AI 실전 10주차

Vision Transformer

Diffusion Transformer

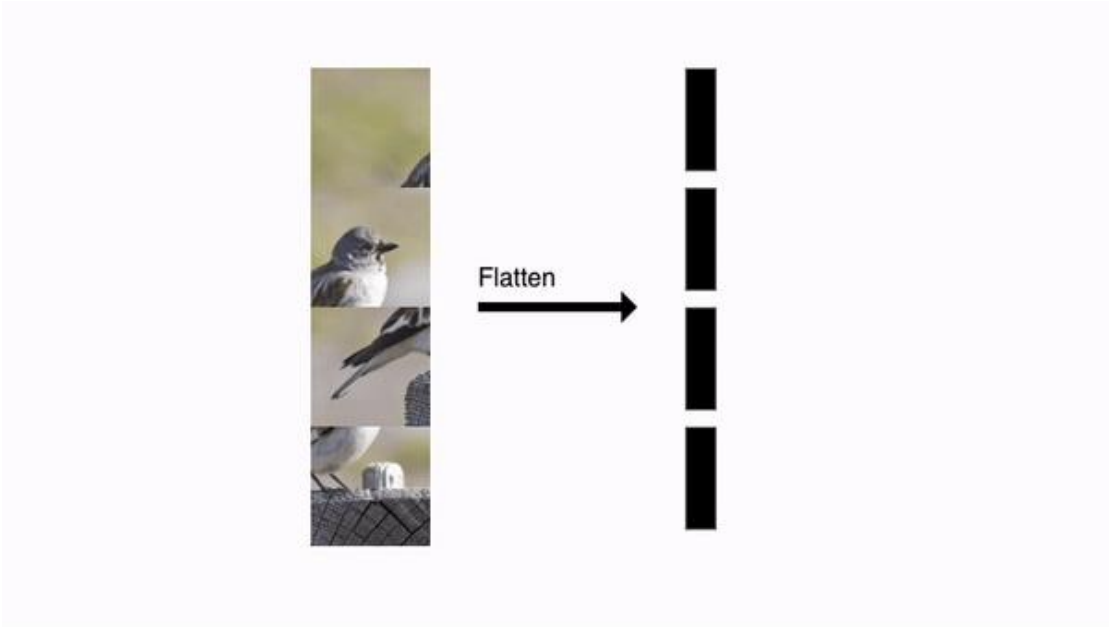
Vision Transformers – 핵심

- 이미지들을 쪼갠 것들을 기반으로 Transformer 연산 진행한 것.

Vision Transformers – patch & flatten

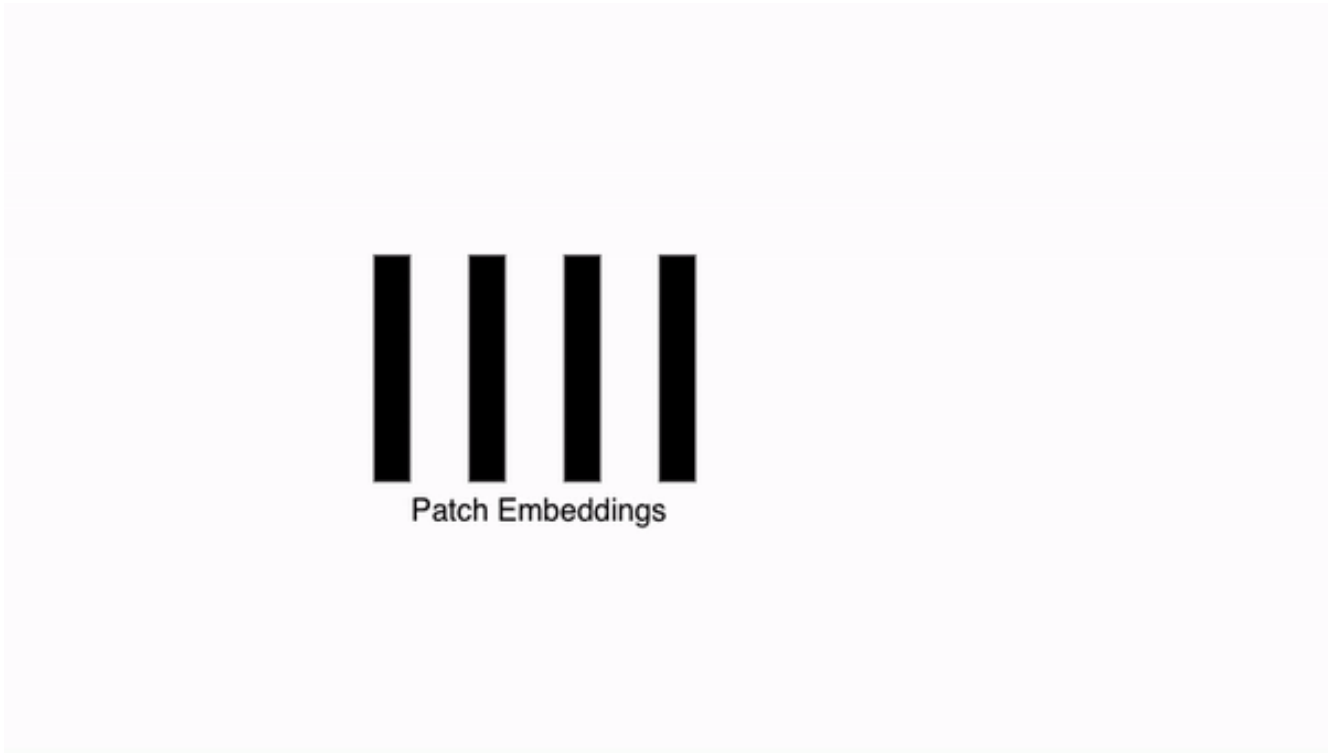
- P 값에 따라서 이미지를 나눔.
- 이미지 나누고나서 1차원으로 나열

