STATDM: Scene Text Aware style Transfer Diffusion Model

목차

1. 문제 정의

- Image Prompt를 이용하여, style-transfer 및 Scene Text Editing을 동시에 진행하는 새로운 Task를 정의하고 이를 해결해보자!
  - Style-transfer를 할 때 scene-text 가 있다면, 해당 scene text는 왜곡..
  - Scene-text editing을 한다면, style-transfer 불가능한 상태..
    - 그렇다고 두 모델을 단순히 붙어서 쓰기에는 diffusion을 두 번 사용, 많은 하이퍼파라미터와 시간 소요

1. Style-Transfer ~~with scene-text editing~~

a cat

I want to edit Scene Text..

change the word 'INVASION' in the picture to 'MLVU Project'

1. Scene-Text Editing models ~~with style-transfer~~

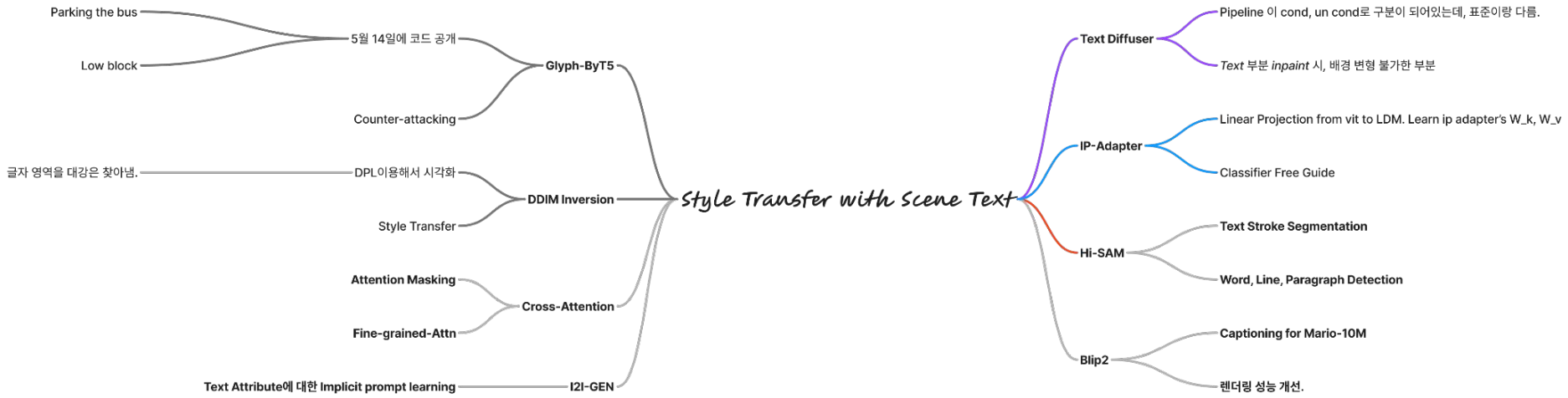
I want this style..

