}^h_{rgb}, V^h_{rgb}\right) + \mathcal{L}_{per}\left(\hat{V}^h_{rgb}, V^h_{rgb}\right) + \mathcal{L}_{edge}\left(\hat{V}^h_{rgb}, V^h_{rgb}\right).$$

$$\mathcal{L}_{vae} = \mathcal{L}_\alpha + \mathcal{L}^s_{rgb} + \mathcal{L}^h_{rgb}.$$

2.2 Text-to-Video Generation

- VAE를 통해 **RGBA** 비디오를 RGB 비디오의 latent space와 동일하게 매핑할 수 있게 되었음.
  - 따라서 기존의 WAN 모델과 같은 DiT(diffusion transformer) 를 사용할 수 있게됨.
- 목표: **생성 비디오의 품질을 향상시켜야함.**
- Architecture

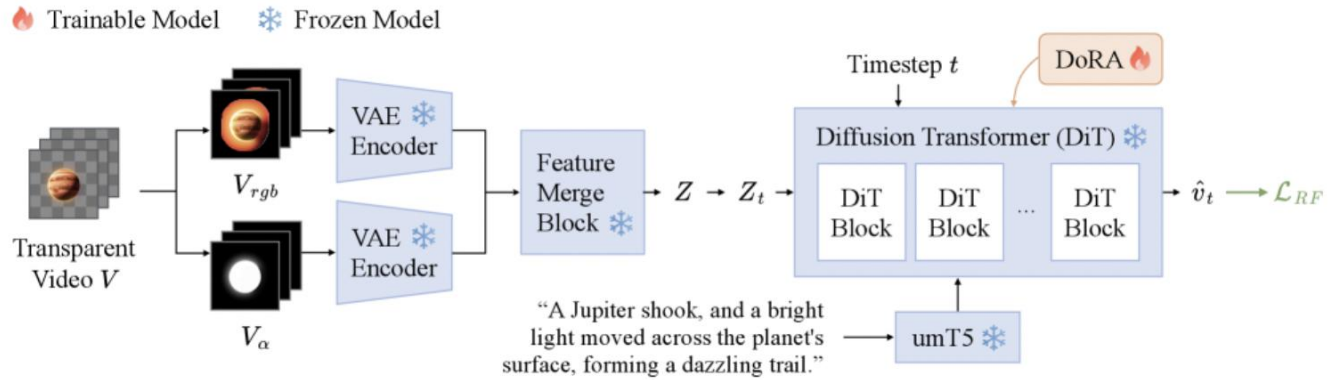


Figure 4: Our text-to-video generation training framework. Transparent videos are first encoded into latents using our VAE, and then a diffusion transformer is trained on these latents using DoRA.

2.2 Text-to-Video Generation

- Tranining Objectives
  - WAN 모델이 rectified flow 기반으로 구성됨
  - 따라서 rectified flow loss 인 vector field 를 학습함.
  - rectified flow
    - 원본 데이터를 노이즈(타겟 분포)로 변하는 과정을 직선 경로로 정의했음
    - 직선 경로 정의: 데이터와 노이즈를 선형 보간

$$Z_t = t\epsilon + (1-t)Z$$

    - loss: 특정 시점 t 에서의 속도 예측을 학습한다.
      - 특정 시점 t 일 때, 속도(어떤 방향으로 움직여야 하는지)를 예측한다.

$$\mathcal{L}_{t2v} = \left\| \hat{v}_t - v_t \right\|^2$$

      - 따라서, Z_t 를 t 로 미분하게 되면 v_t 속도를 얻을 수 있다. $v_t = \epsilon - Z$
      - 모델은 특정 시점 t 와 위치 Z_t 만 가지고 속도를 예측한다.

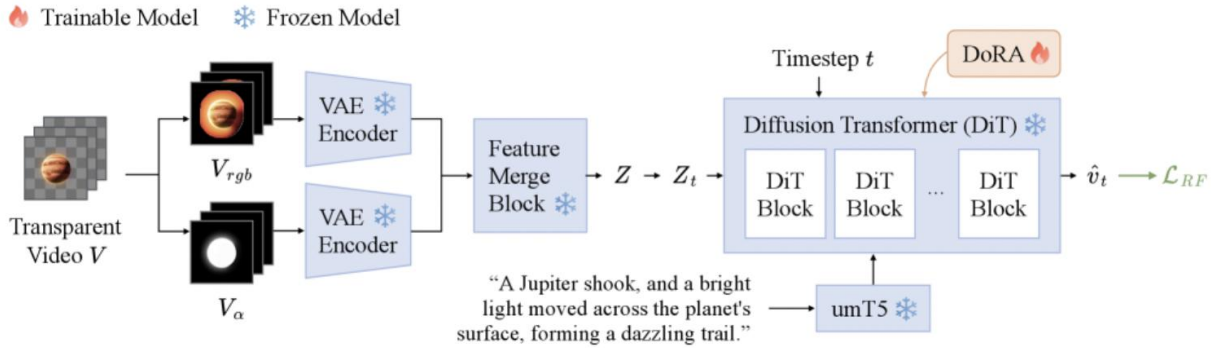


Figure 4: Our text-to-video generation training framework. Transparent videos are first encoded into latents using our VAE, and then a diffusion transformer is trained on these latents using DoRA.

