Dinomaly

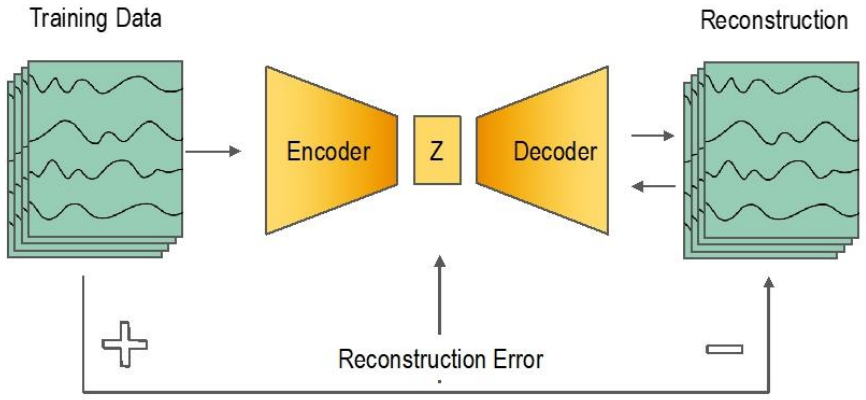
(The Less Is More Philosophy in Multi-Class Unsupervised Anomaly Detection)

김동원

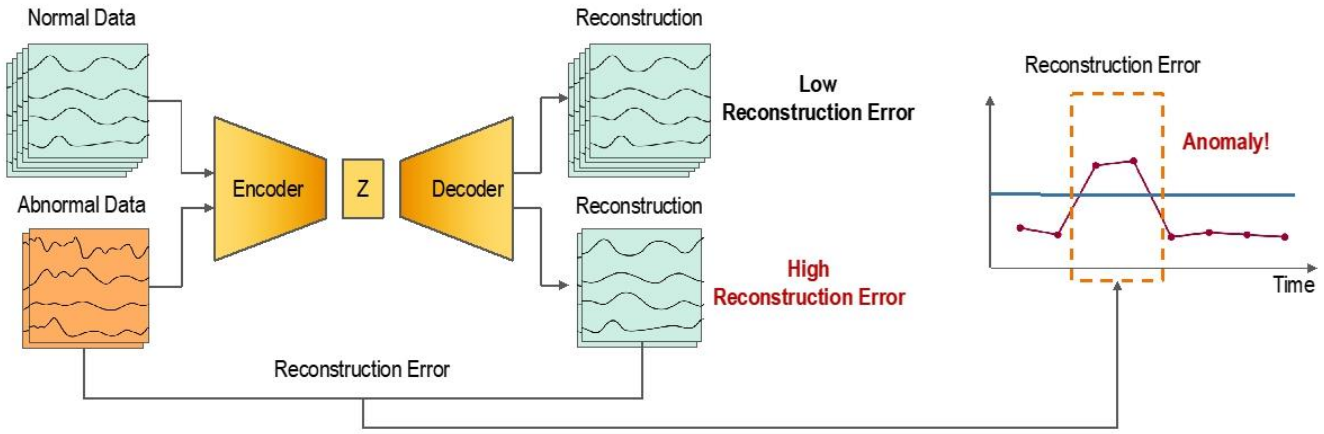
목차

Reconstruction base anomaly detection

Train



inference



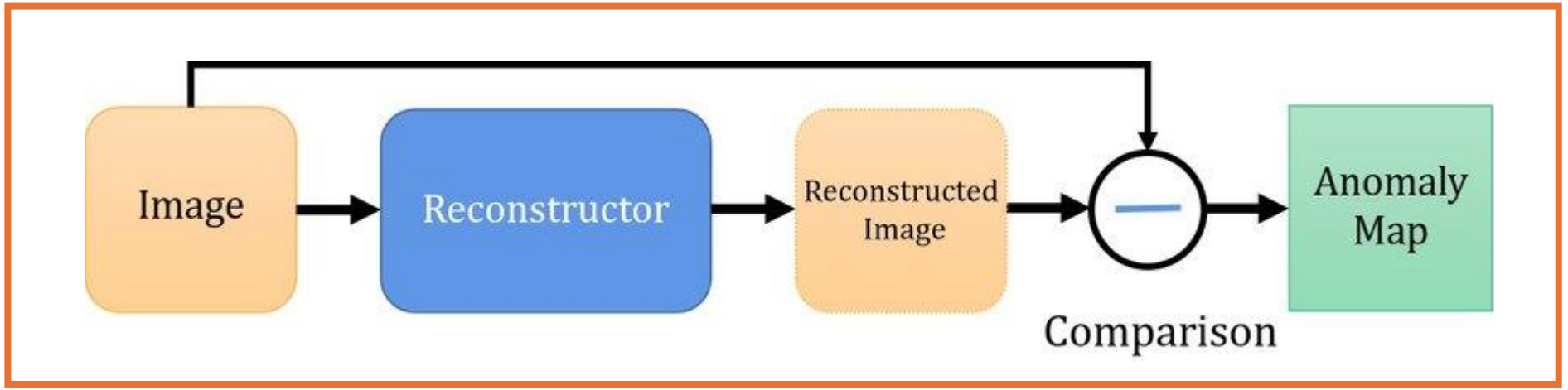
Reconstruction base method를 이용한 시계열 이상치 탐지에 관한 그림(auto-encoder)

