AutoEncoder

A Universal Tool for Data Compression, Denoising, and Anomaly detection

DMQA Open Seminar

한경석

hanks6125@korea.ac.kr

2025.6.27

Index

Introduction

○ **한경석 (Han Kyungseok)**

- 고려대학교 산업경영공학과 석사 과정 (2024.03 ~ present)
- Data Mining & Quality Analytics Lab **(김성범 교수님)**

○ Research Interest

- Regression
- Deep Learning

○ E-Mail

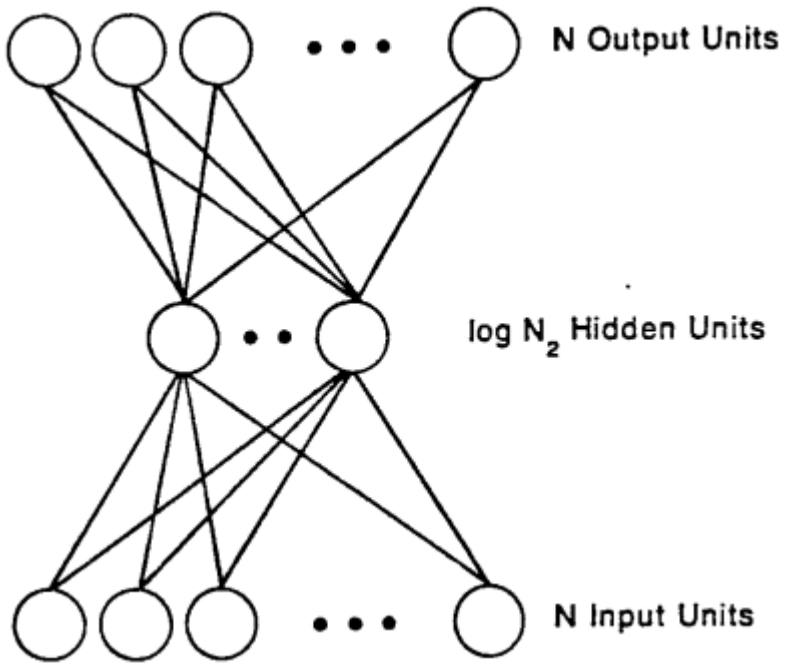
- hanks6125@korea.ac.kr

Background

AutoEnocder

○ Learning Internal Representation by error backpropagation(1986)* 논문에서 처음 Autoencoder 개념을 소개

- Backpropagation 개념과 함께 XOR, Encoding Problem 등에 대해 다룸



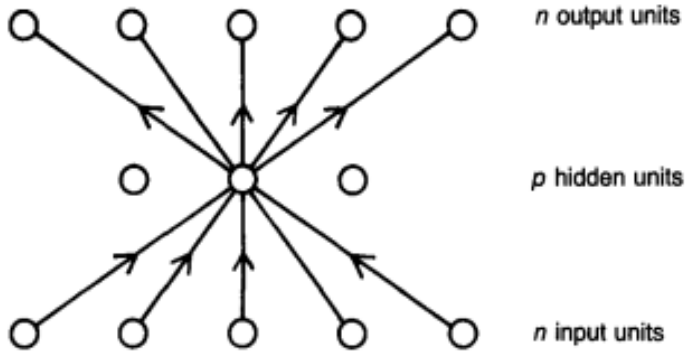
*D.E.Rumelhart, G.E.Hinton and R.J.Williams

Background

AutoEnocder

○ Neural Networks and Principal Component Analysis:Learning from examples without local minima(1989) 논문에서 auto-encoding이라는 용어와 함께 그 개념을 명확히 제시

to realize image compression. Auto-association, which is also called auto-encoding or identity mapping (see Ackley, Hinton, & Sejnowski; 1985; Ellman & Zipser, 1988) is a simple trick intended to avoid the need for having a teacher, that is, for knowing the target values y_t, by setting $x_t = y_t$. In this



*Pierre Baldi and Kurt Hornik, University of California

Background

AutoEnocder (AE)

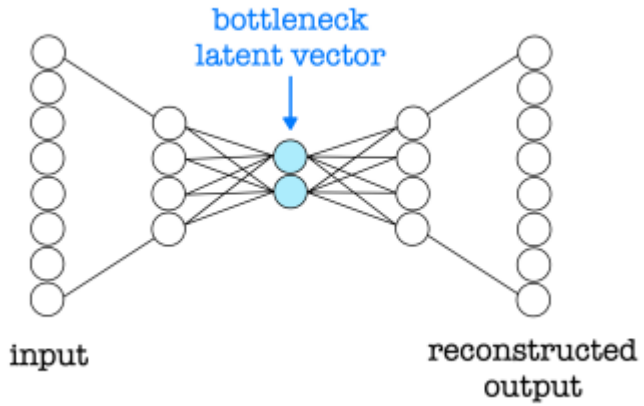
○ Encoder-Decoder를 통한 Data Reconstruction을 수행하는 비지도 학습

- Input $x^k = (x_1^k, x_2^k, \dots, x_d^k)$와 **동일한** output $\hat{y}^k = (\hat{y}_1^k, \hat{y}_2^k, \dots, \hat{y}_d^k)$를 얻고자 함
- Loss : Reconstruction error

. MSE for real values : $L(\theta) = \frac{1}{2}\Sigma_k \left|\left|\hat{y}^k - x^k\right|\right|_2^2 = \frac{1}{2}\Sigma_k\Sigma_i\left(\hat{y}_i - x_i^k\right)^2$

. Cross entropy for binary $\{0,1\}$: $L(\theta) = -\Sigma_k\{x^k log\hat{y}^k + \left(1 - x^k\right)\log\left(1 - \hat{y}^k\right)\}$

. 학습을 위해 backpropagation 알고리즘을 이용

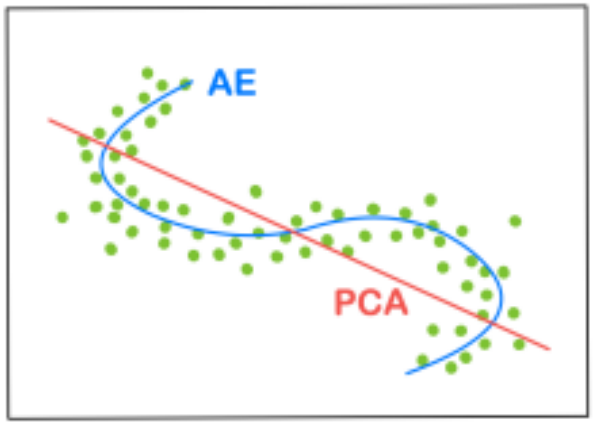


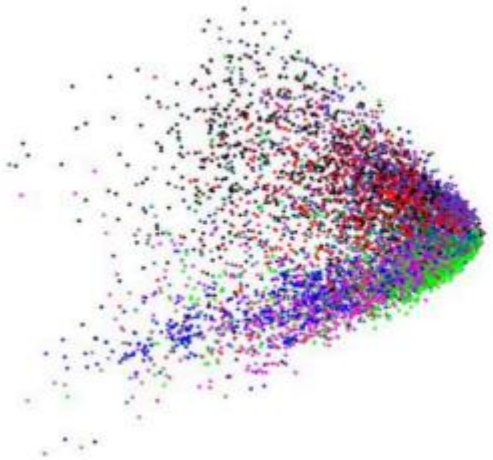
Background

AutoEnocder (AE)

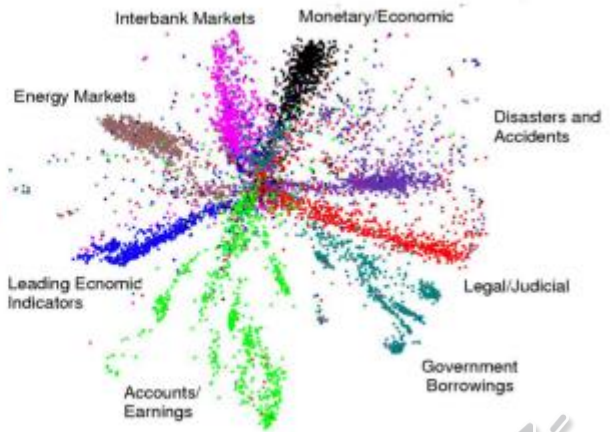
○ 차원 축소

- 복잡한 입력 데이터를 압축된 Latent vector로 만들고, 이 Latent vector는 분류, 이상탐지 등의 다른 알고리즘의 Input으로도 활용될 수 있음
- 대표적인 차원 축소 방법인 PCA가 Linear hyperplane으로 차원 축소를 시도하는 반면, AE는 nonlinear manifold로 차원 축소를 시도함





PCA



AE

Background

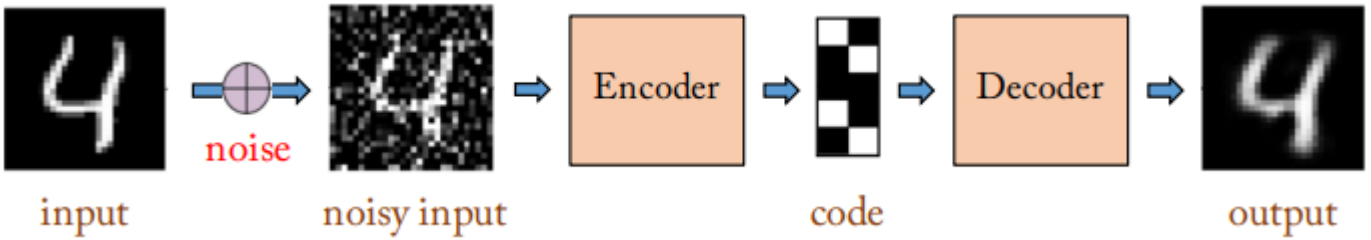
Denoising Auto-Encoder

○ Autoencoder의 한계

- 네트워크가 단순히 입력 패턴을 복사하도록 학습되어, 일반화 능력이 부족하고 노이즈에 취약함
- 의미 없는 latent vector가 학습됨

○ Noise추가 : Noise에 강인한 latent vector 학습, 일반화 성능 향상

- 입력에 노이즈를 주고, 깨끗한 원본을 복원하도록 학습
  . 입력 x → Noise 추가 → $\tilde{x}$
  . 인코더: $z = f_\theta(\tilde{x})$
  . 디코더: $\hat{x} = g_\phi(z)$
  . Loss: $\mathcal{L} = \left||x - \hat{x}\right||^2$, Noise가 추가된 $\tilde{x}$가 아닌 원본 이미지 x와의 reconstruction error를 사용



Background

Variational Auto-Encoder

○ 기존 Autoencoder의 한계

- AE는 입력 x 를 압축했다가 단순 복원 하는 것을 목적으로 함
- 따라서 latent space z 는 비연속적이고, 임의의 z에서 샘플링을 했을 때 의미 없는 출력을 생성함
- 즉, AE는 입력 재구성은 잘하지만, 새로운 데이터를 생성할 수는 없음