a cat

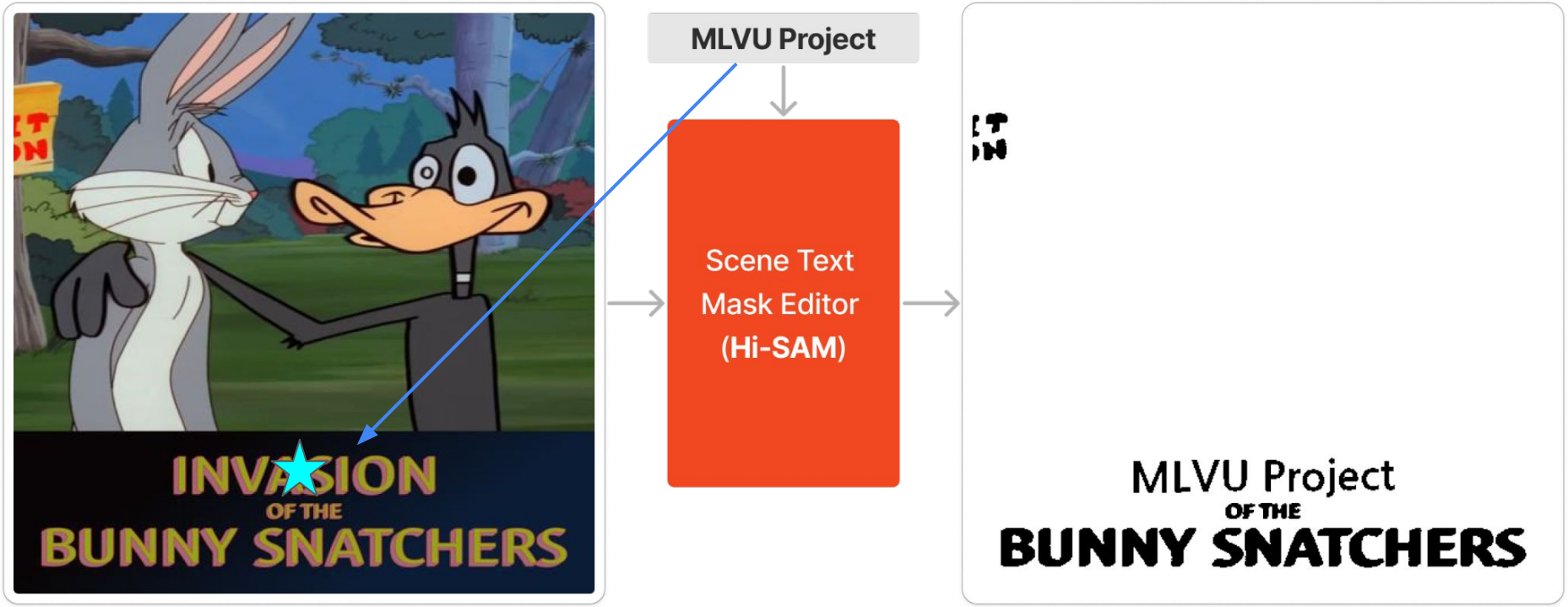
1. 해결 방안 및 Contributions

- Image Style transfer와 scene-text editing을 동시에 해내는 아키텍처 구현
  - **Style-transfer(IP-Adapter) + Scene Text Editing(TextDiffuser) 통합에 성공**
    - **+) Hi-SAM(Segment Anything Model 기반)을 이용**
  - **이후, Baseline 구현체의 문제점을 개선하자.**
  - cross attention map 시각화 및 분석, U-Net block의 특성 분석
    - cross attention masks 도입 (Scene Text Region과 Background 영역 분리)
    - selective style injection 전략 고안 (U-Net 블럭의 특성 고려)
  - End-to-End Model을 및 성능 개선을 위한 Glyph word encoder 학습 방안 연구
    - Glyph-word encoder 아키텍처 구상
    - Glyph image - word text pair 1M 데이터셋 구축 (다양한 glyph rendering 조건 고려)

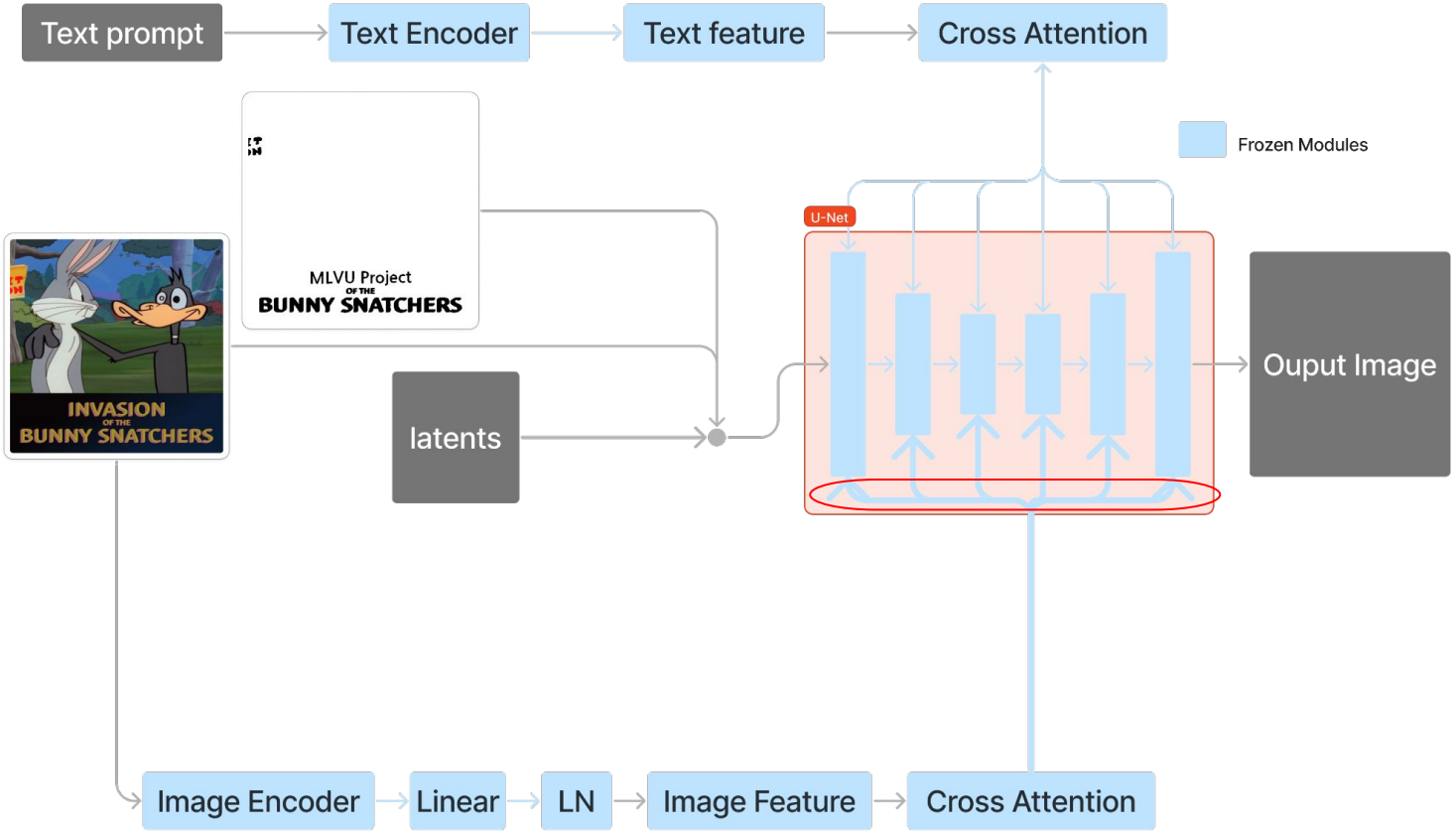
2. 관련 연구: 다양한 시도


Style Transfer with Scene Text
Glyph-ByT5
5월 14일에 코드 공개
Parking the bus
Low block
Counter-attacking
DDIM Inversion
DPL이용해서 시각화
글자 영역을 대강은 찾아냄.
Style Transfer
Cross-Attention
Attention Masking
Fine-grained-Attn
I2I-GEN
Text Attribute에 대한 Implicit prompt learning
Text Diffuser
Pipeline 이 cond, un cond로 구분이 되어있는데, 표준이랑 다름.
Text 부분 inpaint 시, 배경 변형 불가한 부분
IP-Adapter
Linear Projection from vit to LDM. Learn ip adapter's W_k, W_v
Classifier Free Guide
Hi-SAM
Text Stroke Segmentation
Word, Line, Paragraph Detection
Blip2
Captioning for Mario-10M
렌더링 성능 개선.