Patch Encoding

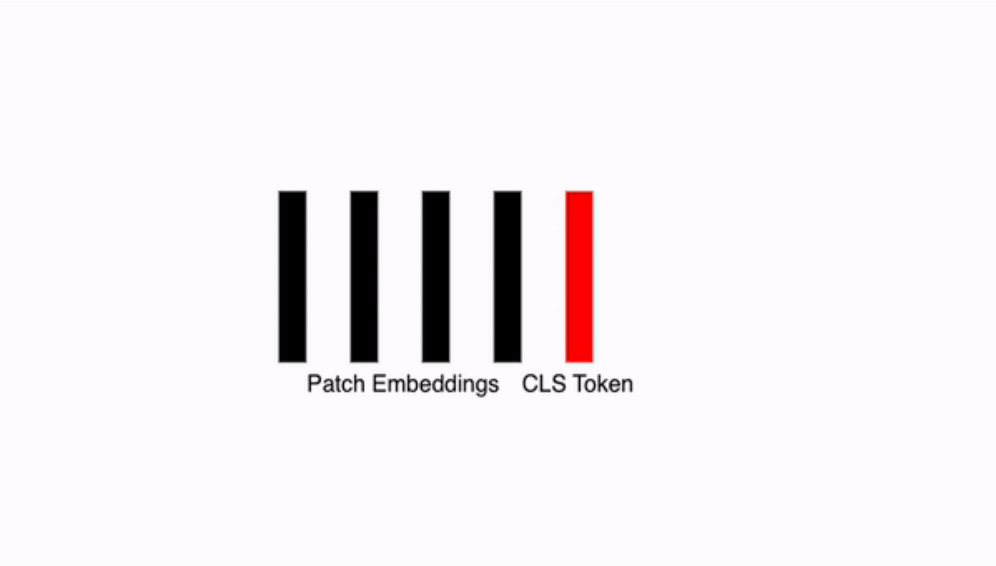


- 이미지 patch 벡터를 인코딩을 거침.