○ VAE의 핵심 아이디어

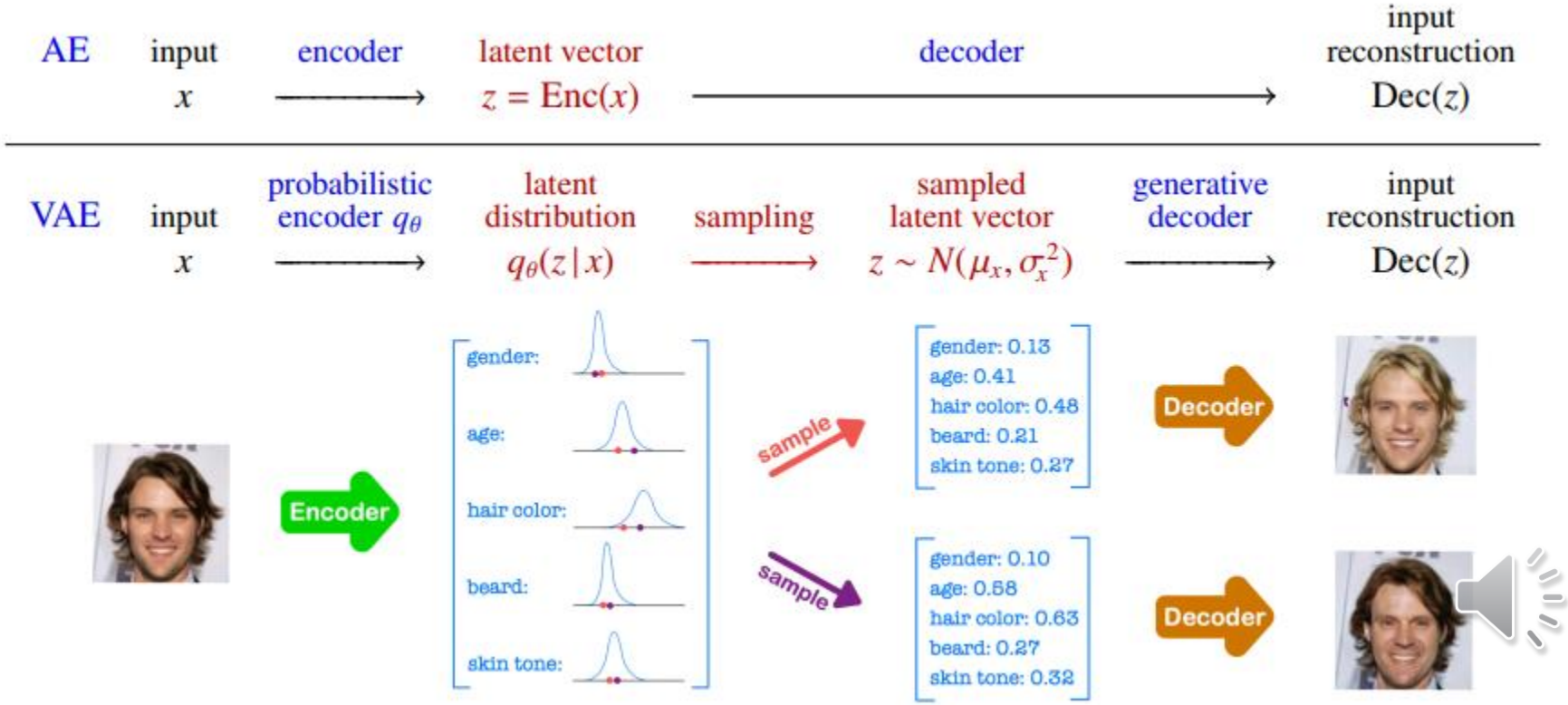
- Latent space를 확률분포로 모델링하여 정규분포와 같은 구조화된 공간으로 만듦
- 구조화된 latent space로부터 sampling 하여 의미 있는 데이터를 생성
  . $z \sim q_\theta(z|x)$: 입력으로부터 분포를 추정 (μ, σ)
  . $z \sim \mathcal{N}(0, I)$: 표준정규분포로 강제함
  . $x \sim p_\phi(x|z)$: 생성
  . Loss : $\mathcal{L} = \frac{1}{2}\Sigma_i(\hat{y}_i - x_i)^2 + \beta D_{KL}(q_\theta(z|x)||\mathcal{N}(0, I))$, tradeoff between MSE and KL divergence

Background

Variational Auto-Encoder

○ AE와 VAE의 비교

- AE는 latent space의 각 attributes를 single point 로 압축
- VAE는 single point가 아닌 정규 분포 $q_\theta(z|x)$ 로 압축



Background

Autoencoder의 활용

○ **Representation Learning**

- 지도 학습 없이 유용한 feature를 추출 → classification, forecasting에 활용

○ **Anomaly detection**

- 정상 데이터를 학습한 후, 복원 오차가 큰 샘플을 이상치로 잡아냄

○ **Noise reduction**

- 영상, 음성 에서 노이즈 제거 필터로 사용

○ **Dimensionality reduction**

- 비선형 차원 축소를 활용해 시각화, 전처리, Feature compression 용도로 사용

○ **생성 모델**

- DAE, VAE, GAN, Diffusion model 등 AE 구조에서 발전하거나 AE를 활용해 더욱 성능을 높임(Latent Diffusion Model)

AutoEncoder in Diffusion Models

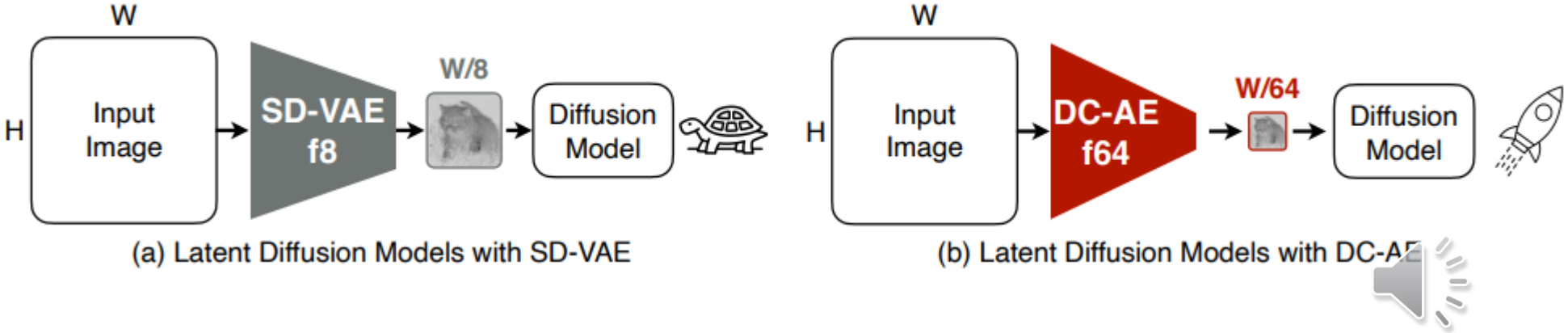
Deep Compression Autoencoder for Efficient High-Resolution Diffusion Models

○ 배경

- Latent Diffusion Model은 입력 이미지를 AE로 압축 후 latent 공간에서 diffusion 적용
- 기존 AE는 Spatial compression ratio f8 (H/8 x W/8) 을 사용
  . f8 : 원래의 이미지 H x W 를 H/8 x W/8 사이즈로 압축
- 고해상도 이미지 (512x512, 1024x1024)에서는 **연산량이 커서 f64, f128 수준의 더 강한 압축이 필요함**
- **하지만 기존 AE는 f가 커질수록 복원 정확도가 급락**

○ 핵심 주제

- 압축 비율을 64, 128까지 높이더라도 복원 품질을 유지하는 **Deep Compression Autoencoder (DC-AE)** 구조를 제안



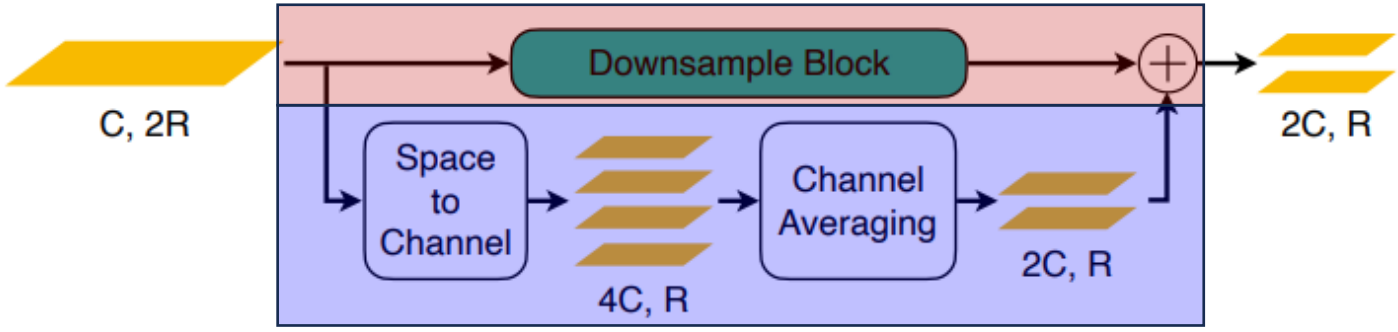
(a) Latent Diffusion Models with SD-VAE

(b) Latent Diffusion Models with DC-AE

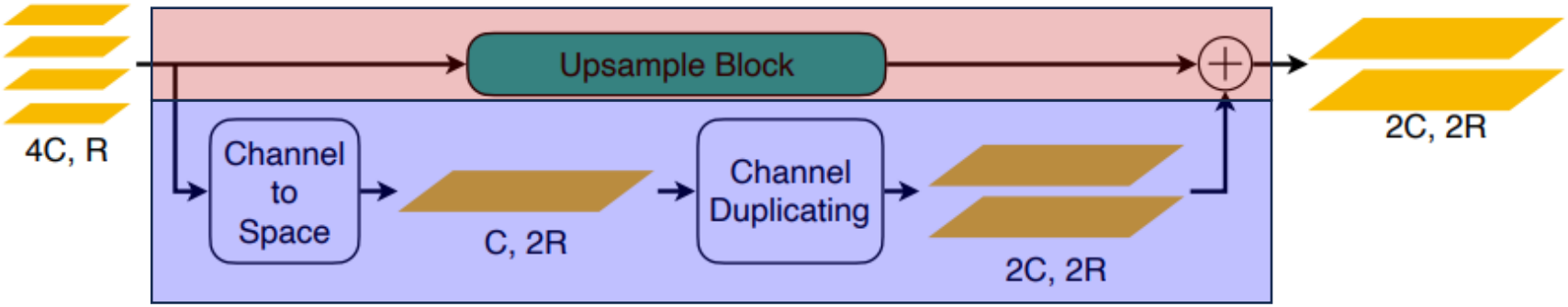
AutoEncoder in Diffusion Models

Deep Compression Autoencoder for Efficient High-Resolution Diffusion Models

○ 핵심 아이디어1 : Residual Autoencoding



(a) DC-AE Downsample Block with Residual Autoencoding



(b) DC-AE Upsample Block with Residual Autoencoding

Figure 4: **Illustration of Residual Autoencoding.** It adds non-parametric shortcuts to let the neural network modules learn residuals based on the space-to-channel operation. 'C' denotes the number of channels. 'R' denotes the image size.