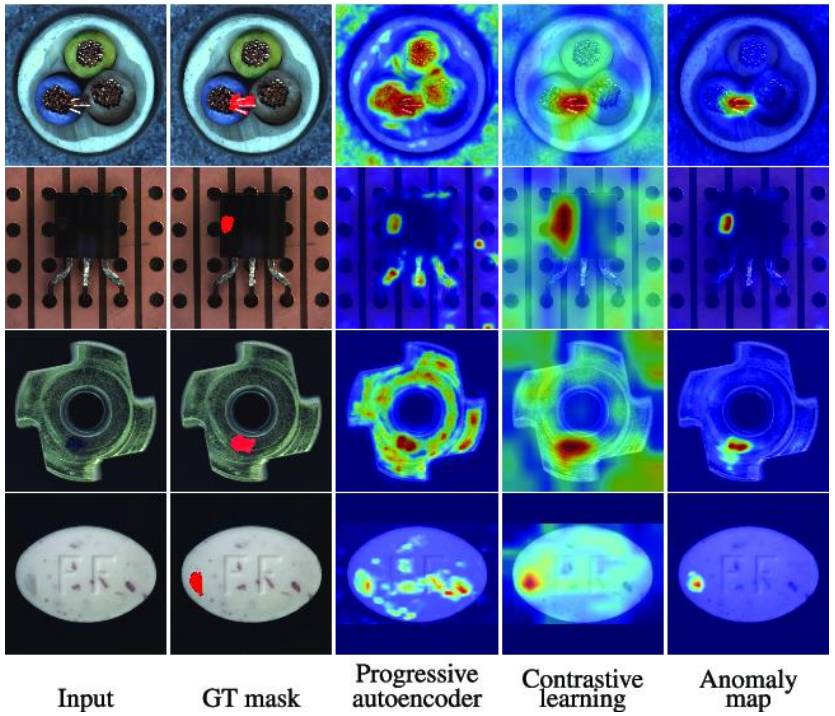
Train, 정상이미지를 이용해 학습,
입력 이미지와 복원 이미지의 차이를 줄이도록 학습

inference 시 encoder decoder은 freeze,
입력이미지와 복원 이미지의 차이를 계산하였을 때 정상 이미지는 차이가 적고, 이상 데이터는 차이가 큰 것을 이용해 이상데이터를 판단