CLS 레이어 추가

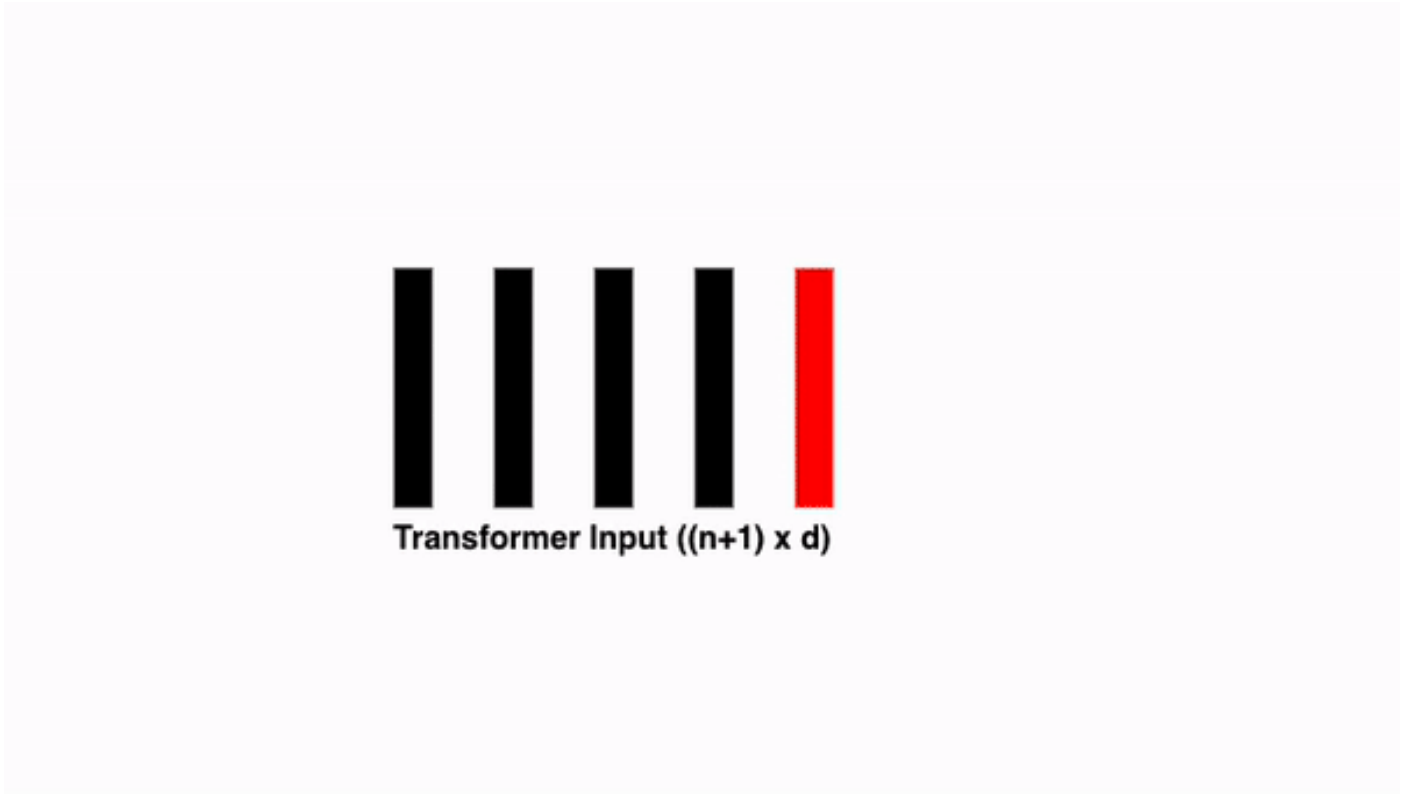


위치 임베딩 추가

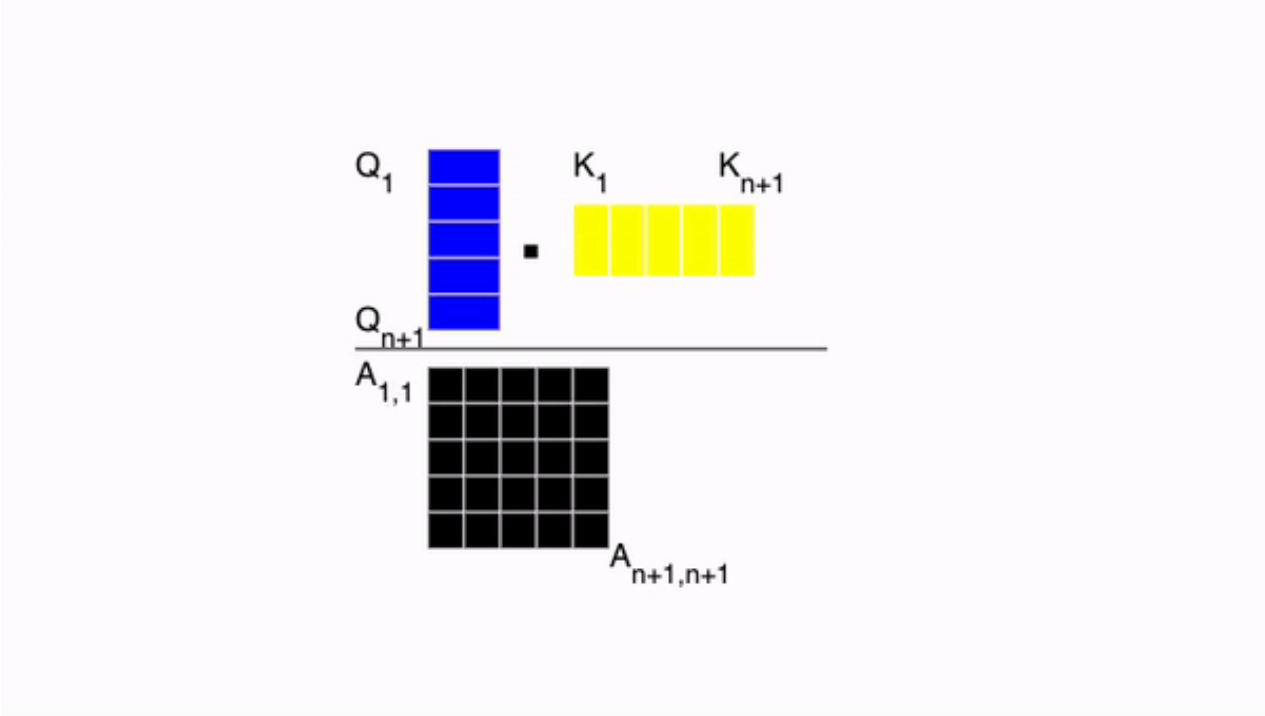


- 각 이미지에 대한 위치 정보가 손실이 됨.