2.2 Text-to-Video Generation

- **Dataset**
  - 데이터 수집
    - image matting 데이터 셋에서 수집(AIM-500, AM-2K, Distinctions-646, ,,,)
      - 윈도우 슬라이딩하여 움직임을 시뮬레이션으로 비디오 데이터처럼 활용할 수 있도록 구축함.
    - video dataset(DVM, VideoMatting108, VideoMatte240K)
  - 데이터 선정
    - 고품질 데이터를 확보하기 위해 선별 과정 거침.
      - **명확한 모션, 반투명 객체, 조명 효과가** 뚜렷한 데이터 선택함.
  - 캡션 생성 과정
    - Qwen2.5-VL-72B 모델 사용해 짧은 캡션과 긴 캡션을 자동으로 생성하고, 수동으로 수정했음.
    - 모든 캡션은 중국어로 작성되었지만, Wan 모델의 다국어 지원 능력 덕분에 중국어 데이터만으로 학습했어도 영어 프롬프트로 RGBA 비디오 생성 가능했음.
  - 추가 라벨링
    - 수동으로 모션 속도, 예술적 스타일, 샷 크기, 시각적 품질 문제 등 라벨을 부여함.
    - 캡션의 앞 또는 뒤에 랜덤으로 삽입했음.

2.2 Text-to-Video Generation

- **Inference**
  - 기존의 Wan 모델을 기반으로 사용자는 오직 **두 개의 VAE decoder LoRAs 와 T2V DoRA** 만 로드하면 됨.
  - LoRA/DoRA weights 는 완전히 베이스 모델에 병합 가능하며 추가 연산 비용 없음.
  - inference pipeline 은 간단한 변화가 있었음.
    - decoder 를 복제하여 하나는 RGB 채널을, 다른 하나는 alpha 채널을 생성하도록 구성함.
    - 구조가 단순해지고 배포 및 가속화가 쉬워졌음.
  - 가속화는 LightX2V(CFG 없이, 4 steps 로 high-quality 결과를 생성할 수 있음)을 사용했음.

3.2 Comparisons

- **비교 대상 모델**
  - TransPixeler
    - open: CogVideoX-5B 기반 오픈소스 모델
    - close: Adobe Firefly 사용
  - TransPixeler 에 사용된 테스트 프롬프트로 비교
- **결과**
  - Visual quality, 움직임 일관성(Motion Consistency) 우수함
  - 알파 채널 경계의 선명도
    - 알파 채널 경계선이 선명하고 정확하게 표현됨.
    - 머리카락과 같은 미세한 디테일에서 두드러짐.
  - 반투명 효과
    - 액체, 연기를 더 현실적으로 생성했음.
  - 추론 속도: 15배 빠름 (480x832, 81 프레임)