AutoEncoder in Diffusion Models

Deep Compression Autoencoder for Efficient High-Resolution Diffusion Models

○ 핵심 아이디어1 : Residual Autoencoding

- Space-to-channel + Channel Averaging

. 입력 이미지를 downsampling 하면서도 구조 정보를 최대한 보존하도록 평균 기반의 shortcut 경로를 만듦

$$H \times W \times C \rightarrow \frac{H}{p} \times \frac{W}{p} \times p^2 C$$

H : height, W : width, C : # of channels

p: $\sqrt{compression\ ratio}$

$$H \times W \times C \xrightarrow{\text{space-to-channel}} \frac{H}{2} \times \frac{W}{2} \times 4C$$

$$\underbrace{\xrightarrow{\text{split into two groups}} [\frac{H}{2} \times \frac{W}{2} \times 2C, \frac{H}{2} \times \frac{W}{2} \times 2C] \xrightarrow{\text{average}} \frac{H}{2} \times \frac{W}{2} \times 2C.}_{\text{channel averaging}}$$

AutoEncoder in Diffusion Models

Deep Compression Autoencoder for Efficient High-Resolution Diffusion Models

○ 핵심 아이디어1 : Residual Autoencoding

- Channel-to-space + Channel Duplicating

. 입력 이미지를 downsampling 하면서도 구조 정보를 최대한 보존하도록 평균 기반의 shortcut 경로를 만듦

$$\frac{H}{p} \times \frac{W}{p} \times p^2 C \rightarrow H \times W \times C$$

H : height, W : width, C : # of channels

p: $\sqrt{compression\ ratio}$

$$\frac{H}{2} \times \frac{W}{2} \times 2C \xrightarrow{\text{channel-to-space}} H \times W \times \frac{C}{2}$$

$$\underbrace{\xrightarrow{\text{duplicate}} [H \times W \times \frac{C}{2}, H \times W \times \frac{C}{2}] \xrightarrow{\text{concat}} H \times W \times C.}_{\text{channel duplicating}}$$

AutoEncoder in Diffusion Models

Deep Compression Autoencoder for Efficient High-Resolution Diffusion Models

○ 핵심 아이디어2 : Decoupled High-Resolution Adaptation

- 고해상도(512x512, 1024x1024)에서 학습이 불안정하거나 일반화 성능이 급락하는 것을 해결

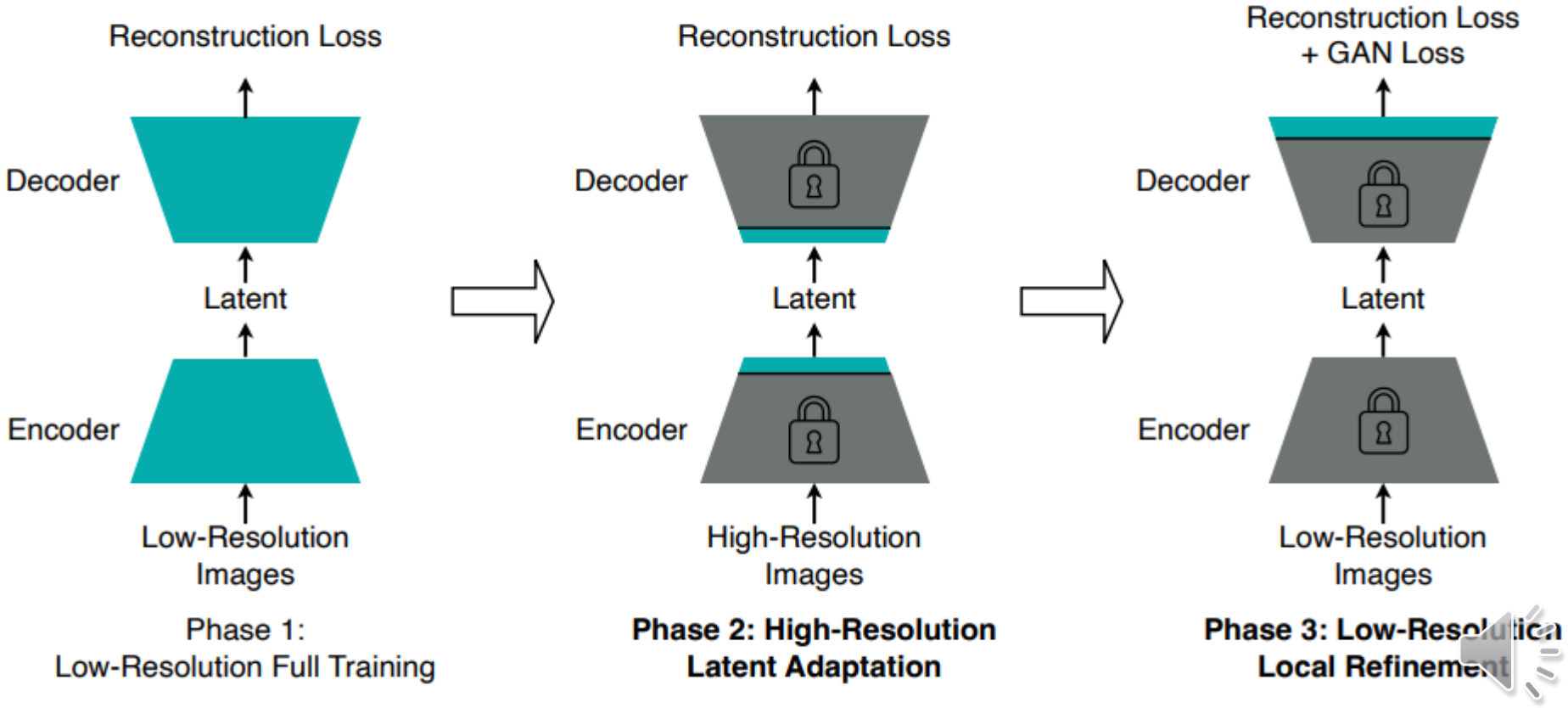


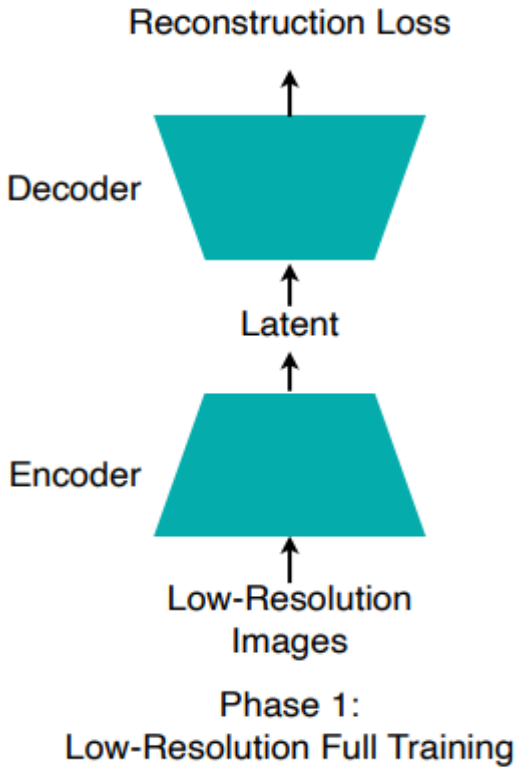
Figure 6: **Illustration of Decoupled High-Resolution Adaptation.**

AutoEncoder in Diffusion Models

Deep Compression Autoencoder for Efficient High-Resolution Diffusion Models

○ 핵심 아이디어2 : Decoupled High-Resolution Adaptation

- **Phase1** : 고해상도 이미지를 downsampling 하여 low-resolution 상태에서 전체 autoencoder를 학습함
  . **전체 AE 구조를 안정적으로 학습 (고해상도부터 하면 학습이 불안정하고 연산량 폭증, 애초에 논문의 배경)**
  . **고해상도의 복잡한 Detail은 차치하고 semantic content + 구조를 재구성**하는 역할



① 원본 고해상도를 donwsampling

$$x \in \mathbb{R}^{3\times512\times512} \rightarrow x_{low} = Resize(x, 256 \times 256)$$

② Full AE Training, 이때 AE는 VAE가 아니라 original VAE를 사용함

> we use the original autoencoders instead of the variational autoencoders for our models, as they perform the same in our experiments and the original autoencoders are simpler

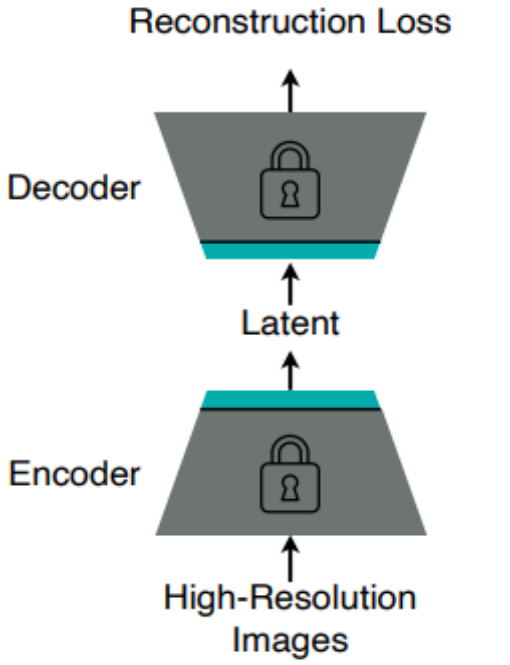
③ 이 과정에는 Residual shortcut 구조가 포함됨

AutoEncoder in Diffusion Models

Deep Compression Autoencoder for Efficient High-Resolution Diffusion Models

○ 핵심 아이디어2 : Decoupled High-Resolution Adaptation

- **Phase2** : 고해상도 이미지를 입력으로 사용하지만, **전체 AE학습은 안하고, latent space 일부만 fine-tuning**
  . Phase1에서 학습한 latent space는 저해상도에서는 잘 작동하지만 고해상도에서는 해상도 차이로 표현이 왜곡됨
  . 전체 AE를 학습하진 않고 중간 layer 일부만 조정해서 **고해상도에서 잘 작동하도록 fine-tuning**



Phase 2: High-Resolution Latent Adaptation

① 원본 고해상도 이미지를 입력으로 사용

$$x_{high} \in \mathbb{R}^{3\times512\times512 \; or \; 1024\times1024}$$

② Phase1에서 학습한 전체 AE 구조를 그대로 사용

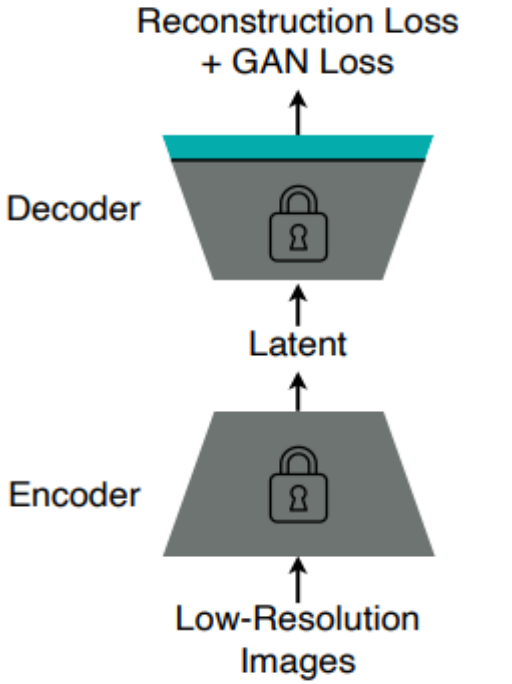
③ Projection block만 학습, 나머지는 Freezing

AutoEncoder in Diffusion Models

Deep Compression Autoencoder for Efficient High-Resolution Diffusion Models

○ 핵심 아이디어2 : Decoupled High-Resolution Adaptation

- **Phase3** : GAN Loss를 활용하여 **decoder의 마지막 부분만 미세조정, local detail 품질을 향상**
  . Phase1,2를 통해 구조, 의미, 해상도 적응까지 완료
  . 복원된 이미지가 흐릿하거나, 디테일이 부족하므로 이를 고치기 위한 refinement 수행



