Paper summary

AI VISION Lab

1. 공부한 논문의 제목, 게재된 학회 혹은 저널 등 논문 기본 정보를 적으세요.
   - **A. 이름: Scalable Diffusion Models with Transformers**
   - **B. 저널: ICCV**
   - **C. 도메인: Diffusion**
   - **D. 출판연도: 2022**
   - **E. 저자: William Peebles, Saining Xie**

2. 논문에서 제안한 알고리즘 및 프레임워크에 대해 본인이 이해한대로 다이어그램을 그려보세요. **논문 Figure를 그대로 따라 그리면 안됩니다.**
   - A. **[선행연구] DDPM 이후 Diffusion model은 U-Net 기반의 Neural Network를 사용했음. 그러나 U-Net의 Inductive bias는 Generative model인 Diffusion에 큰 효능이 없다고 생각함. 따라서, U-Net을 Transformer(ViT)로 대체할 수 있는 새로운 Architecture를 제안하고자 함.**

   - B. **기본적인 구조는 ViT model을 따름**

   - C. Input image을 Z(여기서는 Latent Diffusion, VAE의 Pretrained된 Encoder 사용) patch화 함. (이때, patch size는 hyperparameter. 제안한 Design space: 2, 4, 8)

   - D. 해당 모델은 Conditional한 상태를 고려해 구성됨. 즉, Condition(Class label: c, Timestep: t)를 어떻게 처리하는가를 기준으로 나눠서 고려할 수 있음.

   - E. In-context의 경우, Image-token + (embedded t, c) condition token을 결합(concatenation) 하여 동일하게 처리해주는 Block임. (decode 전, condition token의 경우 제거)

   - F. 혹은 Cross-Attention을 적용할 수 있는데, 이때는 2 Branch로 구성 Image-Token | Condition Token으로 나눠, Cross Attention을 적용하는 구조임. (연산량이 15% 증가한다는 단점이 존재함.)

G. Attention Block 말고, Layer Norm도 변화를 줄 수 있는데, 선행연구인 Improved DDPM에서 제안한 Adaptive Layer Norm (adaLN)를 사용하는 구조임. Token에 대해서, Layer Norm을 진행할 때, Scale Factor를 Condition Token에서 계산하여, Normalization을 진행함.

H. 마지막으로는, Zero-Initialization인데, 마지막 Batch에 대해서, 값을 0으로 초기화 했을 경우, 성능이 높게 나왔다는 선행연구 결과를 반영함.'

I. 최종적으로 Decoder를 사용해, Noise, Covariance를 예측함

3. 본인이 생각하는 이 논문의 장점이 무엇이라고 생각하나요? **논문 Contribution bullet을 그대로 따라 적으면 안됩니다.**
   A. U-Net대신 Transformer를 적용했다는 것뿐만 아니라, 이는 오랫동안 연구되어온 Transformer의 장점을 Diffusion model에 추가적으로 적용할 수 있는 Framework를 설립했고 결과적으로, Diffusion model의 새로운 발전 가능성을 제시했다는 점에서 큰 의미가 있다고 생각함.

4. 이 논문을 읽으면서 느낀 점, 혹은 배운 점이 있으면 적어보세요.
   A. U-Net 기반의 선행연구 대비 성능향상에 기여했다는 점에서, 정답이라 여겨지는 다양한 조건들을 Tuning해보는 시도도 의미가 있는 연구활동이 아닐까 생각됨.

5. 이 논문의 한계점이 있다면 무엇이라고 생각하나요?
   A. 기존 모델과의 성능 비교(FID, Recall 등)를 진행하여 성능이 향상됨을 보였다는 점은 긍정적이었으나, Flops 연산을 DiT – ㄴsmall, large 사이의 비교에만 치우쳐져, 기존 모델 대비 연산량 비교가 이뤄지지 않았다는 점이 아쉬움.

6. 본인의 연구에 접목시켜볼 점이 있을지 생각하고 적어보세요.
   A. Diffusion 선행연구

7. 본 Summary를 작성하는 과정에서 생성형AI를 사용했나요?
   A. 아니요

날짜: 2025-07-15

이름: 신준원

Scalable Diffusion models with transformers 07/19

Introduction

기존 Diffusion model

① U-net Architecture (DDPM ~)
② Resdct Block → pixel CNN++
③ spatial self-attention blocks
④ adaptive normalization layer (Improved DDPM ~)

논문 목표.

① U-net 의 inductive-bias ⇒ Diffusion model 영향 x.
   - model 학습을 위한 과정
   - i) Locality : Convolution 기반 ⇒ edge, corner. low-level 학습 ↑
   - ii) Hierarchical : encoder → high representative / decoder → low + high.
   - iii) Positional : skip connection
   - iv) Symmetric : 대칭구조.

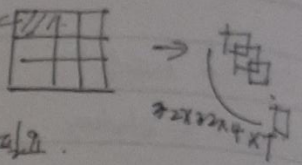
   transformer로 대체가능!

② U-Net 변경.
③ (ViT) transformer + DDPM.
   (LDM Based)
   → Latent Diffusion model (Diffusion+VAE)
④ GFlops 기반 모델 비교.
   → parameter수 증가, 그에 따른 연산량(GFlops)과 성능 향상 가능.
   → GFlops 기반 비교 필요.

Diffusion transformer (DiT) Design space

ex) 256×256.

Patchify : DiT first layer



input : 256×256×3.

z : 32×32×4.

input → patch로 자름.

T개의 token with dimension d.

⊕ positional embedding.

→ patch size (hyperparameter p)

Block design → 4개의 option

↳ Condition control 방식.

- In-context Conditioning

  ↳ T+2 token ⇒ T+2

  vector-embed t, c (t: timestep, c: class labels)

  ↳ 2개의 추가적인 token ⇒ 마지막인 사용.

  * 연산량 영향 미미, ViT 공통적 (= cls)

- Cross attention block

  image token | t, c (concat) token.

  Cross attention 진행.

  * 연산량 15% 증가.

- Adaptive layer Norm (adaLN) block ⇒ 가장 효율적인 계산 방식.

  layer (layer(1) + γ) + β.

  ↳ (t, c)의 summation 통해 계산.

- adaLN-Zero block

  Resnet → 모든 layer 중요성 초기화 필요.

  ⇒ 마지막 batch, γ (batch norm scale-factor)

  ↳ "0" 초기화

- Transformer decoder

Image token (after DiT) → decode.

↳ noise prediction.
Covariance prediction