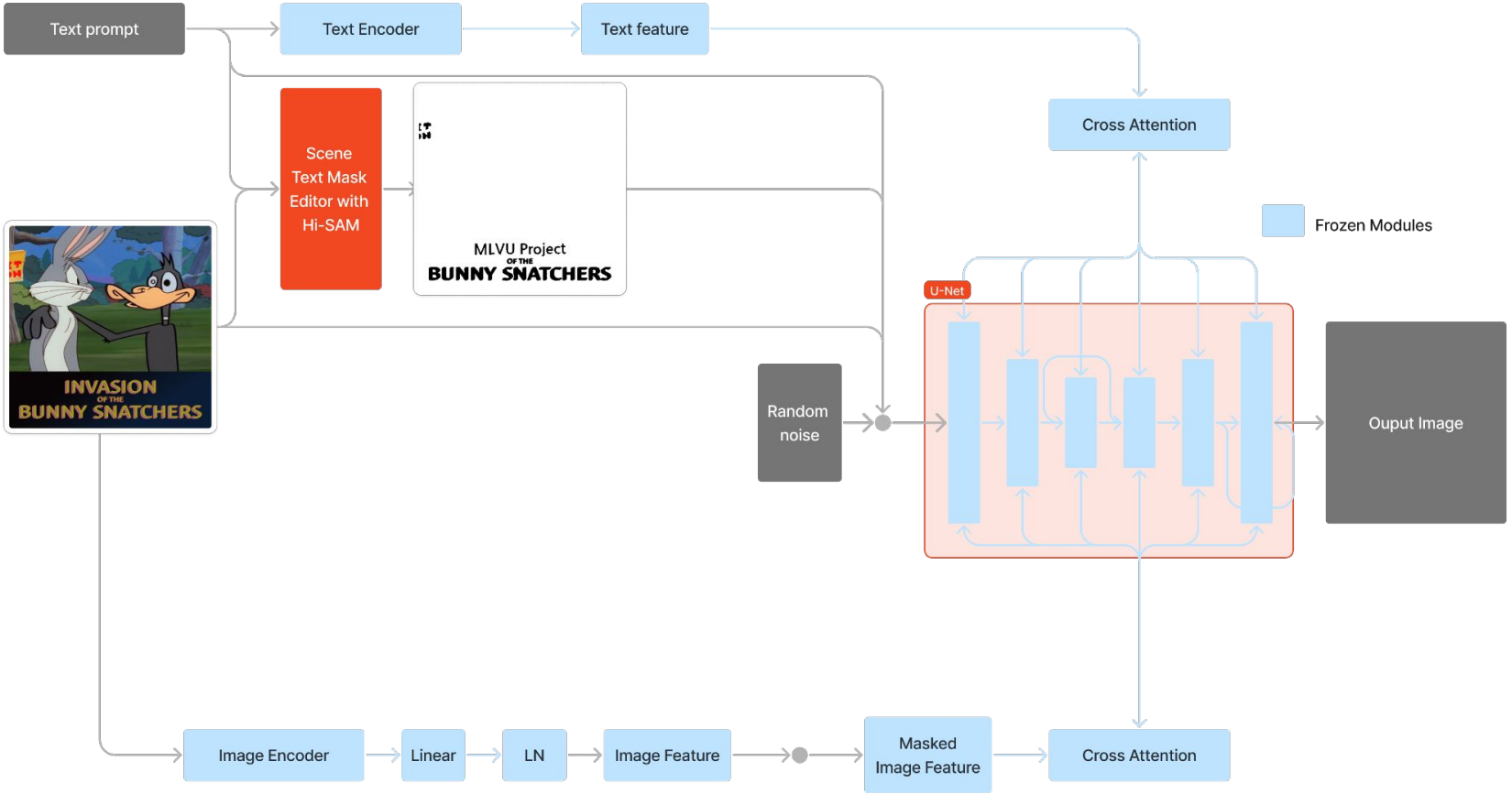
3. 초기 아키텍처 구성: 1) Scene Text Segmentation & Mask Editing



3. 초기 아키텍처 구성: 2) Novel Style-transfer with scene-text editor



3. 초기 아키텍처 구성: 전체 아키텍쳐



4. 초기 아키텍처 문제 분석: 문제 상황

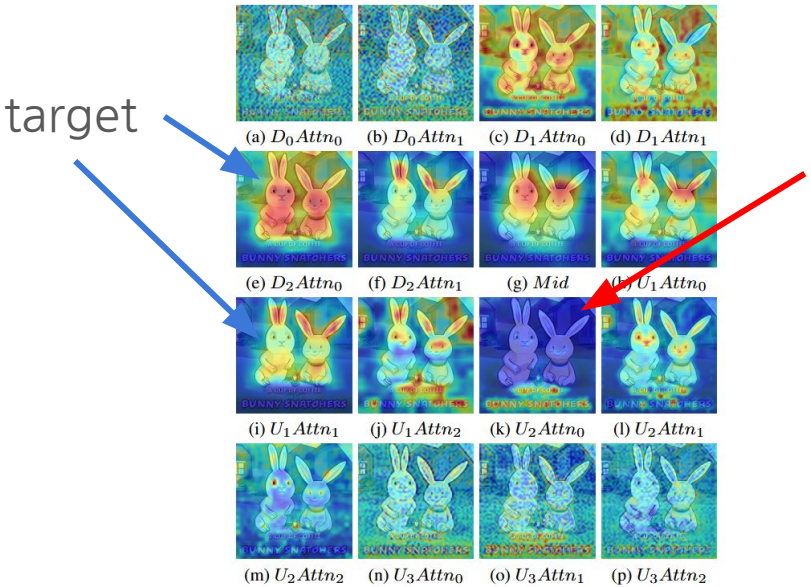
- **흐린 결과물**, 이미지 특성이 **entangled**되는 문제

(a) Image prompt (b) Output 1 (c) Output 2

Figure 6. Examples of blurry and entangled outputs from the prompt 'two dogs' and an image prompt 6a. Style transfer was applied with intensities of 0.4 and 0.7, respectively.