Phase 3: Low-Resolution Local Refinement

① 다시 low-resolution 이미지 사용 (학습 안정성을 위함)

$$x_{low} \in \mathbb{R}^{3\times256\times256}$$

② AE구조는 고정, decoder의 마지막 head layer만 학습

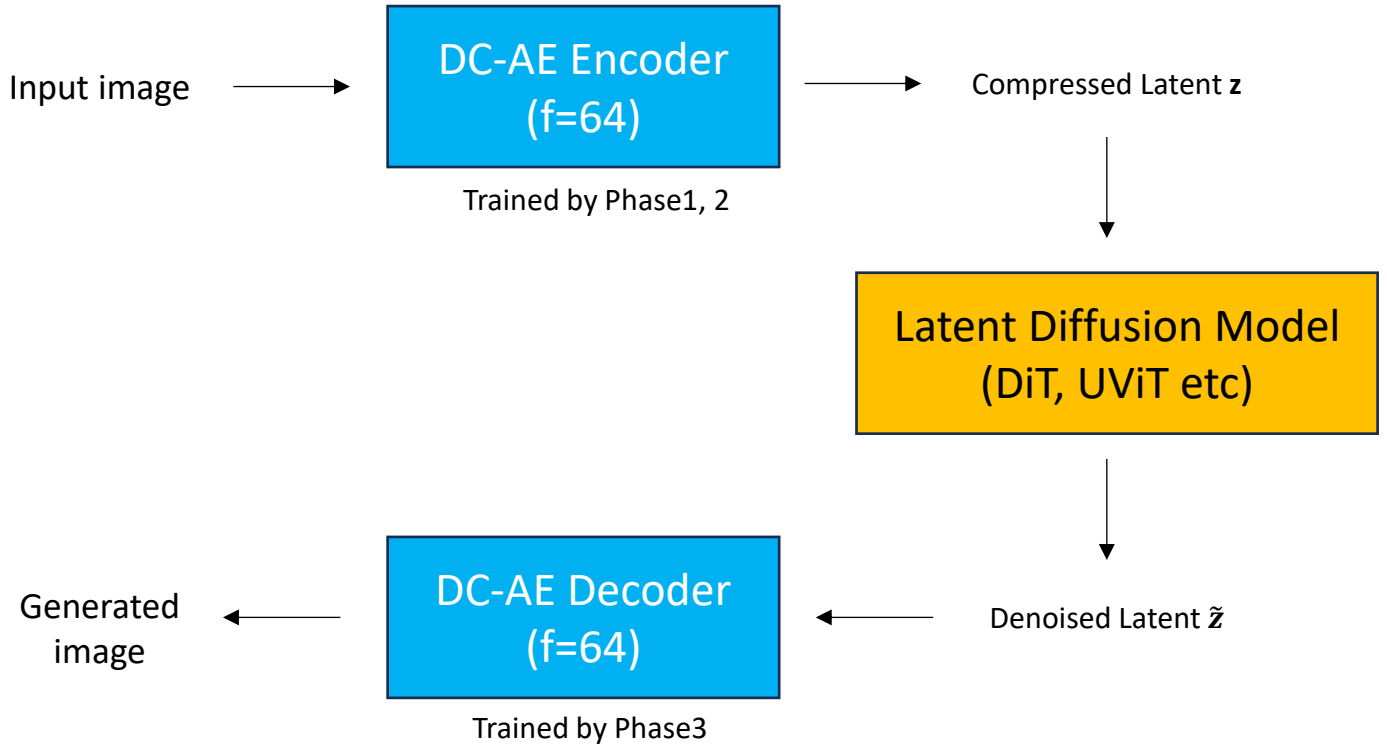
③ reconstruction error와 함께 추가적으로 GAN Loss를 적용
. GAN Loss는 local detail과 잡티를 제거하기 위함
. 전체적인 구조와 semantic은 reconstruction error만으로도 충분함

of this design is based on the finding that the reconstruction loss alone is sufficient for learning to reconstruct the content and semantics. Meanwhile, the GAN loss mainly improves local details and removes local artifacts (Figure 5). Achieving the same goal of local refinement only tuning the

AutoEncoder in Diffusion Models

Deep Compression Autoencoder for Efficient High-Resolution Diffusion Models

○ LDM과 결합



AutoEncoder in Diffusion Models

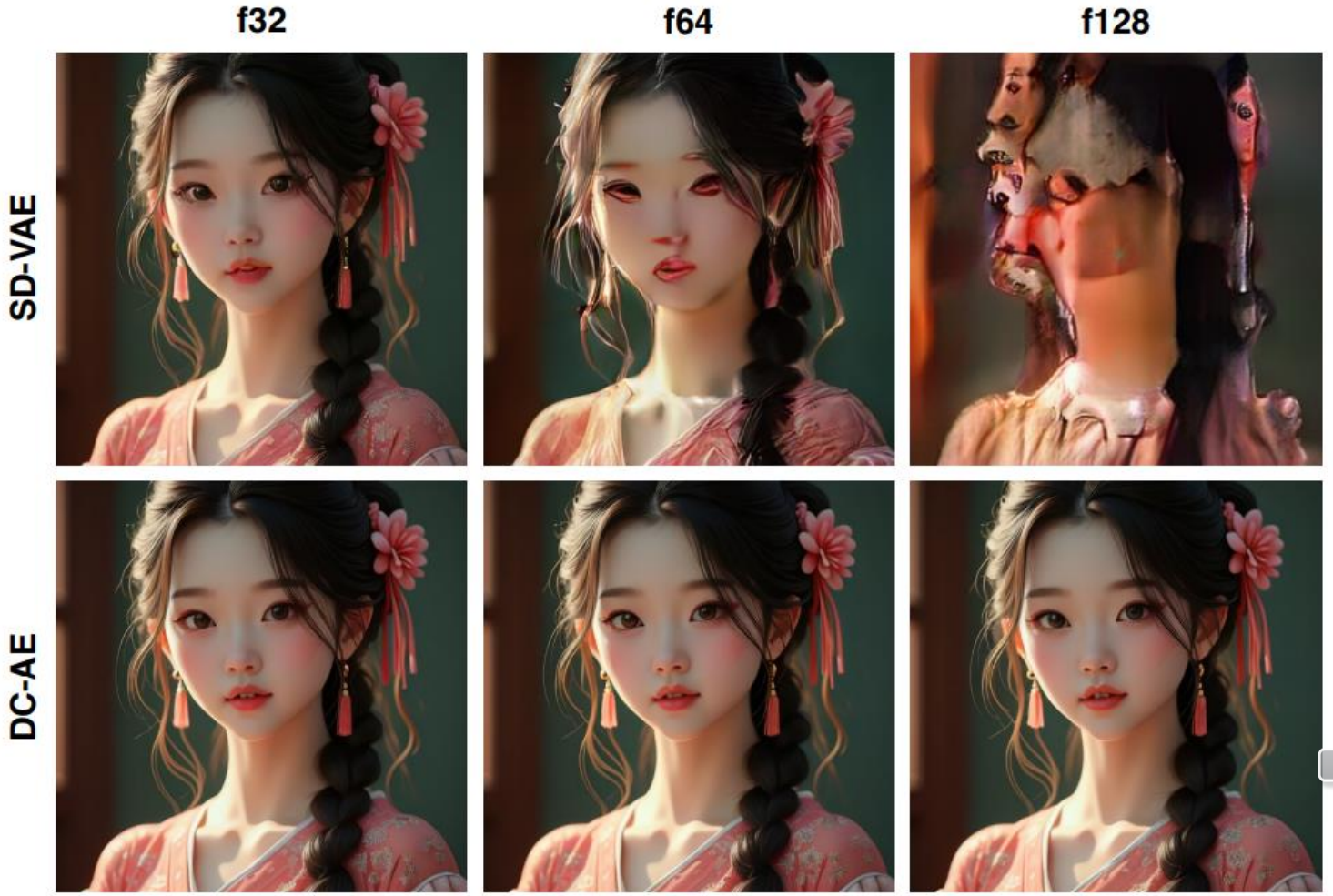
Deep Compression Autoencoder for Efficient High-Resolution Diffusion Models

○ 결과

AutoEncoder in Diffusion Models

Deep Compression Autoencoder for Efficient High-Resolution Diffusion Models

○ 결과



AutoEncoder for Anomaly Detection

A multilayer deep autoencoder approach for cross layer IoT attack detection

○ 배경

- IoT(Internet of Things) 시스템은 센서, 엣지, 네트워크, 서버 등 다양한 구성요소와 프로토콜 계층으로 이루어짐
- 공격자는 이러한 다양한 계층을 조합해 Cross-Layer 공격을 수행할 수 있으며, 이는 **기존의 단일 계층 기반 탐지 시스템으로는 파악하기가 어려움**
- Cross-Layer attack은 Network, Transport, Application 등 다양한 계층을 타깃으로 삼는 **동시다발적 attack을 의미**
- Cross Layer attack 예시 : network layer에서 MITM, transport layer에서 DDoS 공격을 함

① **Man-in-the-Middle(MITM)** attacks at the network layer
. 공격자가 통신 경로의 중간에 몰래 개입하여 데이터를 가로채거나 조작
. 피해자는 자신이 안전하게 통신하고 있다고 생각하지만 실제로는 공격자와 통신중임

② **Distributed denial of Service(DDoS)** attacks at the transport layer
. 여러 대의 컴퓨터를 이용하여 대상 시스템을 과부하 상태로 만들고 서비스 거부 상태로 만듦
. 트래픽, 연결 요청, 패킷 등을 대량으로 보내 자원을 소모 시킴