- 각 이미지에 대한 위치 레이어와 결합

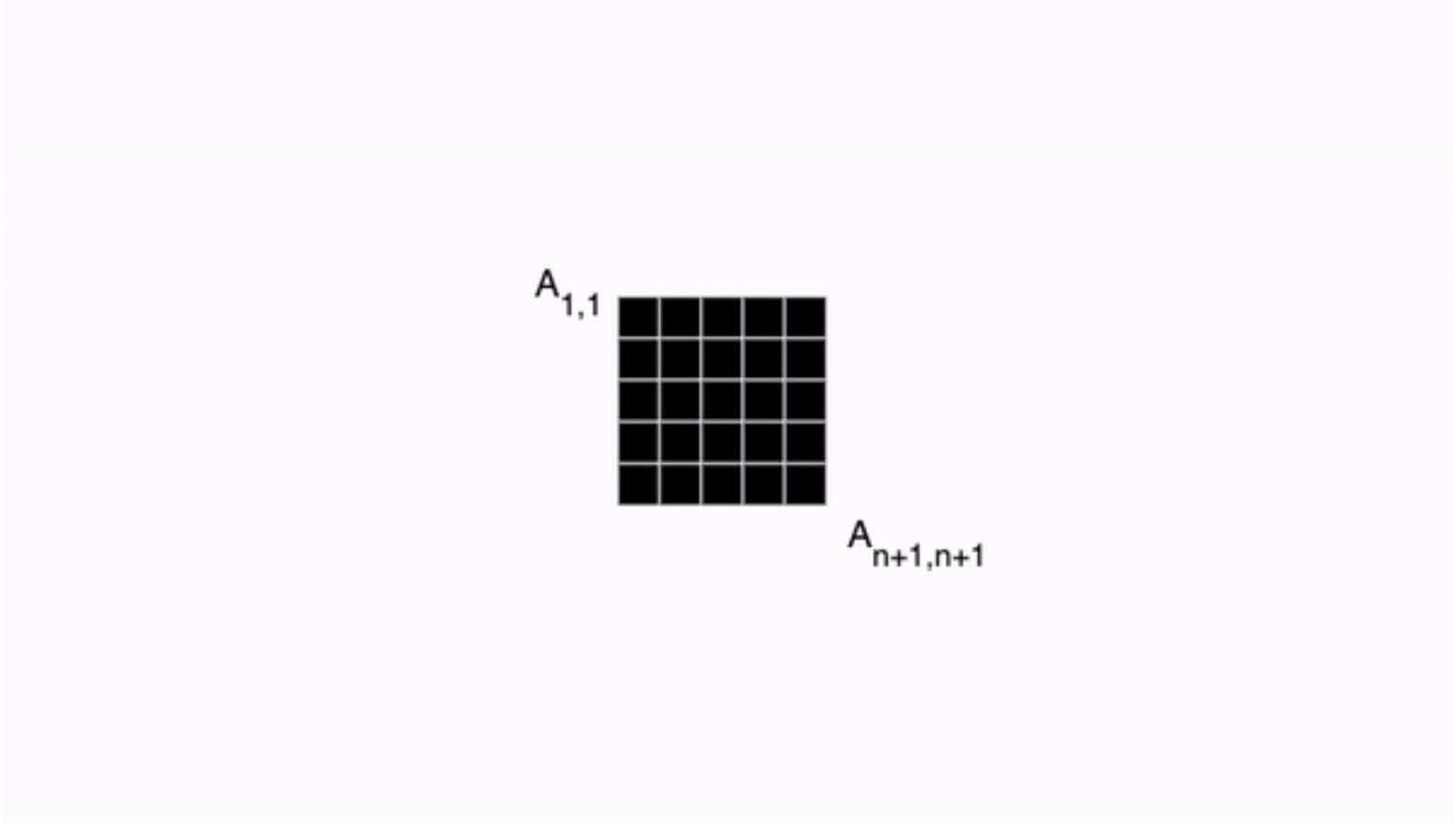
Transformer 연산



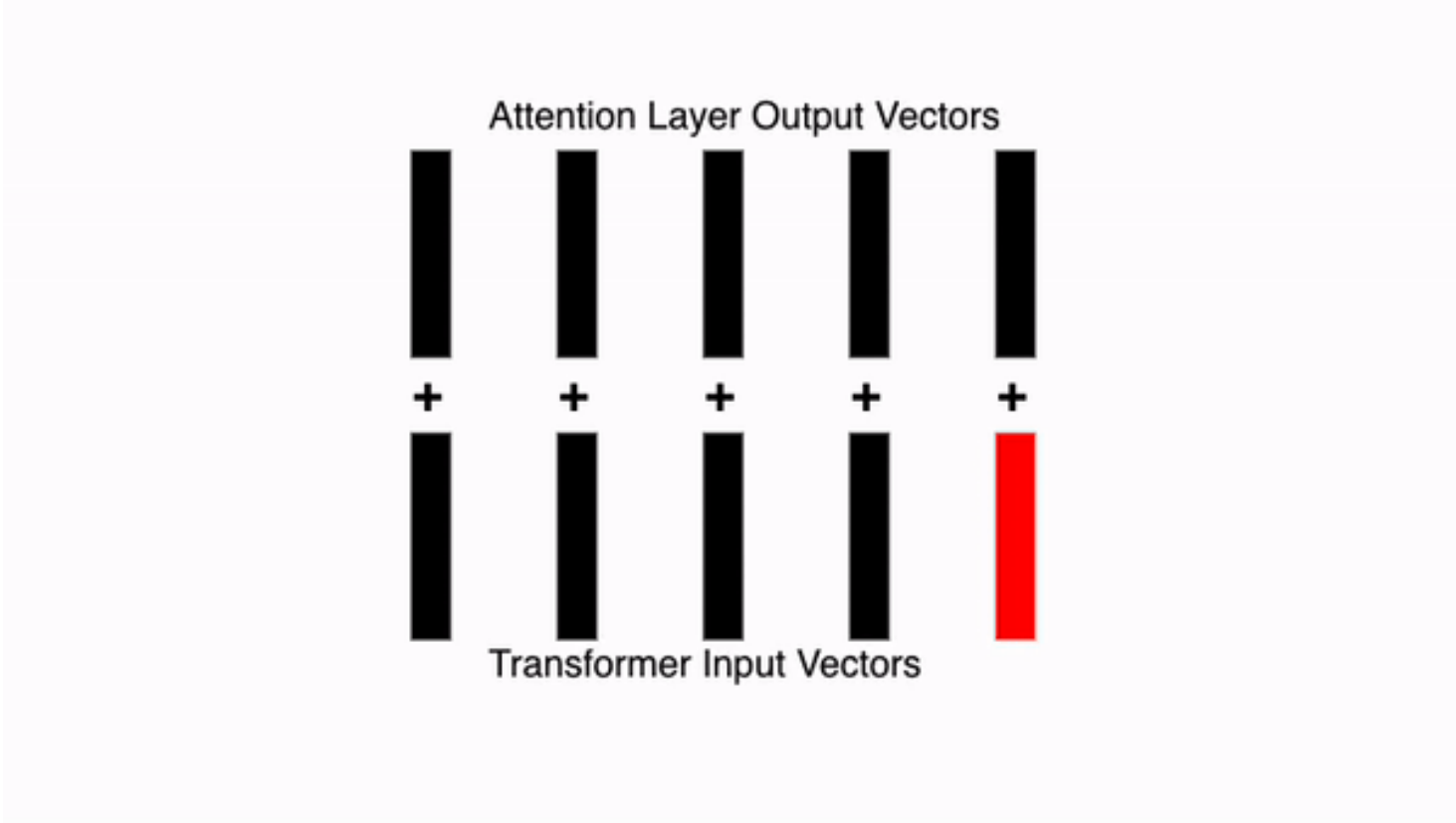
Attention 점수 구하기