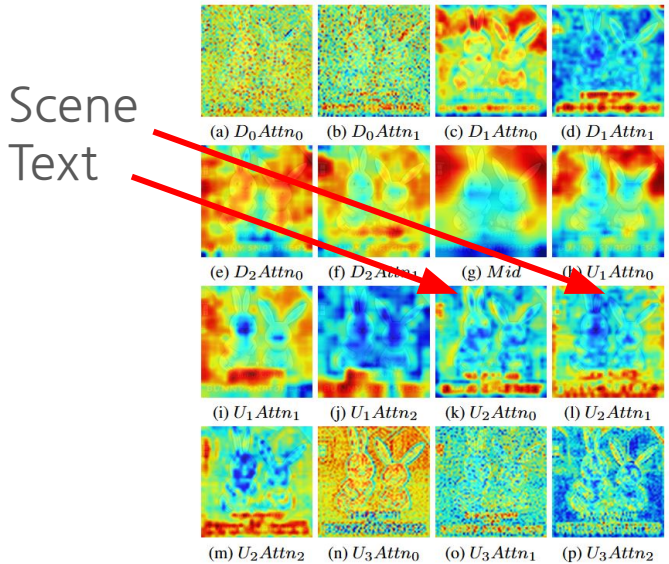
4. 초기 아키텍처: 문제 원인 분석

- Cross attention mapping 분석



Query: from Latent in diffusion U-Net
Key, Value: from text embeddings

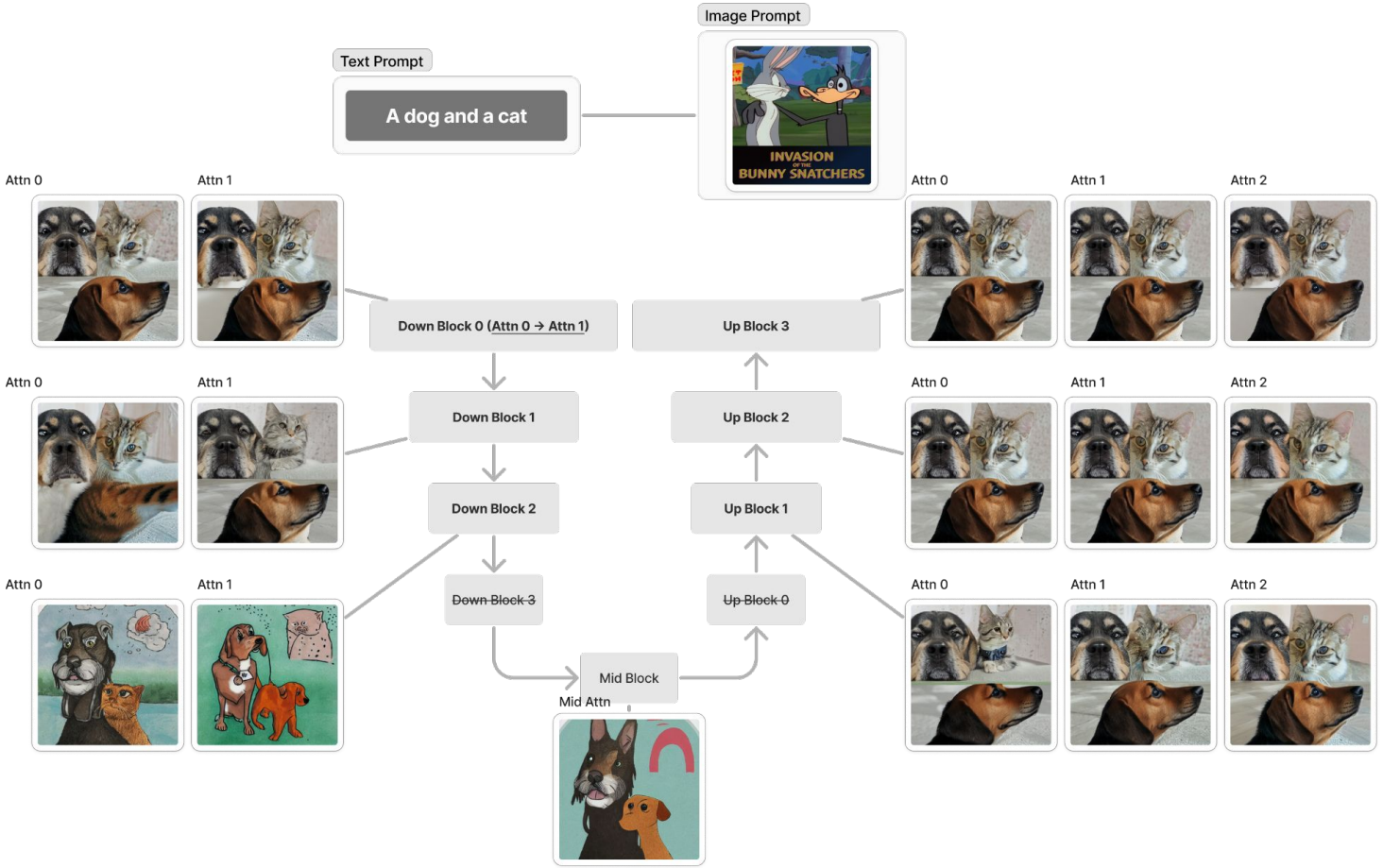
Figure 6. Attention weights by "rabbits" prompt in our baseline's U-Net blocks (D_0Attn_0 means that Down block 0 and internal attention 0. U_1Attn_0 means that Up block 1 and internal attention 0)