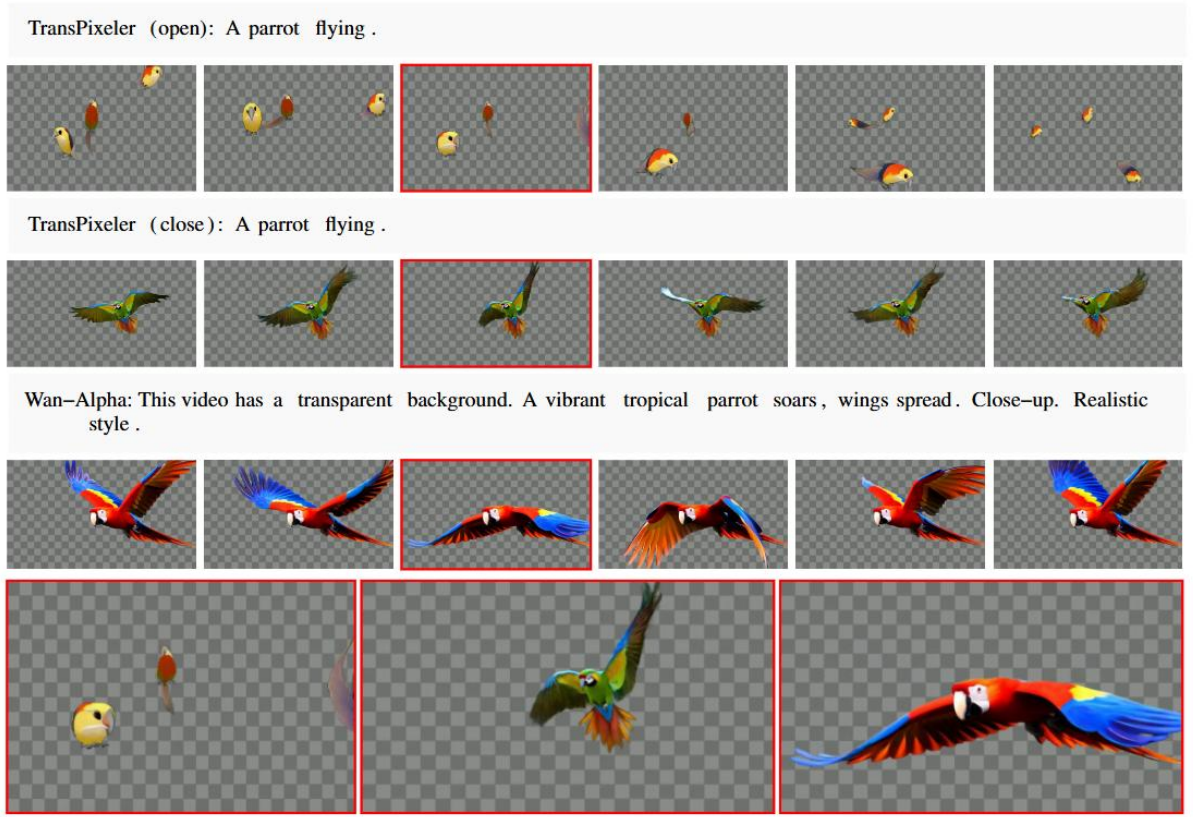
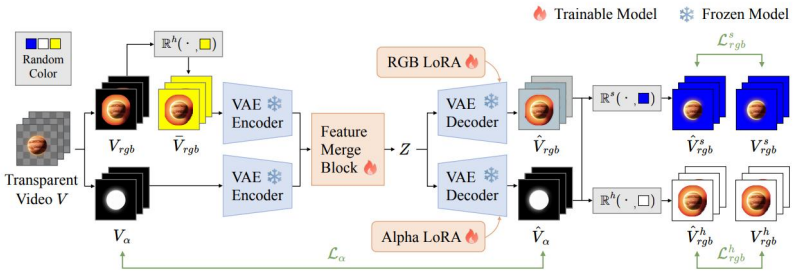


Figure 10: The video generated by TransPixeler (open), TransPixeler (close), and Wan-Alpha. At the bottom of the figure, we provide the visualization of one frame from each of the three videos, respectively.

Wan-Alpha: High-Quality Text-to-Video Generation with Alpha Channel

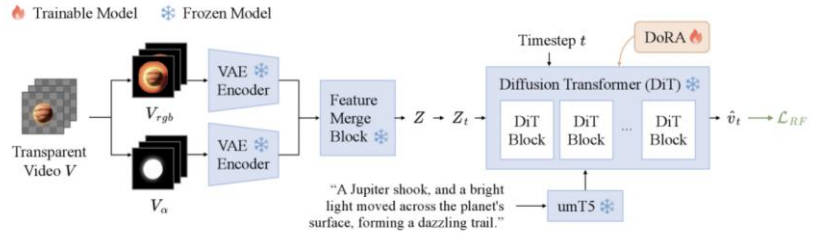
- 알파 채널을 포함하여 투명도를 표현하는 RGBA 비디오를 **고품질**로 생성하는 T2V 모델

1. 알파채널을 RGB latent 로 임베딩: VAE training



- **Feature merging block 설계: RGB 와 Alpha 채널의 특징을 통합**
- loss: reconstruction + perceptual + edge
  - 학습 대상: feature merge block, RGB/Alpha LoRA
  - 학습 전처리: **soft/hard rendering**

2. Visual quality 를 높이기 위한 Wan Fine-tuning



- **앞서 VAE로 알파 채널을 RGB latent 에 매핑할 수 있었음**
  - Wan 기반 모델을 사용할 수 있게 되어 생성 능력을 유지할 수 있었음
- loss: rectified flow
  - 학습 대상: DiT DoRA

- Wan-Alpha는 TransPixeler와 비교하여 **시각적 품질**(움직임 일관성, alpha edge sharpness, 반투명 효과)에서 **우수한 성능을 보였음.**
- TransPixeler (open)보다 약 **15배 빠른 추론 속도를 달성**, 고품질 비디오를 **효율적으로 생성**할 수 있음을 입증함.
- Wan-Alpha는 반투명 객체, 빛나는 효과, 머리카락과 같은 미세한 디테일 등 다양한 semi-transparent object와 효과를 포함하는 고품질 비디오를 성공적으로 생성함.