집계된 컨텍스트 정보 계산

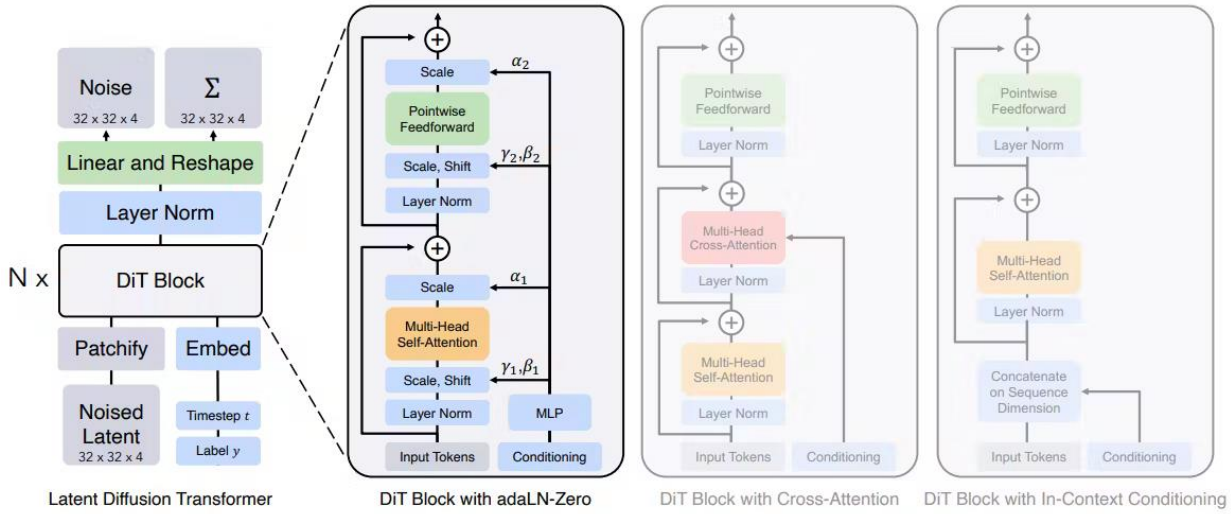


입력 레이어와 출력된 레이어 결합

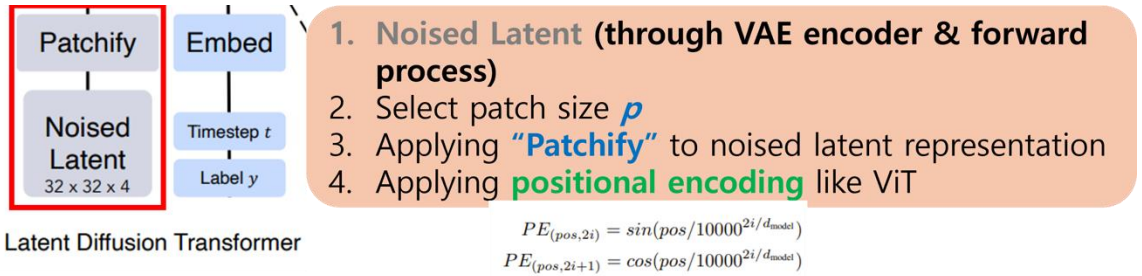


DiT 요약