Train시 decoder는 encoder layer의 featuremap과 코사인 유사도를 최대화 시키는 방향을 학습

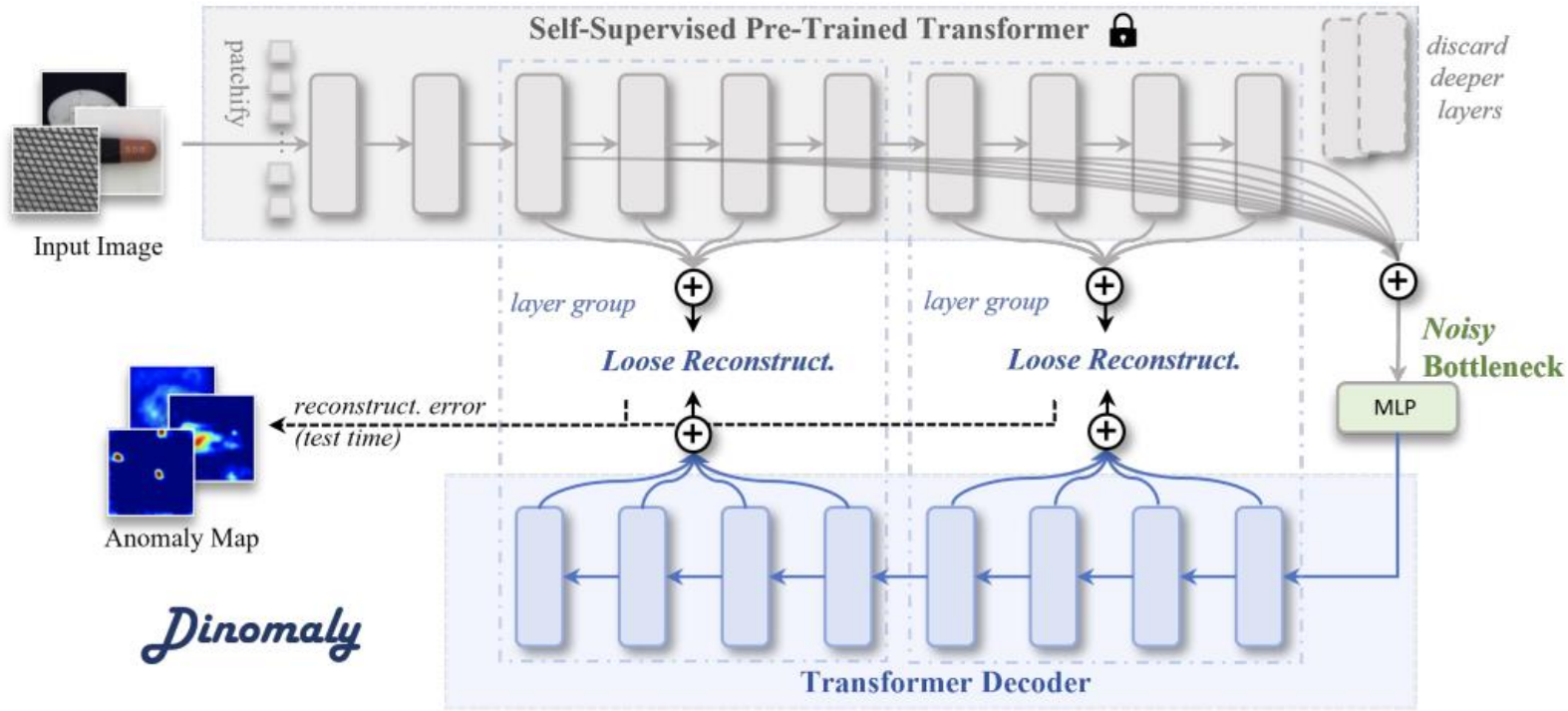
Reconstruction base anomaly detection





Class	Image Size	Normal Num	Defective Num	Defective class	Total
bottle	900 x 900	229	63	3	292
cable	1024 x 1024	282	92	8	374
capsule	1000 x 1000	242	109	5	351
carpet	1024 x 1024	308	89	5	397
grid	1024 x 1024	285	57	5	342
hazelnut	1024 x 1024	431	70	4	501
leather	1024 x 1024	277	92	5	369
metal_nut	700 x 700	242	93	4	335
pill	800 x 800	293	141	7	434
screw	1024 x 1024	361	119	5	480
tile	840 x 840	263	84	5	347
toothbrush	1024 x 1024	72	30	1	102
transistor	1024 x 1024	273	40	4	313
wood	1024 x 1024	266	60	5	326
zipper	1024 x 1024	272	119	7	391