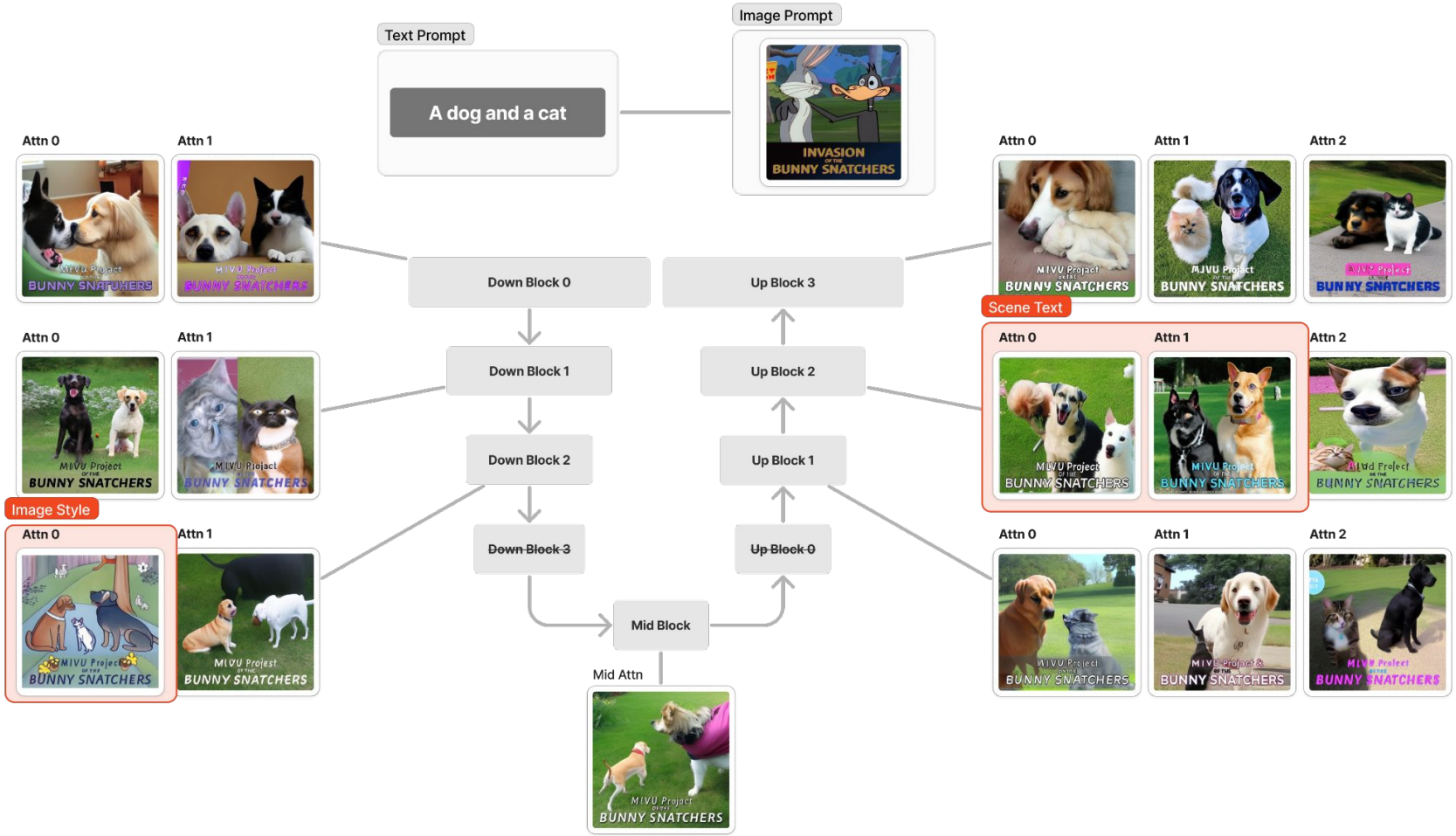
Query: from Latent in diffusion U-Net
Key, Value: from image embeddings

Figure 7. Attention weights of the first IP-Adapter token in U-Net blocks (D_0Attn_0 means that Down block 0 and internal attention 0. U_1Attn_0 means that Up block 1 and internal attention 0)

4. 초기 아키텍처: 문제 원인 분석 (SD 1.5 + Style Transfer)