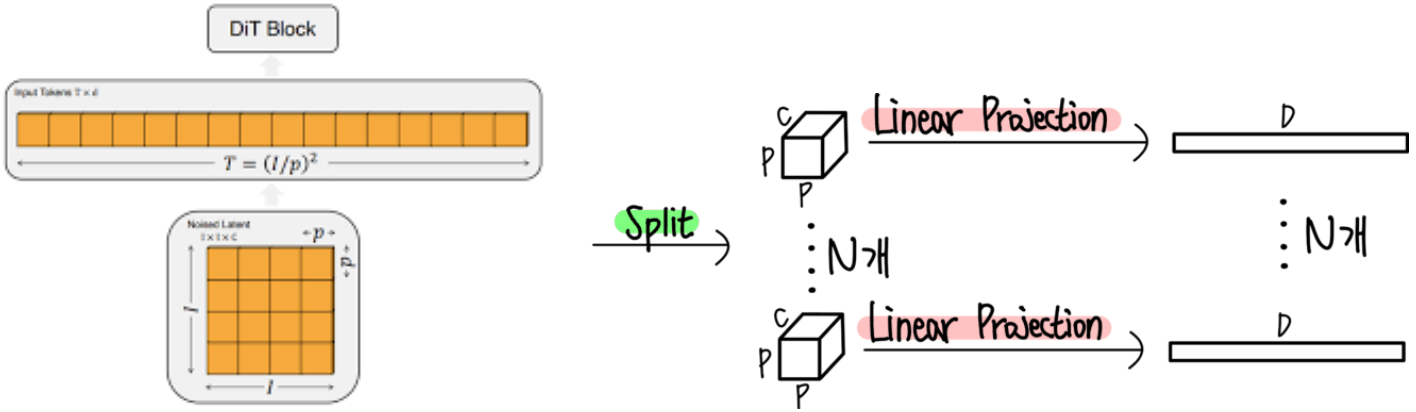
- Diffusion 처리하는 것을 Transformer 기법을 이용하여 제안
- 기존 Latent Diffusion Model 에 비해 기존보다 성능 높히고, 사용자의 요구사항에 맞추어 생성