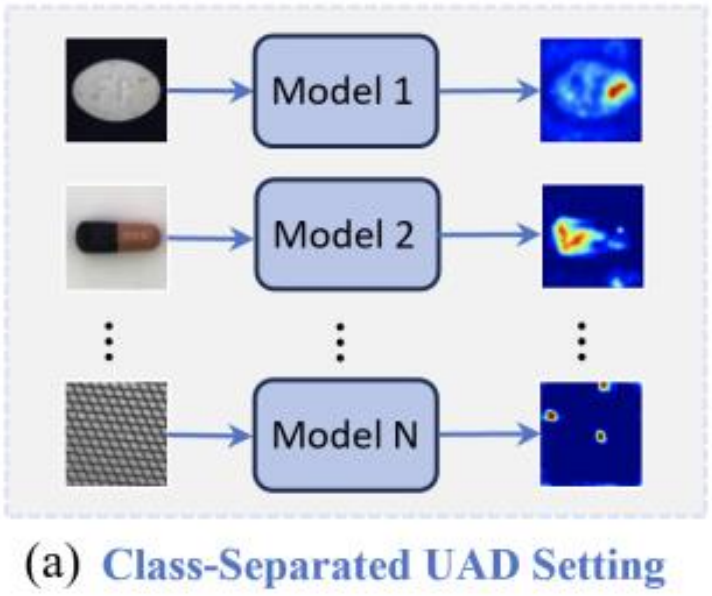
Dinomaly 구조

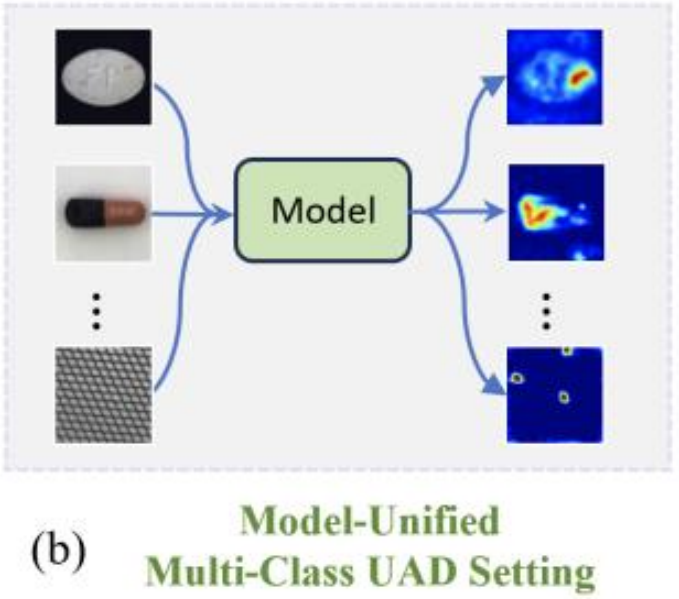


Dino v2으로 학습한 ViT encoder를 사용한다 encoder layer의 feature map을 decoder 레이어가 코사인 유사도가 최대가 되도록 학습한다

MUAD(Multi-class unsupervised anomaly detection)

Class-separated model vs MUAD

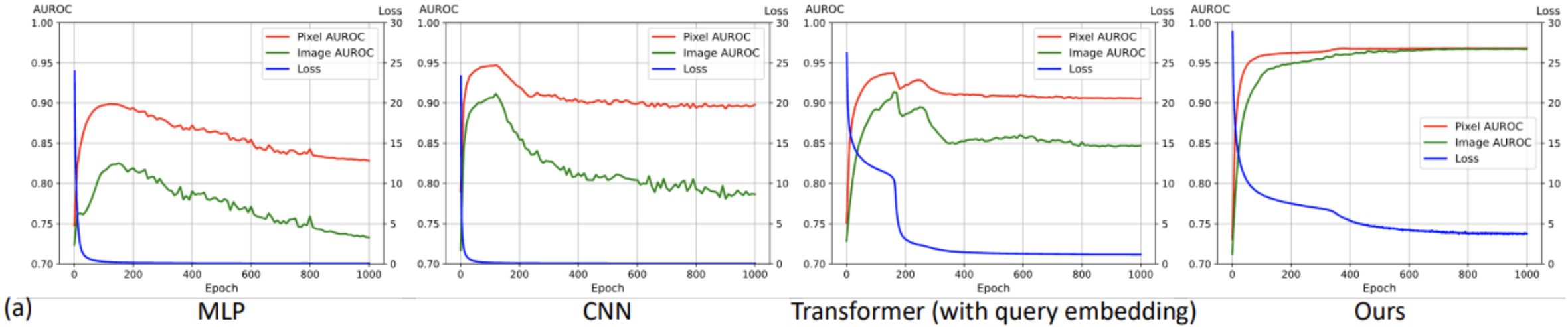




메모리와 학습의 효율을 위해 MUAD를 연구, class separated model보다 성능이 떨어짐

Identity mapping(identical shortcut)