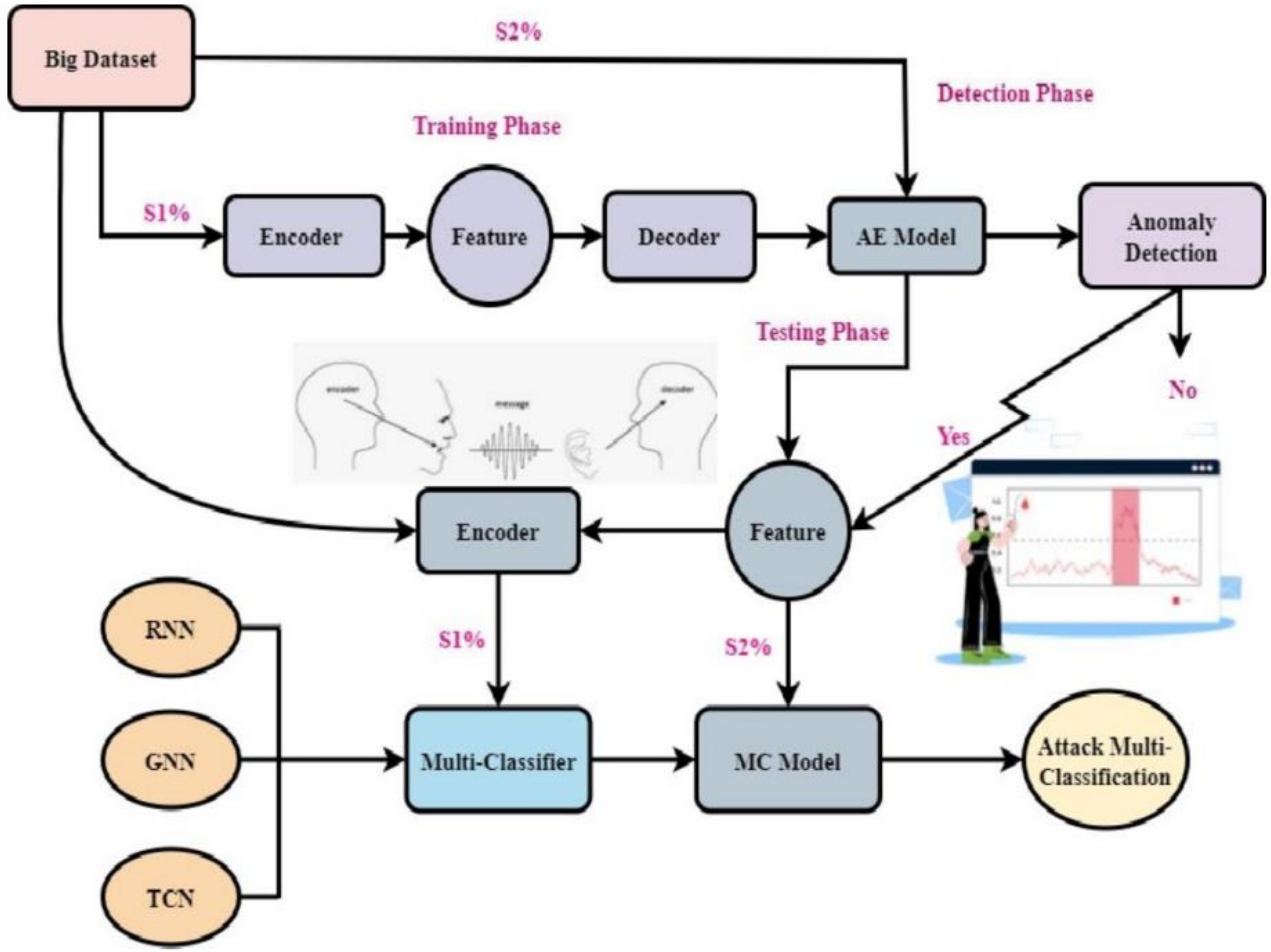
○ 연구 목적

- **Cross-Layer IoT 공격 탐지를 위한 새로운 딥러닝 기반 모델을 제안**
- **M-LADE(Multi-Layer Deep Autoencoder)** 모델을 통해 네트워크, 전송, 애플리케이션 계층을 통합적으로 분석하고, 각 계층에서 발생하는 다양한 보안 위협을 효과적으로 탐지

AutoEncoder for Anomaly Detection

■ **A multilayer deep autoencoder approach for cross layer IoT attack detection**

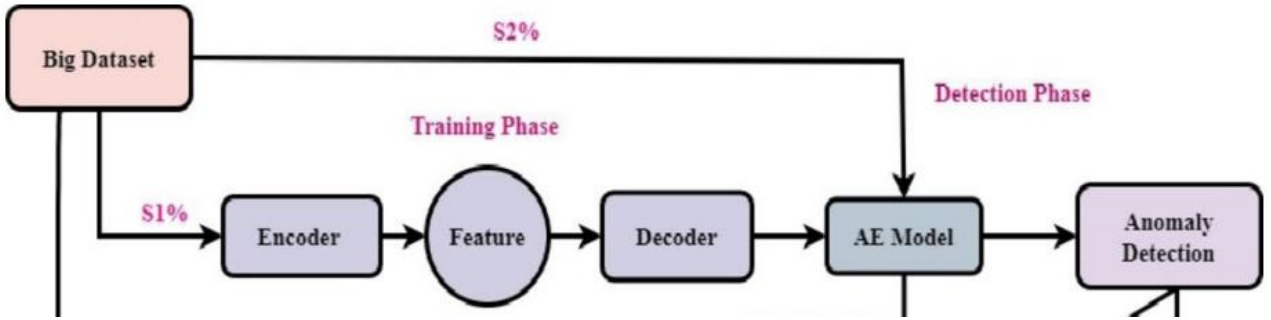
○ **Proposed method : Multi-Layer Deep Autoencoder for Anomaly detection**



AutoEncoder for Anomaly Detection

A multilayer deep autoencoder approach for cross layer IoT attack detection

○ Proposed method : Multi-Layer Deep Autoencoder for Anomaly detection

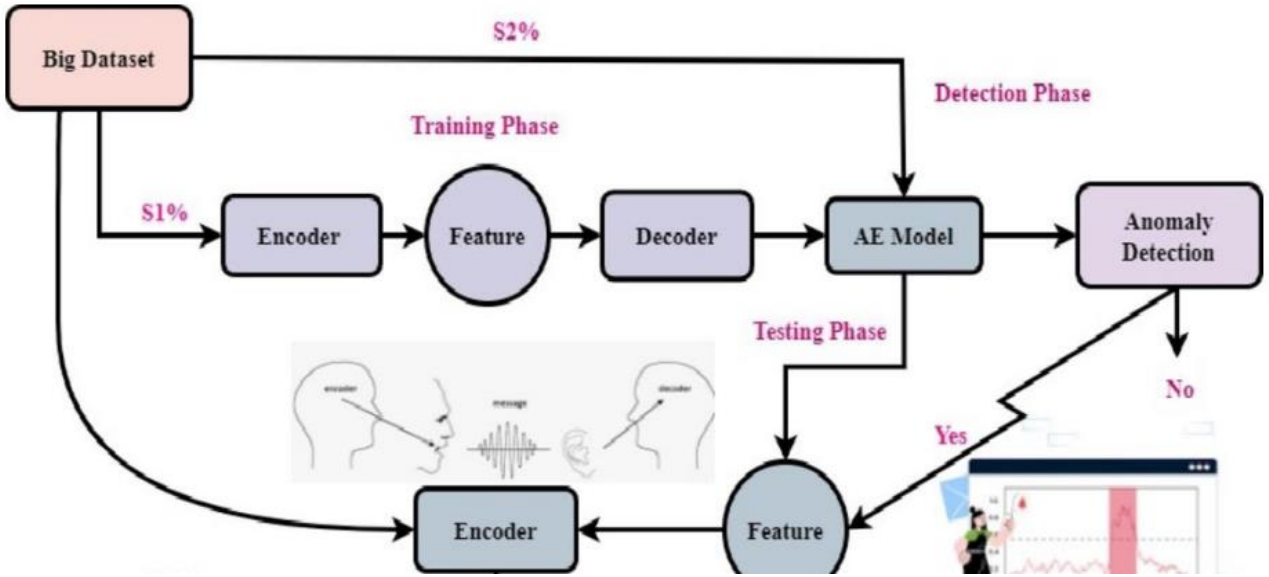


- 정상 데이터의 일부(S1%)로 AE 모델을 학습하고, 학습 완료된 AE는 저장되어 테스트 및 특징 축소 단계에 활용
- 나머지 정상 데이터(S2%)와 이상 데이터(공격 데이터)를 혼합하여 Detection을 수행함
  . Detection은 reconstruction Error로 판단함

AutoEncoder for Anomaly Detection

A multilayer deep autoencoder approach for cross layer IoT attack detection

○ Proposed method : Multi-Layer Deep Autoencoder for Anomaly detection



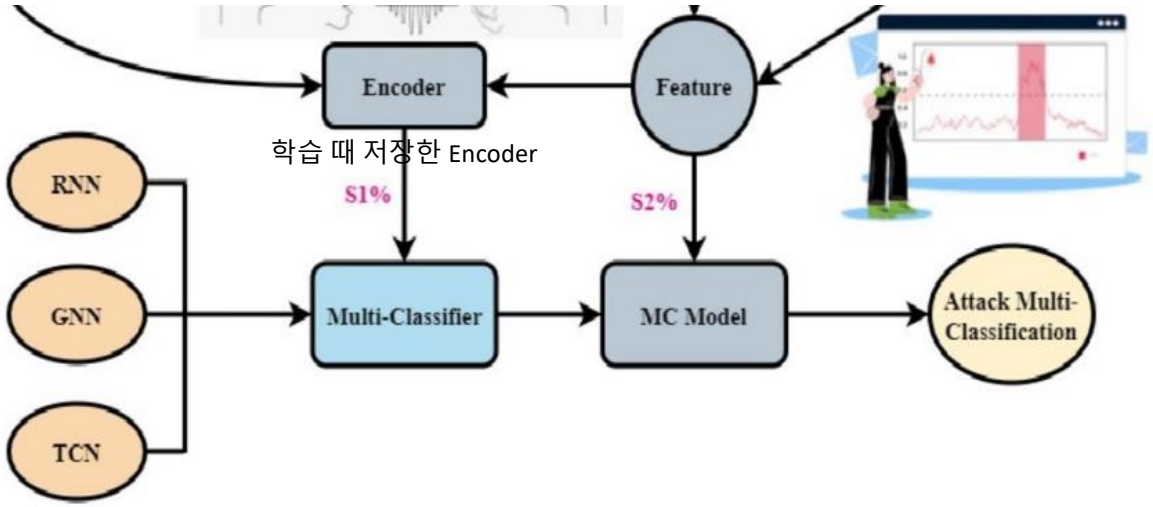
학습 때 저장한 Encoder

- Detection 결과 이상이 탐지되지 않으면 시스템은 종료
- 이상이 탐지되면, **저장된 Encoder를 사용하여 이상 데이터의 특성을 차원 축소 시킴.**

AutoEncoder for Anomaly Detection

A multilayer deep autoencoder approach for cross layer IoT attack detection

○ Proposed method : Multi-Layer Deep Autoencoder for Anomaly detection



- 차원 축소된 Feature는 다시 S1%와 S2%로 나뉨

① S1% : Multi Class model (RNN, GNN, TCN) 을 학습시키는데 사용

. RNN : 시계열, GNN : 노드 관계, TCN : 장기의존성 대응

② S2% : 학습된 Multi Class model을 테스트 하는데 사용

AutoEncoder for Anomaly Detection

A multilayer deep autoencoder approach for cross layer IoT attack detection

○ 결과

- 여러 공격 type에 대하여 우수한 성능으로 detecting 함

Approach	Detection Accuracy (%)	False Positive Rate (FPR) (%)	Computational Cost
Traditional Deep Autoencoder	85.2	12.4	High
CNN-Based IoT Attack Detection	88.5	10.2	Moderate
RNN-Based Attack Detection	90.1	9.5	Moderate-High
Proposed M-LDAE	94.8	6.8	Optimized

Table 5. Comparative analysis of approaches.