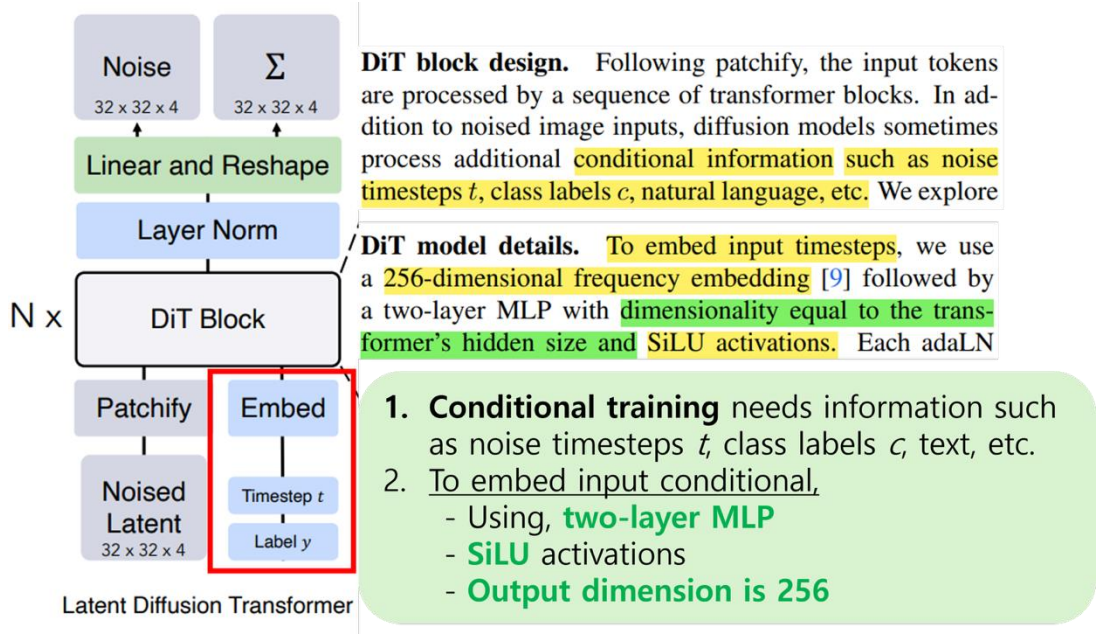
DiT 특이점 – **Noised Latent & Patchify**



- ViT에서 적용한 Patch Embedding 처리
  - 이미지들을 조각내어 이미지 나열
  - 이미지를 구분할 P Value의 값에 따라 나누어진 개수가 달라짐



DiT 특이점 - **To embed conditional information**



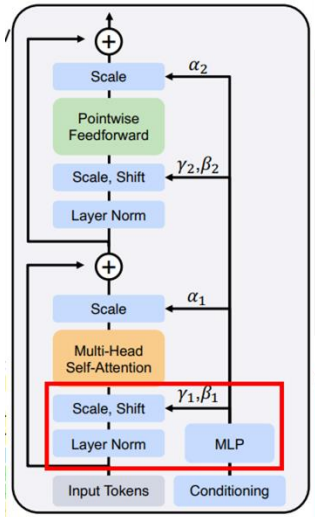
DiT block design. Following patchify, the input tokens are processed by a sequence of transformer blocks. In addition to noised image inputs, diffusion models sometimes process additional conditional information such as noise timesteps t, class labels c, natural language, etc. We explore

DiT model details. To embed input timesteps, we use a 256-dimensional frequency embedding [9] followed by a two-layer MLP with dimensionality equal to the transformer's hidden size and SiLU activations. Each adaLN

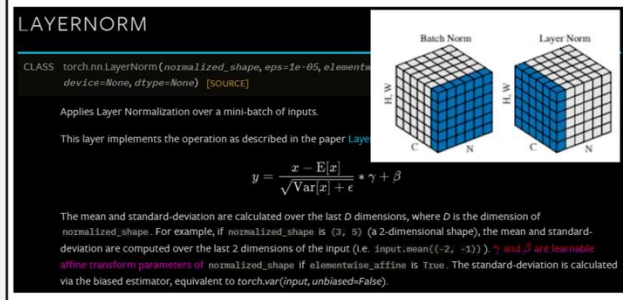
1. **Conditional training** needs information such as noise timesteps *t*, class labels *c*, text, etc.
2. To embed input conditional,
   - Using, **two-layer MLP**
   - **SiLU** activations
   - **Output dimension is 256**