Identity mapping이란 모델이 이상 이미지에서도 잘 복원해내는 문제가 발생하는 것을 의미



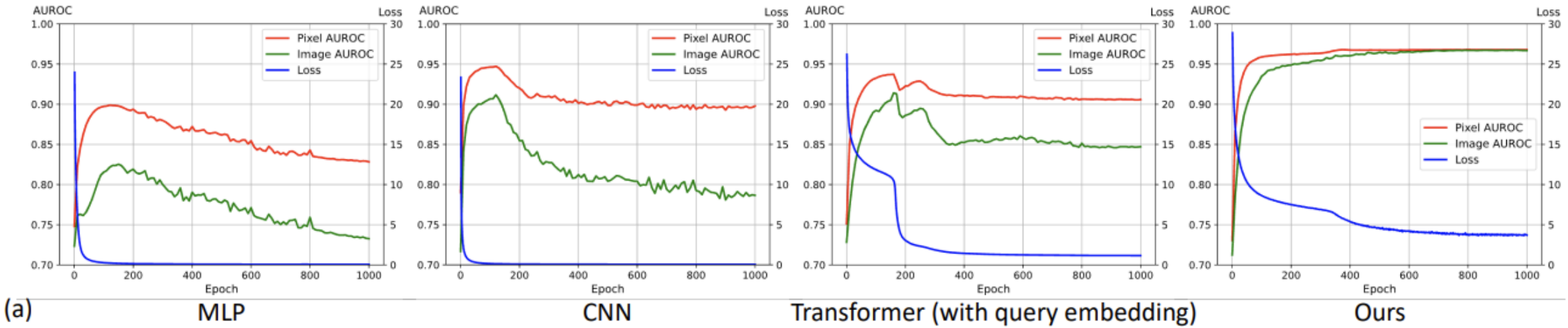
Dinomaly - Identity mapping(identical shortcut)

Dinomaly 논문에서의 identity mapping 원인

- multi-class 이미지에 대한 변동
- 정상 이미지 내에서의 변동

정상 이미지의 변동(패턴)과, 이상 이미지의 변동(패턴)을 일반화 시켰기 때문(over-generalize)

+ overfit 과 비슷한 관점에서 접근 -〉 학습을 방해하는 구조를 추가하여 loss는 커져도 정상 이미지의 패턴을 잘 학습시키게

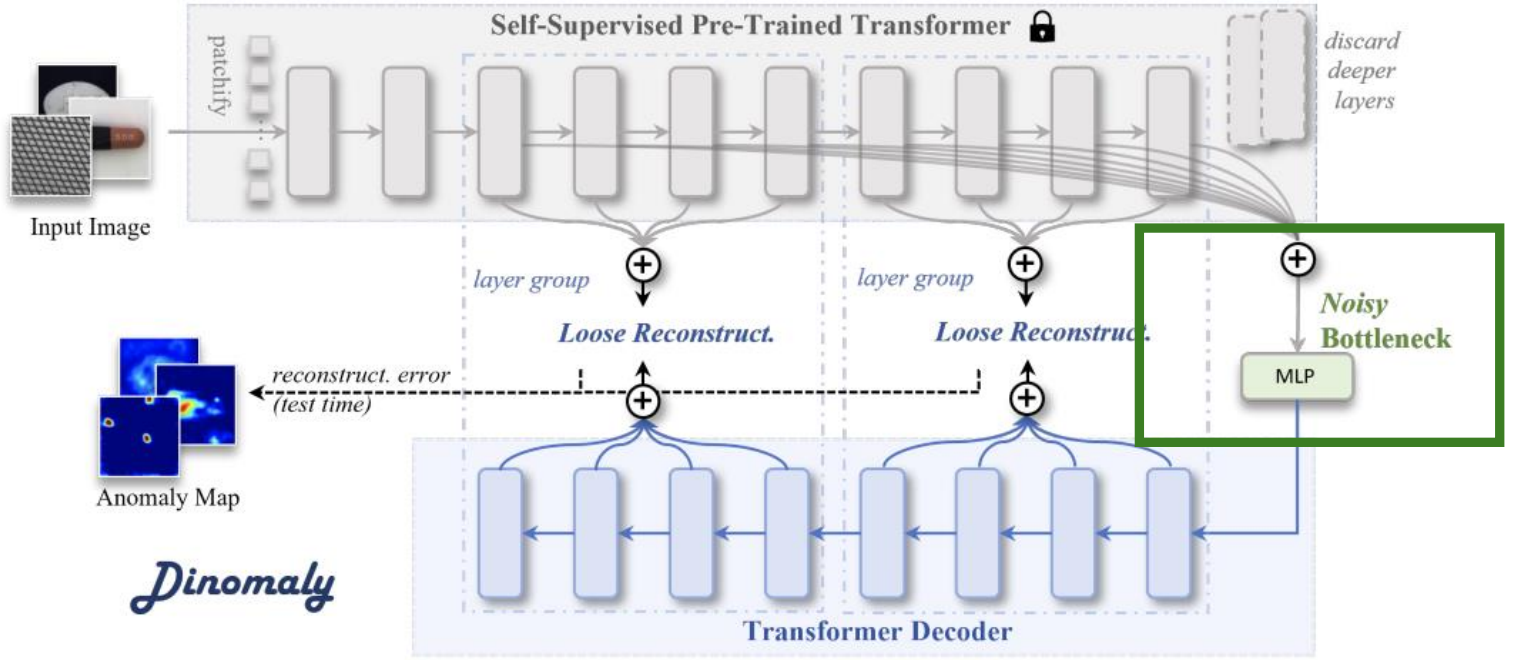


Dinomaly- 1 Noisy Bottleneck(NB

Encoder이후 MLP구조를 추가하여 dropout을 적용시킨 Noisy Bottleneck을 추가

Dropout을 적용함으로 입력된 정상 이미지에 노이즈를 추가하고 이를 다시 decode에서 복원하는 과정이 추가 되므로 denoising과 같은 역할