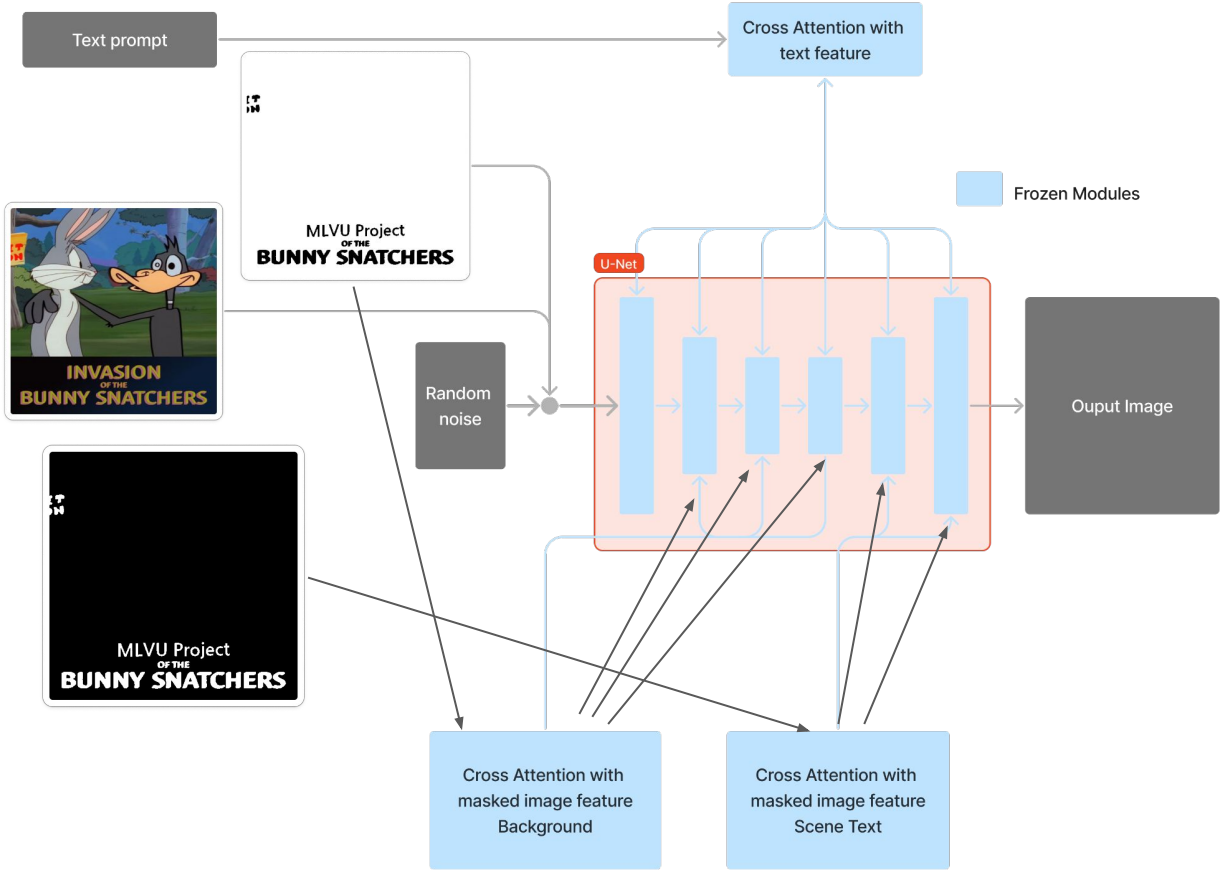


4. 초기 아키텍처: 문제 원인 분석 (Our Initial architecture)

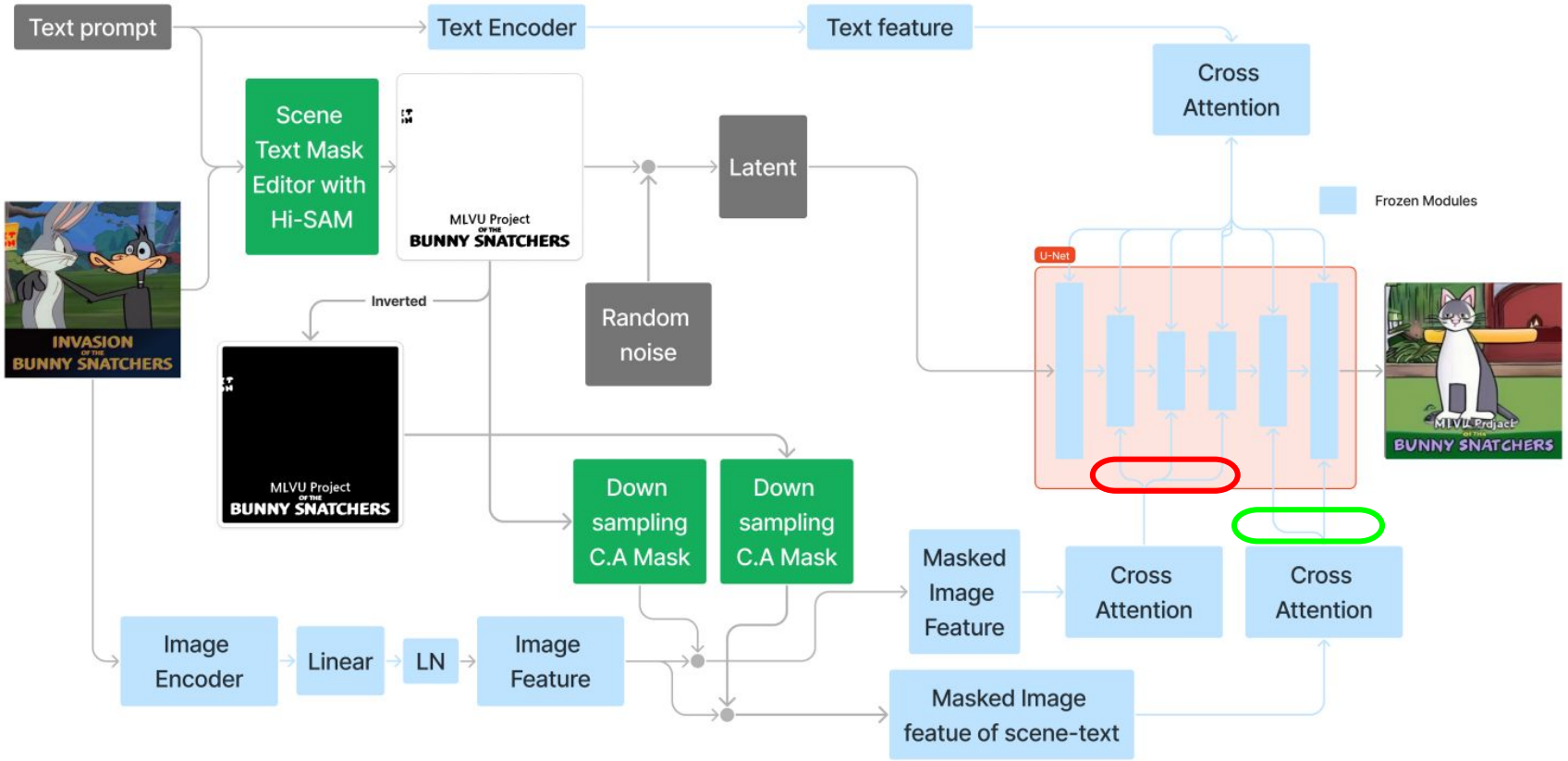


4. 초기 아키텍처: 문제 해결 (Cross Attention Mask + Selective Style Injection)