- DiT Block 계산하기 전에 계산할 이미지를 미리 제공
  - CFG(classifier-free guidance)
  - CG(classifier-guidance)
  - Etc…

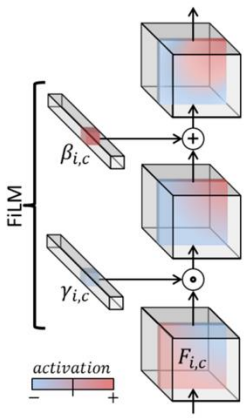
DiT 특이점 - **Adaptive Layer Norm-Zero Block**



DiT Block with adaLN-Zero

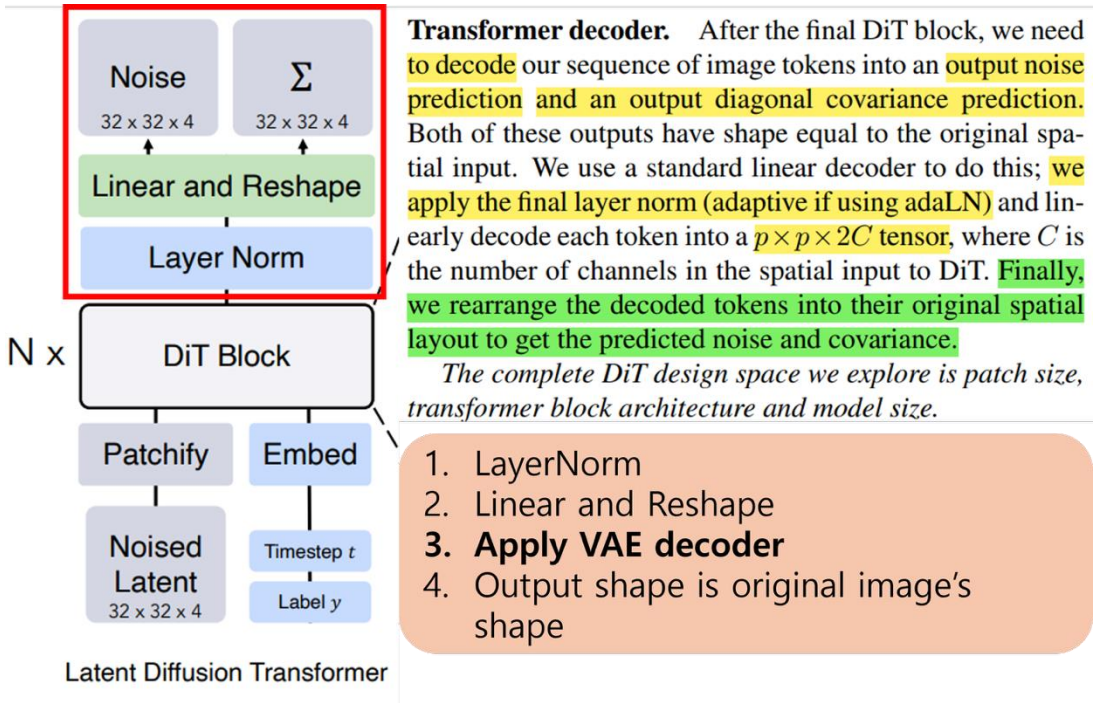


1. First, **apply Layer-Norm without learnable parameters shift and scale.**
2. Second, **extract two embeddings factor: shift and scale** (through MLP with conditional info).
3. Third, **apply two factors with LN output.**



- 서로 다른 정보들을 결합하기 위한 방법을 적용한 구조.
- FiLM 의 구조에서 Layer normalization 추가된 개념.
- 직접적으로 learnable하는 것이 아닌, timestep과 label의 embedding을 shift와 scale값으로 활용

DiT 특이점 - **Transformer Decoder**



Transformer decoder. After the final DiT block, we need to decode our sequence of image tokens into an output noise prediction and an output diagonal covariance prediction. Both of these outputs have shape equal to the original spatial input. We use a standard linear decoder to do this; we apply the final layer norm (adaptive if using adaLN) and linearly decode each token into a $p \times p \times 2C$ tensor, where C is the number of channels in the spatial input to DiT. Finally, we rearrange the decoded tokens into their original spatial layout to get the predicted noise and covariance.

The complete DiT design space we explore is patch size, transformer block architecture and model size.

1. LayerNorm
2. Linear and Reshape
3. **Apply VAE decoder**
4. Output shape is original image's shape

- DiT Block 연산 이후 VAE 디코더 수행함.
- **VAE decoder에 noise 값을 넣어서 실제 이미지를 생성**