정상이미지의 변동에 대해서 denoising하는 효과를 가지게 된다



Dinomaly- 2 Linear Attention(LA)

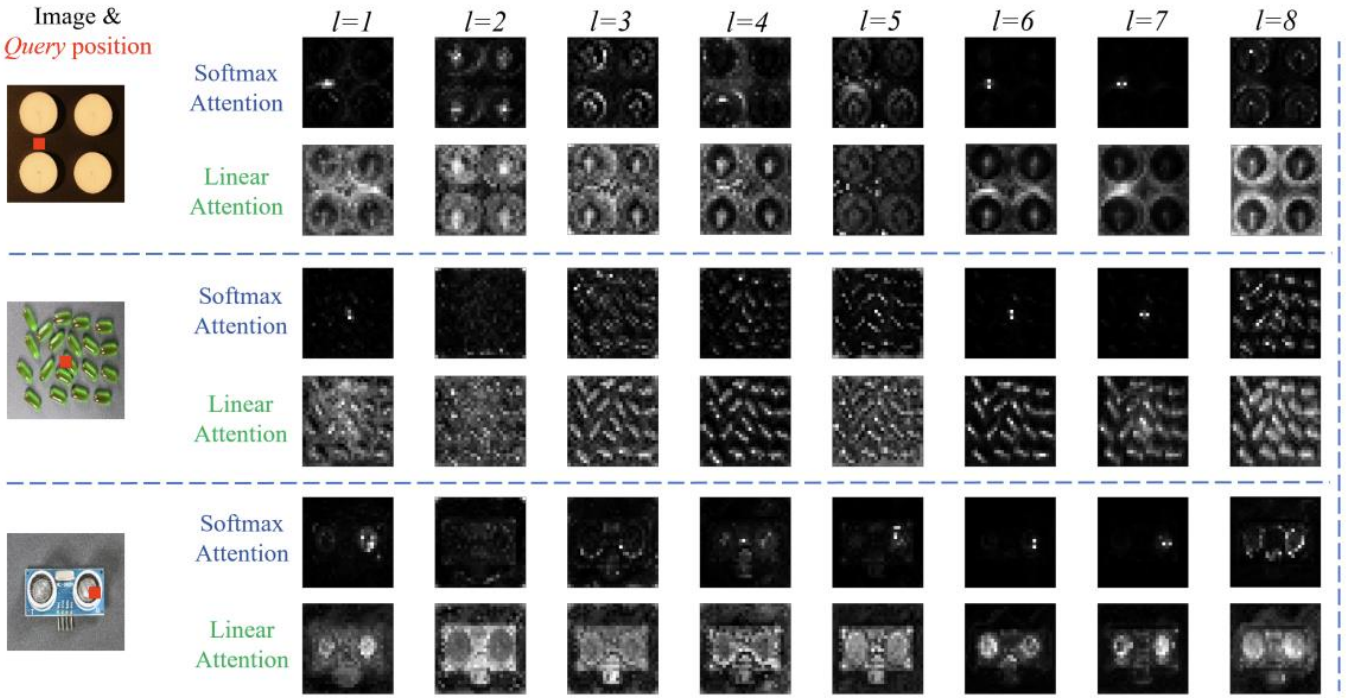
Linear Attention : Decoder transformer에 linear한 activation을 적용한다

Softmax attention을 거친 feature map을 보면 한곳에 집중되는 모습을 보인다

〉〉〉 전역적인 특징이 다음 layer로 전달되는게 쉽지 않다

〉〉〉 이상데이터가 입력되었을 경우, 이상데이터의 패턴이 깊은 layer까지 전달되어 임의로 재구성되는 것을 막는다

+ Linear attention이 본 목적인 연산 복잡도 감소 효과까지 얻게 된다.



attention map of decoder layer *l* (mean of 12 heads)