AutoEncoder for denoising

Denoising Autoencoder facilitated MEMS computational spectrometer

○ 배경

- 실리콘 포토닉 칩 기반의 소형 분광기(spectrometer)는 소형화와 집적도에서 강점이 있지만, 고해상도 구현이 어렵고 MEMS 기반 조정 장치는 노이즈에 매우 취약함
- MEMS 기기의 기계적 떨림/노이즈 때문에 스펙트럼 재구성 해상도가 저하됨
- Spectrometer란?
  . 물체가 스스로 내는 빛이나, 물체에 빛을 비추어 투과 또는 반사되는 빛을 분석하여 그 파장 분포를 살펴보아 물체의 조성을 알아내기 위한 장치
- MEMS란?
  . Micro Electro Mechanical Systems의 약자. 초소형 기계 부품과 전자 부품을 결합한 시스템. 센싱, 구동, 유체 제어 등 다양한 기능을 수행하고, 소형이기 때문에 전력 소모가 적다는 장점이 있음. 스마트폰, 자동차, 의료기기, IoT 센서 등 다양한 응용 분야가 있음
- 이 논문에서는 MEMS 기반의 Interferometer를 사용.
- Interferometer란?
  . 간섭계, 빛의 간섭 현상을 이용하여 변위를 측정하는 광학계.

AutoEncoder for denoising

Denoising Autoencoder facilitated MEMS computational spectrometer

○ 논문에서 해결하고자 하는 Task

1. 빛 조사 (Irradiation) : 백색광(광원)을 분석 대상 물질에 쏨
2. 빛의 상호 작용 : 물질이 특정 파장을 흡수, 반사, 산란. 빛의 파장별 세기 분포가 변화함
3. Interferometer 측정 : 변화된 빛을 interferometer에 통과시키면 경로차로 인해 간섭 발생 → interferogram 생성
4. 간접 데이터(interferogram) : 시간/거리 vs 세기를 나타내는 1차원 곡선. 직접적으로는 스펙트럼 해석 불가함
5. 재구성 : 푸리에 변환 혹은 **인공지능 모델을 사용하여 스펙트럼을 복원함(이 논문에서는 CAE로 함)**
6. **물질 식별 : 스펙트럼 특징을 분석하여 특정 물질의 농도나 존재 여부를 판단**

○ 논문의 목적

- MEMS interferometer는 소형화와 저전력 측면에서 매우 유리하지만 아래의 문제가 있음
  ① 기계적 진동, 열 잡음, 외란 등으로 인해 interferogram이 **쉽게 노이즈에 오염됨**
  ② **기존의 스펙트럼 복원 방식 (Fourier transform)은 노이즈에 매우 취약**
- **따라서, Convolutional AutoEncoder를 사용하여 노이즈가 섞인 interferogram으로부터 정확하고 고해상도인 spectrum을 복원하고자 함**

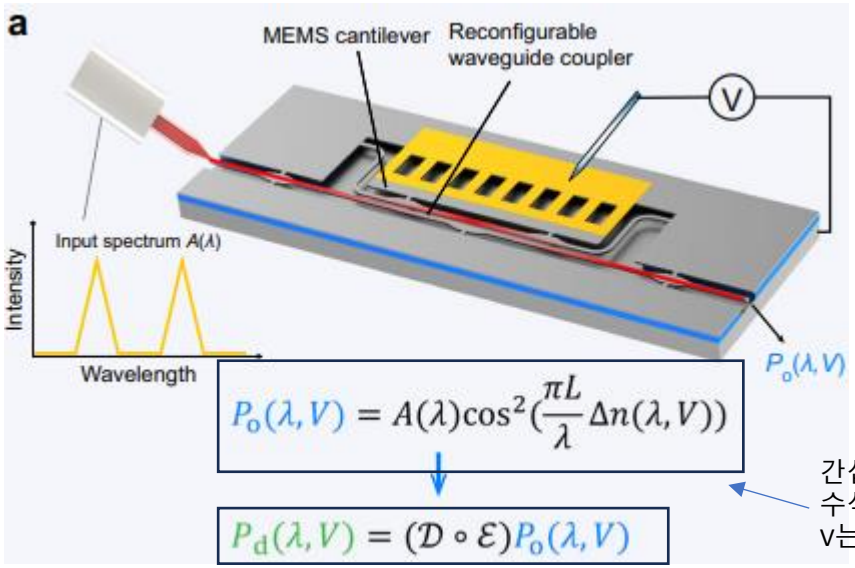
AutoEncoder for denoising

Denoising Autoencoder facilitated MEMS computational spectrometer

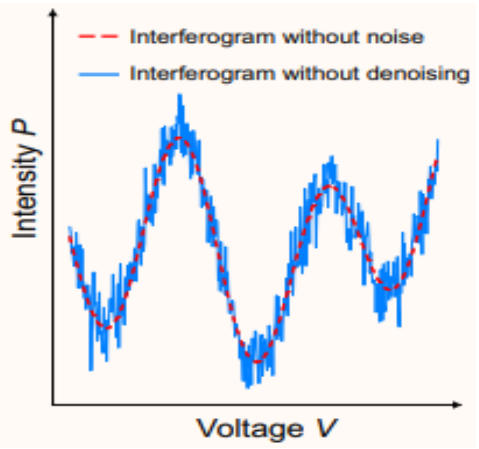
○ 전체 흐름

1. Raw Input : Noisy Interferogram

- MEMS interferometer로부터 나온 interferogram은 노이즈가 포함된 신호
- 전압에 따른 Intensity 곡선임



Noisy interferogram



간섭 출력 수식
수식의 결과는 하나의 숫자지만, 입력값인 λ, V는 노이즈로 인해 흔들리므로 결과도 Noisy함

$\mathcal{D}, \mathcal{E}$는 Decoder와 Encoder를 의미.
즉 $P_d(\lambda, V)$는 Denoised interferogram

AutoEncoder for denoising

Denoising Autoencoder facilitated MEMS computational spectrometer

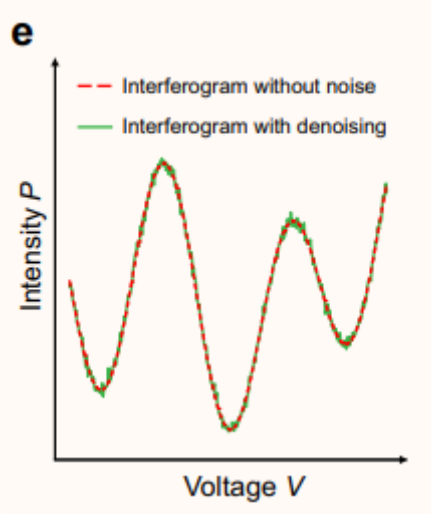
○ 전체 흐름

2. CAE Encoder

- CNN 기반의 Encoder가 interferogram의 특징을 추출
- 잡음 제거에 초점을 맞춘 latent representation 생성

3. CAE Decoder

- 압축된 특징을 다시 denoised interferogram으로 복원



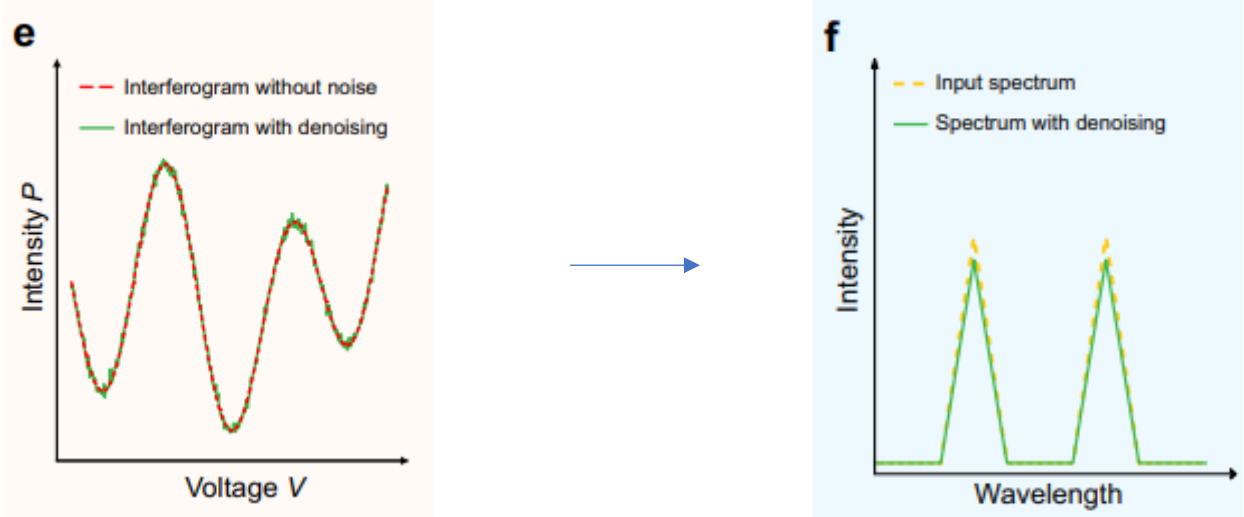
AutoEncoder for denoising

Denoising Autoencoder facilitated MEMS computational spectrometer

○ 전체 흐름

4. Denoised Interferogram을 Spectrum으로 변환

- 정규화 및 보정행렬을 적용하여 최종 스펙트럼을 reconstruction



AutoEncoder for denoising

Denoising Autoencoder facilitated MEMS computational spectrometer

○ Denoising CAE

- 보통의 Denoising AE는 original data를 복원하도록 학습시키는데, 이는 아래와 같은 Risk가 있음
  ① 만약 오리지널 데이터가 너무 복잡하면 자원이 많이 들고 underfitting 될 수 있음
  ② AE가 특정 타입의 Input에 specialized 되어 일반화 능력을 잃어버림. 즉 다른 spectrum에는 또다른 모델이 필요.
- 따라서 본 연구에서는 original data가 아닌, **noise-oriented training strategy를 채택. 즉 오리지널 데이터를 복원하지 않고, noise pattern을 복원**

different input spectra. Hereby, we employ a different, noise-oriented training strategy: instead of training the autoencoder to recover the input pattern, we recover the noise pattern and then subtract it from the initial input data (see details in Supplementary Note 9)[49]. To be

AutoEncoder for denoising

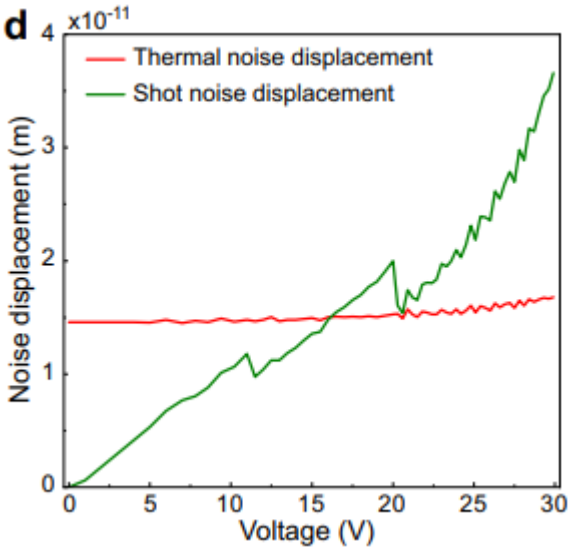
Denoising Autoencoder facilitated MEMS computational spectrometer