- Scene-text 부분과 background Image 부분을 각각 다른 U-net block에서 cross attention 적용



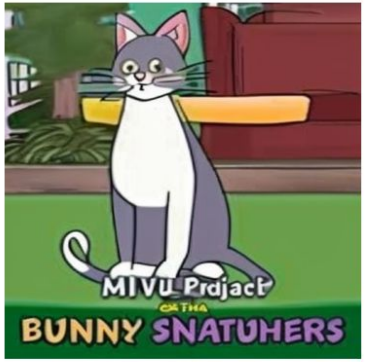
5. 최종 아키텍처 구성


Text prompt
Text Encoder
Text feature
Cross Attention
Scene Text Mask Editor with Hi-SAM
Latent
Frozen Modules
U-Net
Inverted
Random noise
Down sampling C.A Mask
Down sampling C.A Mask
Masked Image Feature
Cross Attention
Cross Attention
Image Encoder
Linear
LN
Image Feature
Masked Image featue of scene-text

6. 실험결과: Baselines components와의 비교

Figure 9. Comparison Result with our final pipeline and others

6. 실험결과: Ablation study



(a) w/o cross attention mask, w/ selective injection

(b) w/ cross attention mask, w/ selective injection

(c) w/ cross attention mask, w/o selective injection

Figure 10. Effect of separated cross attention masks and selective style injection. (Text prompt: ”a cat”, Image Prompt: Fig. 4a)