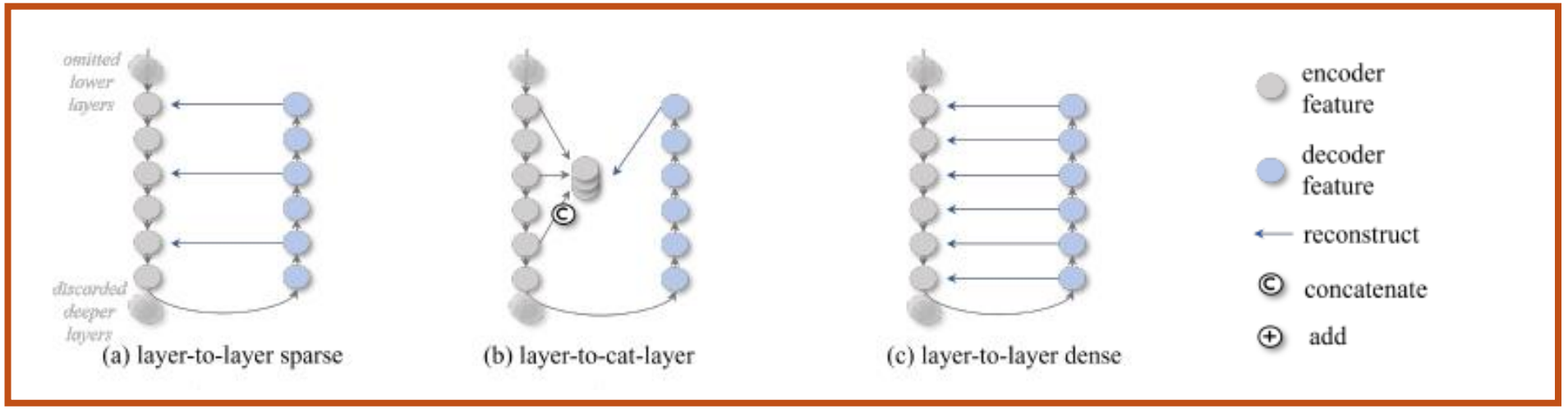
concerning the number of tokens [26]. By substituting Softmax operation with a simple activation function $\phi(\cdot)$ (usually $\phi(x) = \text{elu}(x) + 1$), we can change the computation order from $(\mathbf{Q}\mathbf{K}^T)\mathbf{V}$ to $\mathbf{Q}(\mathbf{K}^T\mathbf{V})$. Formally, Linear Attention (LA) is given as:

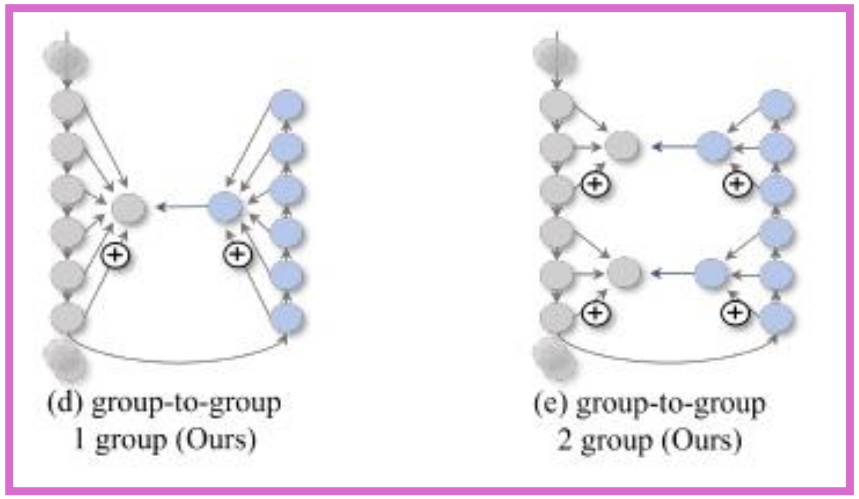
$$\text{LA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = (\phi(\mathbf{Q})\phi(\mathbf{K}^T))\mathbf{V} = \phi(\mathbf{Q})(\phi(\mathbf{K}^T)\mathbf{V}) , \quad (3)$$

where the computation complexity is reduced to $\mathcal{O}(Nd^2)$ from $\mathcal{O}(N^2d)$. The trade-off between complexity and expressiveness is a dilemma. Previous studies [15, 48] at-

Dinomaly- 3 Loose Construction(LC)

대부분의 reconstruction 모델은 encoder layer와 decoder layer를 그림과 같은 방법으로 대응시켜 decoder가 각각의 encoder layer를 따라하게 학습하는데, encoder와 연결이 많아지면 학습이 over fit하게 진행되어 다음과 같은 방법을 사용한다.