○ Denoising CAE

- Noisy observation I 가 있고, 그에 상응하는 original data I_a가 있으면 아래와 같이

$$I = I_a + e \qquad e : \text{noise}$$

- e가 더 간단하고 consistent한 pattern이 있으므로, Autoencoder로 하여금 e를 학습하도록 하고
- $I - e$ 를 해서 I_a를 구하는 것이, 직접 I_a를 학습하는 것보다 훨씬 더 효율적이라고 함



인가 전압에 따른 Noise의 displacement

Autoencoder는 이 Noise를 학습하는 것임

AutoEncoder for denoising

Denoising Autoencoder facilitated MEMS computational spectrometer

○ Denoising CAE

- Training Phase : Noise e와 AE의 output간의 MSE Loss를 최소화하는 encoder와 decoder의 parameter를 찾음

$$\theta^*, \theta`^* = argmin_{\theta,\theta'} \frac{1}{M} \Sigma_{i=1}^{M} \mathcal{L}\left(e^{(i)}, g_{\theta'}\left(f_\theta\left(I^{(i)}\right)\right)\right)$$

f_θ: $encoder$
$g_{\theta'}$: $decoder$
$I^{(i)}$: i 번째 $noisy\ input$
$e^{(i)}$: i 번째 $groud\ truth\ I_a^{(i)}$ 에서 $I^{(i)}$ 를 뺀 $Noise$
$\mathcal{L}$: $Loss,\ MSE\ between\ two\ inputs$

- Testing Phase : 학습된 AE가 예측한 $\tilde{e}^{(j)}$를 Noisy input $I^{(j)}$에서 뺀 결과 $\tilde{I}_a^{(j)}$를 구함

$$\tilde{I}_a^{(j)} = I^{(j)} - g_{\theta'^*}\left(f_{\theta^*}\left(I^{(j)}\right)\right)$$

Denoised signal | Noisy input | AE의 output : Noise $\tilde{e}^{(j)}$

AutoEncoder for denoising

Denoising Autoencoder facilitated MEMS computational spectrometer

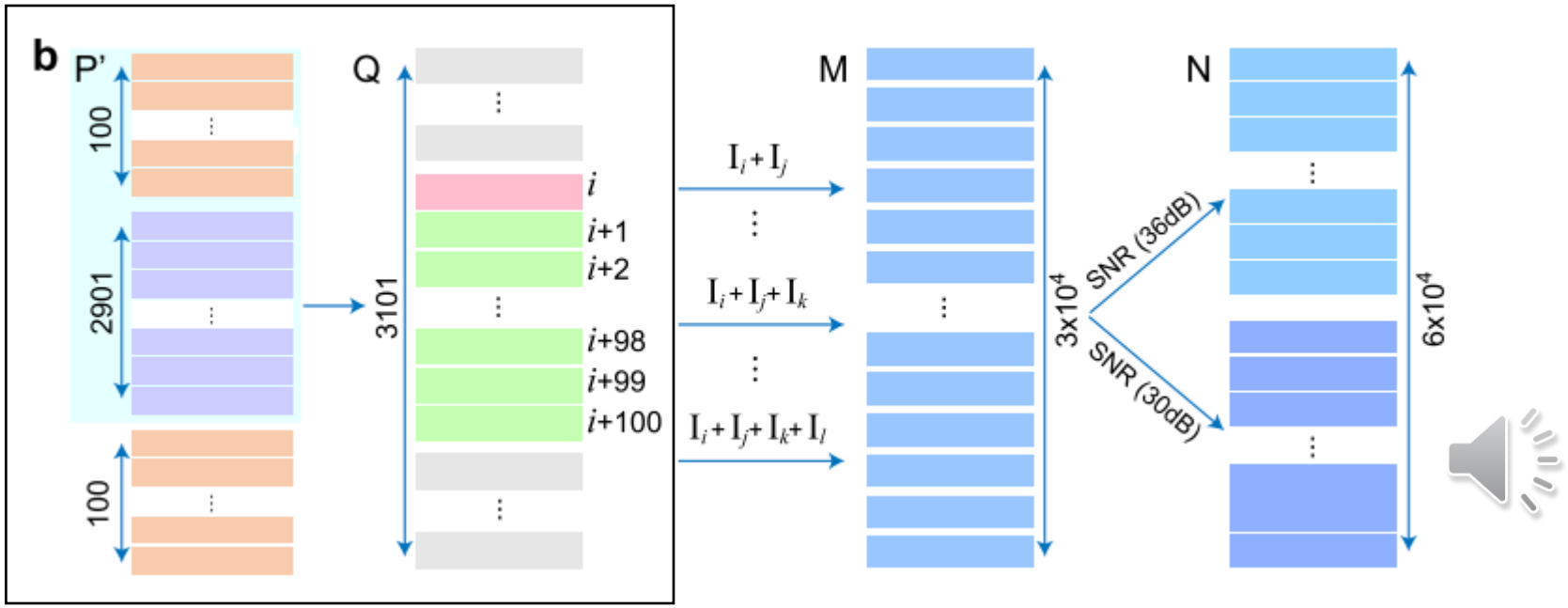
○ Denoising CAE

- Data preparation

① P′는 calibration matrix (3001x64), 각 행은 특정 파장 조합에 대한 interferogram

② P′의 앞쪽 100개 행을 끝에다가 덧붙여서 Q를 만듦 (3101x64)

- 학습용 interferogram을 만들 때 임의의 i행부터 i+100번째 행까지 선택해서 샘플링을 하는데, 임의의 i가 2,901을 초과할 경우, 100개 샘플링을 못하기 때문에 이를 막기 위함임. 예를 들어 i=2950일 때, 100개 행을 선택하면 3050인데 원래 P는 3001까지이므로 샘플링이 불가함



AutoEncoder for denoising

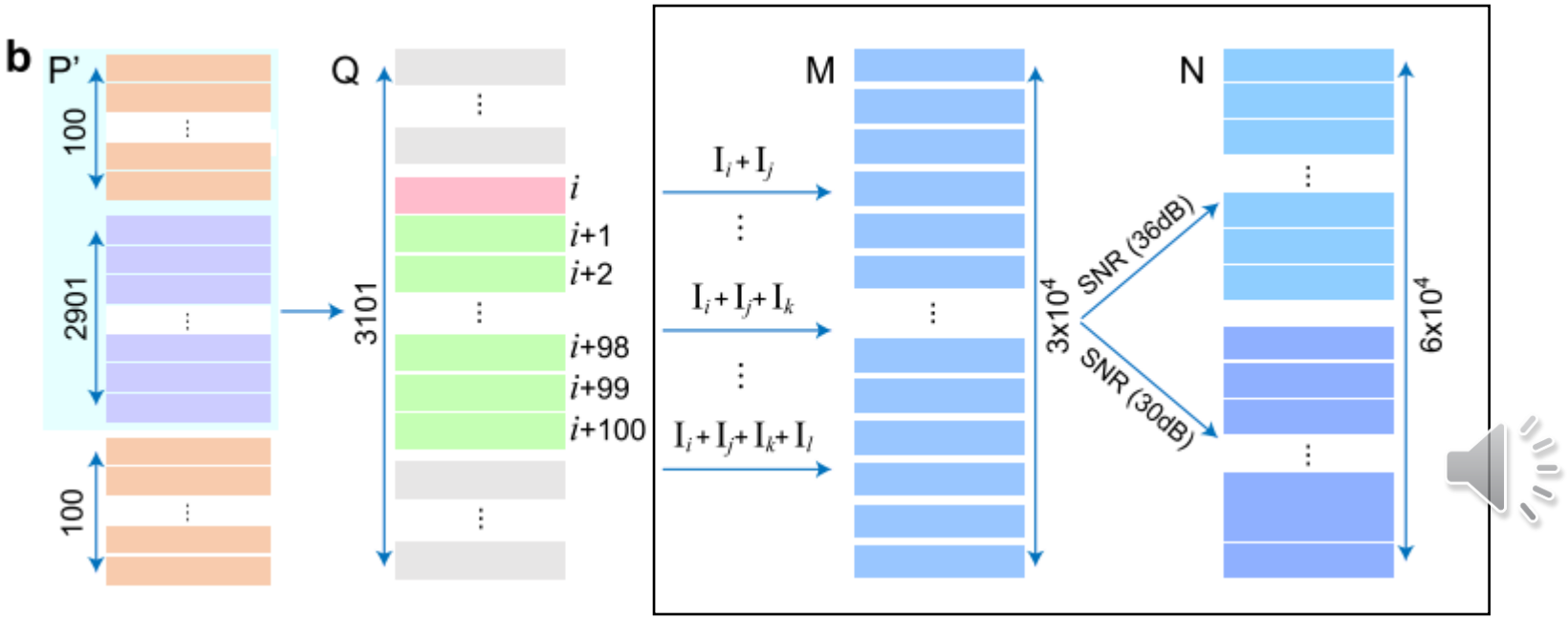
Denoising Autoencoder facilitated MEMS computational spectrometer

○ Denoising CAE

- Data preparation

③ Q에서 (i, j), (i, j, k), (i, j, k,l)을 임의로 샘플링해서 더함. 각각 10000개씩 총 3만개의 데이터셋 생성

④ 30000개의 더해진 interferogram에다가 30, 36dB의 Gaussian Noise를 추가함. 그래서 총 60000개의 학습용 데이터 준비.



AutoEncoder for denoising

Denoising Autoencoder facilitated MEMS computational spectrometer

○ Denoising CAE

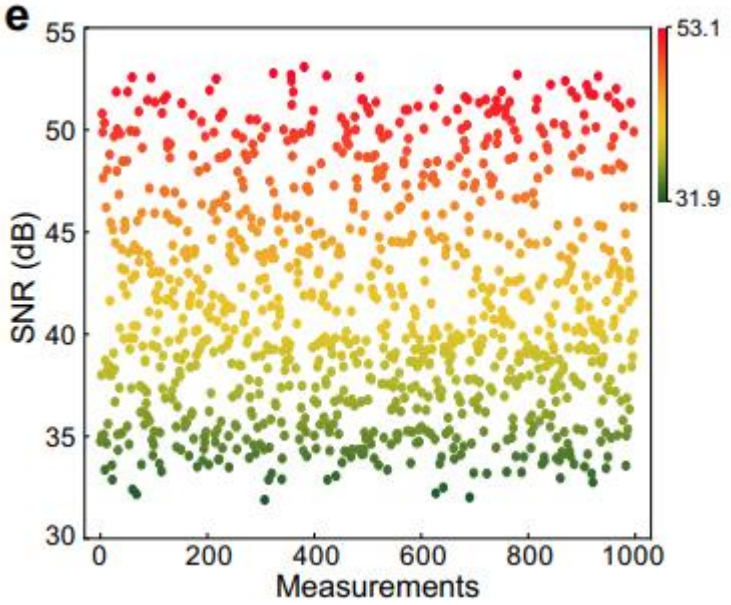
- Data preparation

⑤ Gaussian Noise를 더한 이유는, SNR(Signal to Noise Ratio) 강도가 아래와 같이 30~55dB 사이에 완전히 랜덤으로 분포하기 때문임. 여기서 SNR은 아래의 수식으로 구함. **노이즈가 많으면 분모가 커지므로 SNR이 작아짐**