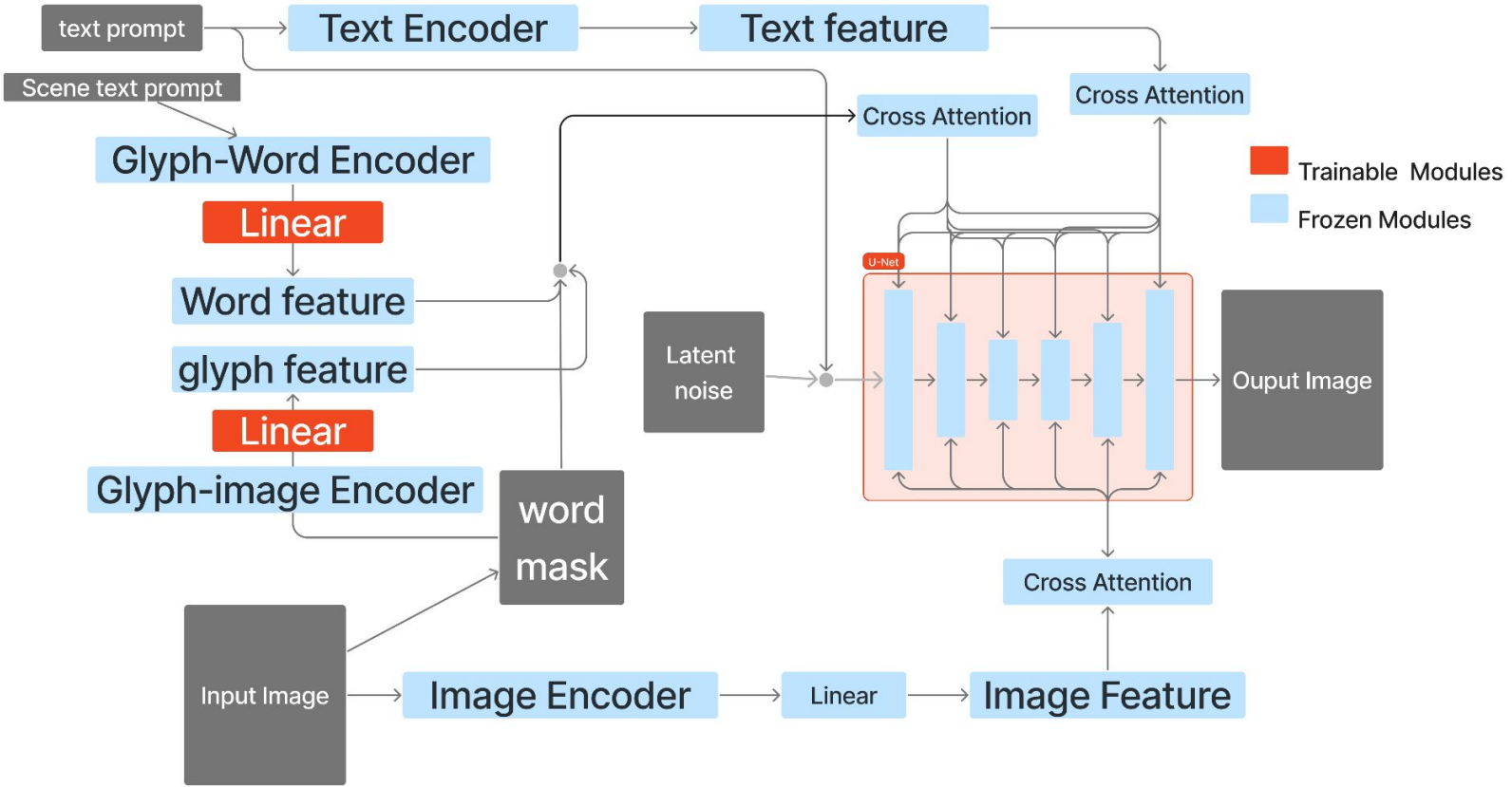
해결 방안 및 Contributions

- Image Style transfer와 scene-text editing을 동시에 해내는 아키텍처 구현
  - Style-transfer(IP-Adapter) + Scene Text Editing(TextDiffuser) 통합에 성공
    - +) Hi-SAM(Segment Anything Model 기반)을 이용
  - 이후, Baseline 구현체의 문제점을 개선하자.
  - **cross attention map 시각화 및 분석, U-Net block의 특성 분석**
    - **cross attention masks 도입 (Scene Text Region과 Background 영역 분리)**
    - **selective style injection 전략 고안 (U-Net 블럭의 특성 고려)**
  - **End-to-End Model을 및 성능 개선을 위한 Glyph word encoder 학습 방안 연구**
    - **Glyph-word encoder 아키텍처 구상**
    - **Glyph image - word text pair 1M 데이터셋 구축 (다양한 glyph rendering 조건 고려)**

7. 한계점

- Text prompt와 Image prompt 조합에 따라 최적의 style transfer scale 다름.
  - 현재는 수동으로 튜닝이 필요
  - 각 프롬프트 임베딩 간의 semantic gap이 매번 다름
  - 또한 원하는 결과물의 semantic도 매번 달라짐..
- 작은 글씨는 잘 변환하지 못 한다
  - Scene-text를 rendering하는 backbone model인 text-diffuser 자체가 얇은 글씨, 작은 글씨를 잘 생성하지 못함.
- 무거운 text-segmentation pre-traiend 모델을 사용한다(like RPN..)
  - Hi-SAM이라는 무거운 text-segmentation 모델을 사용하며 진정한 end-to-end라 볼 수 없다. 이를 위해 미래 연구에서 새로운 알고리즘을 고안 및 데이터셋 구축

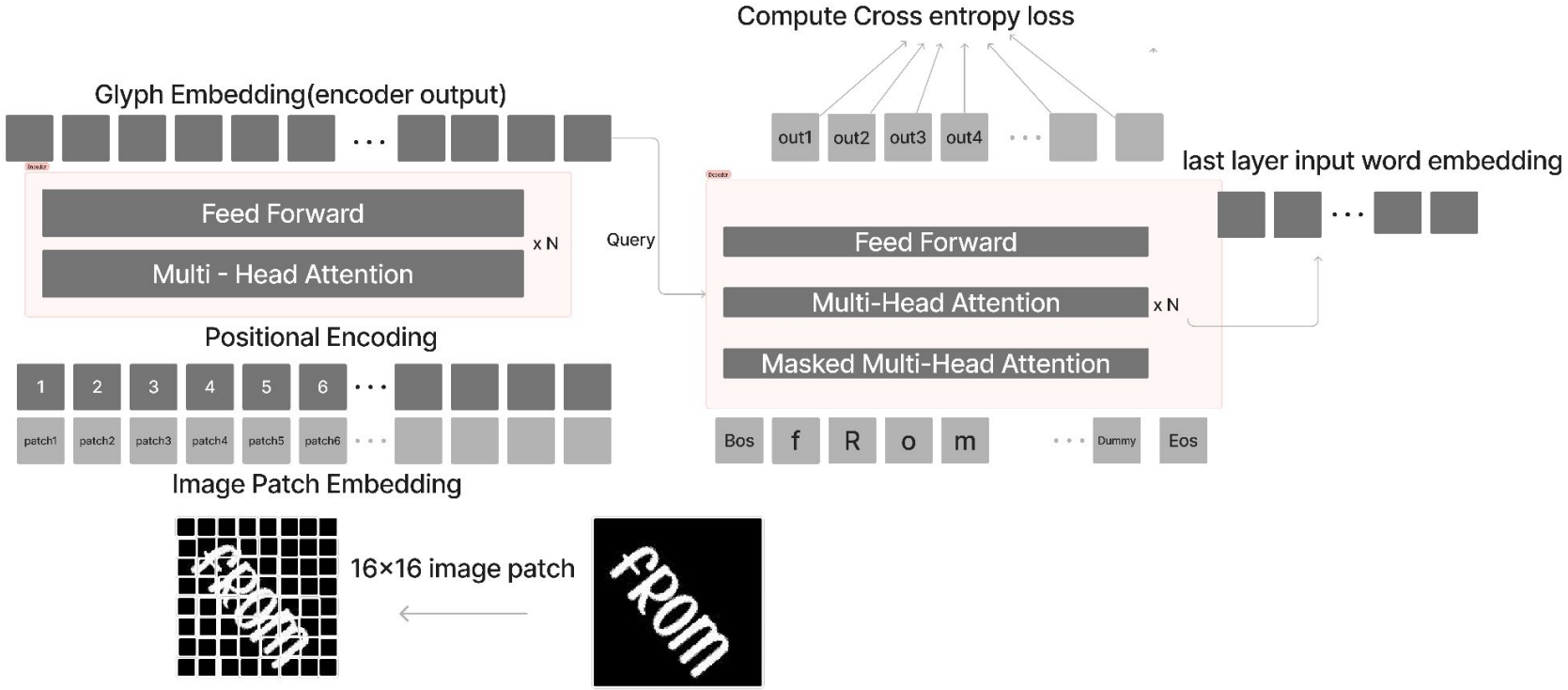
8. 미래연구: New method for end to end text rendering



8. 미래연구: Glyph-Word 1M dataset 현재 생성 완료

- (128,128,1) 사이즈, 검은 배경, 흰 글자
- 영단어 빈도순 5만개 추출
- 대, 소문자 구별
- Random font size, font style, font angle
- Consider vertical writing case
- 20 augmentation per single word(1000000 glyph-word pair, 16GB)

8. 미래연구: Glyph-Word Transformer Model



Thank you