$$SNR : 10log\frac{||I_1||^2}{||I_2 - I_1||^2}$$

I_1:noise free signal

I_2:noisy signal



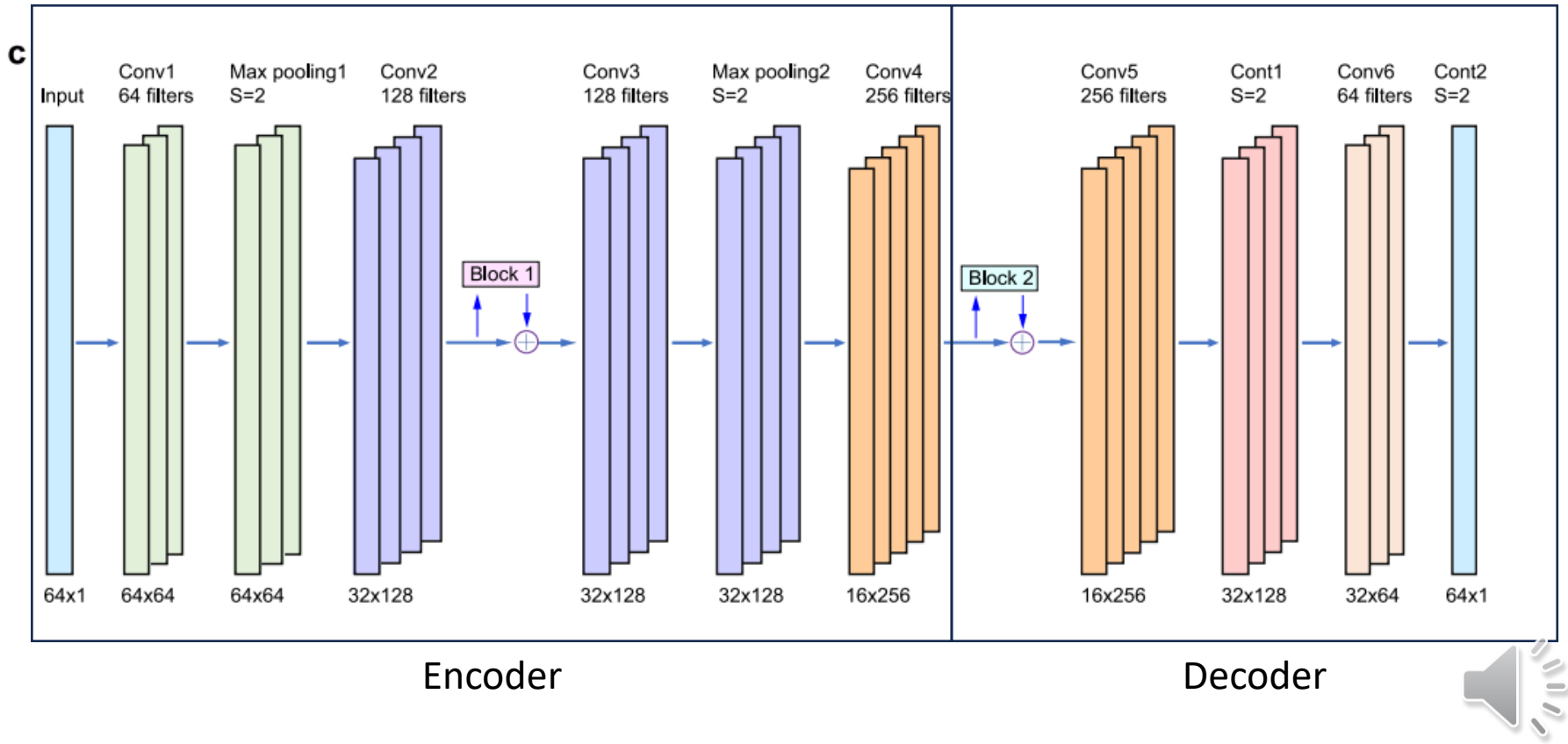
이 그래프는 1~1000번까지의 interferogram이 다양한 SNR을 가진다는 그래프임. 즉 1000개의 샘플이 랜덤으로 퍼진 SNR을 가짐. 그 말은 **Noise가 랜덤이라는 것**

AutoEncoder for denoising

Denoising Autoencoder facilitated MEMS computational spectrometer

○ Denoising CAE

- Architecture

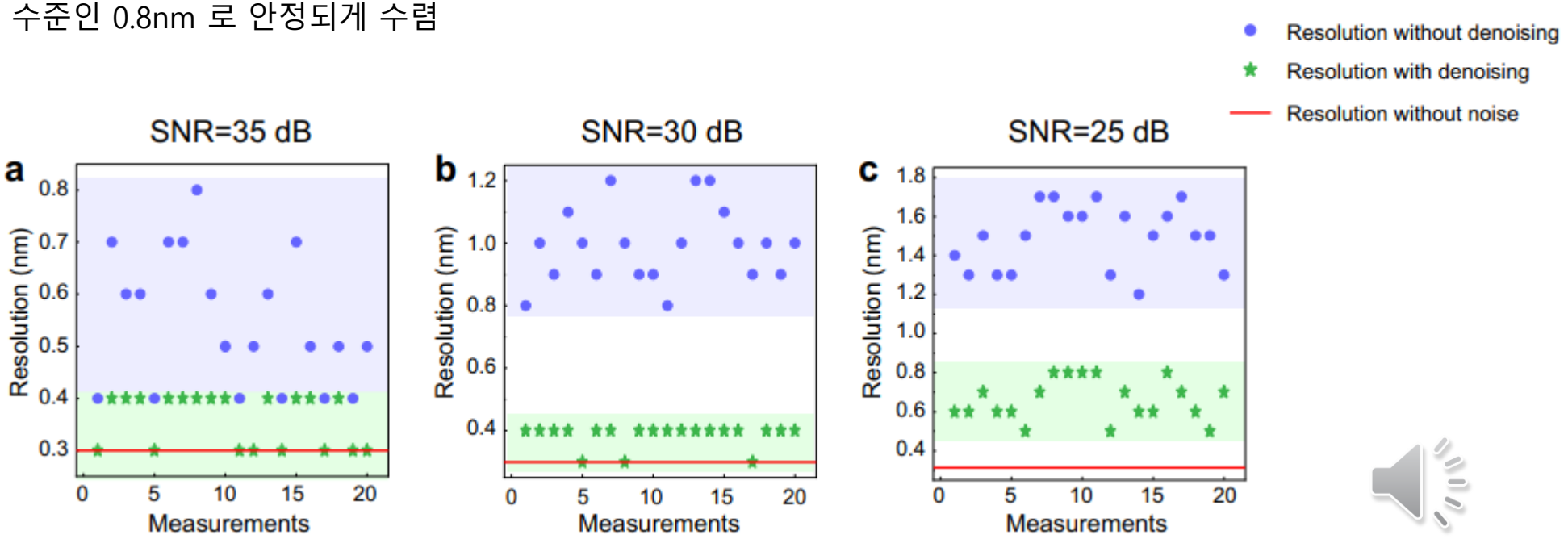


AutoEncoder for denoising

Denoising Autoencoder facilitated MEMS computational spectrometer

○ **결과 : Denoising 여부에 따른 SNR별 복원 해상도(resolution, 낮아야 좋음) 차이**

- SNR이 클 때(35dB), 즉 노이즈가 비교적 적을 때에는 Resolution이 0.4nm 이하로 비교적 안정되게 수렴
- SNR 30dB일 때, denoising을 하지 않으면 resolution이 0.8~1.2nm까지 올라가지만, denosing을 하면 0.4nm 이하로 안정됨
- SNR 25dB일 때, denoising을 하지 않으면 resolution이 1.2~1.8nm까지 올라가지만, denoising을 하면 절반 이하 수준인 0.8nm 로 안정되게 수렴

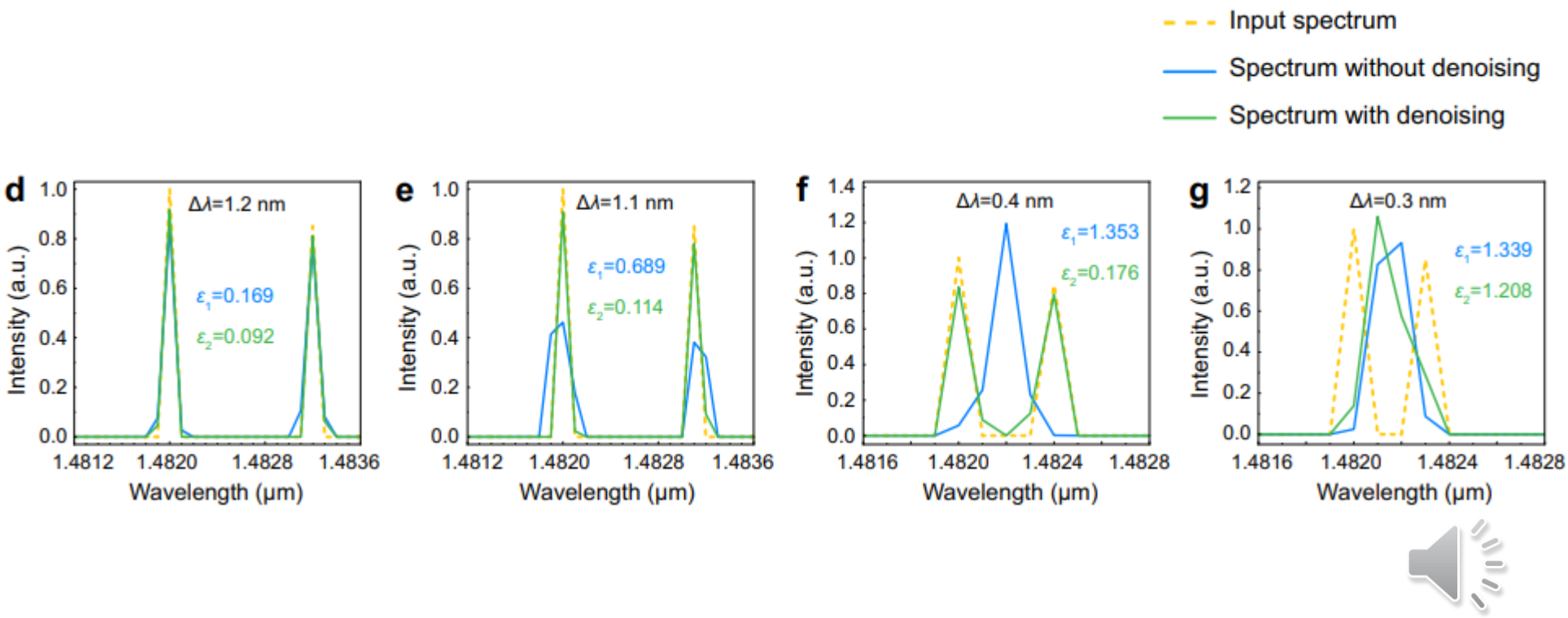


AutoEncoder for denoising

Denoising Autoencoder facilitated MEMS computational spectrometer

○ 결과 : Denoising 여부에 따른 Spectrum reconstruction 결과

- 파장이 작아질수록 denoising을 하지 않은 spectrum은 오차가 큼
- 특히 0.4nm에서는 denoising 하지 않았을 때 input spectrum의 peak를 완전히 다르게 포착함



Summary

고맙습니다.