Decoder의 더 많은 자유도를 주므로 입력된 이상 데이터에서의 unseen pattern에 대해 encoder와 다르게 행동할 수 있게 한다

Dinomaly- 4 Loose Loss(LL)

$$\mathcal{L}_{global-hm} = \mathcal{D}_{cos}(\mathcal{F}(f_E), \mathcal{F}(\hat{f}_D)), \quad (4)$$

: Encoder와 decoder의 차이를 구하는 loss함수

$$\hat{f}_D(h,w) = \begin{cases} sg(f_D(h,w))_{0.1}, \text{ if } \mathcal{D}_{cos}(f_D, f_E) < k\%_{batch} \\ f_D(h,w), \text{ else} \end{cases} \quad (5)$$

: 코사인 거리가 적은 batch의 k%의 가중치를 10%감소시킨다.

$$\mathcal{D}_{cos}(a,b) = 1 - \frac{a^T \cdot b}{\|a\| \ \|b\|}, \quad (6)$$

where $\mathcal{D}_{cos}$ denotes cosine distance, $\mathcal{F}(\cdot)$ denotes flatten

〉〉〉 어느 정도 코사인 유사도가 줄어들게 학습한 feature point의 가중치를 줄이므로 전체 loss는 증가하지만 전역적으로 학습할 수 있도록 한다

Dinomaly 실험

논문에서 추가한 Noisy bottleneck(NB), linear attention(LA), loss constraint(LC), loose loss(LL)에 따른 성능 향상 정도를 확인

NB	LA	LC	LL	Image-level			Pixel-level			
				AUROC	AP	F_1-max	AUROC	AP	F_1-max	AUPRO
				98.41	99.09	97.41	97.18	62.96	63.82	92.95
✓				99.06	99.54	98.31	97.62	66.22	66.70	93.71
	✓			98.54	99.21	97.62	97.20	62.94	63.73	93.09
		✓		98.35	99.04	97.43	97.10	61.05	62.73	92.60
			✓	99.03	99.45	98.19	97.62	64.10	64.96	93.34
✓	✓			99.27	99.62	98.63	97.85	67.36	67.33	94.16
✓		✓		99.50	99.72	98.87	98.14	68.16	68.24	94.23
✓		✓	✓	99.52	99.73	98.92	98.20	68.25	68.34	94.17
✓	✓	✓		99.57	**99.78**	99.00	98.20	67.93	68.21	94.50
✓	✓	✓	✓	**99.60**	**99.78**	**99.04**	**98.35**	**69.29**	**69.17**	**94.79**

NB, LL의 경우 직접적으로 성능향상에 기여하지만 LA와 LC는 NB가 존재할 경우에만 성능 향상에 기여할 수 있다.

Dinomaly 실험

Multi class UAD모델과 class separated UAD에서 모두 가장 좋은 성능을 보였다.

Dateset	Method	Image-level			Pixel-level			
		AUROC	AP	F_1-max	AUROC	AP	F_1-max	AUPRO
MVTec-AD [3]	RD4AD [10]	94.6	96.5	95.2	96.1	48.6	53.8	91.1
	SimpleNet [34]	95.3	98.4	95.8	96.9	45.9	49.7	86.5
	DeSTSeg [67]	89.2	95.5	91.6	93.1	54.3	50.9	64.8
	UniAD [60]†	96.5	98.8	96.2	96.8	43.4	49.5	90.7
	ReContrast [14]†	98.3	99.4	97.6	97.1	60.2	61.5	93.2
	DiAD [18]†	97.2	99.0	96.5	96.8	52.6	55.5	90.7
	ViTAD [65]†	98.3	99.4	97.3	97.7	55.3	58.7	91.4
	MambaAD [17]†	98.6	99.6	97.8	97.7	56.3	59.2	93.1
	Dinomaly (Ours)	**99.6**	**99.8**	**99.0**	**98.4**	**69.3**	**69.2**	**94.8**
VisA [70]	RD4AD [10]	92.4	92.4	89.6	98.1	38.0	42.6	91.8
	SimpleNet [34]	87.2	87.0	81.8	96.8	34.7	37.8	81.4
	DeSTSeg [67]	88.9	89.0	85.2	96.1	39.6	43.4	67.4
	UniAD [60]†	88.8	90.8	85.8	98.3	33.7	39.0	85.5
	ReContrast [14]†	95.5	96.4	92.0	98.5	47.9	50.6	91.9
	DiAD [18]†	86.8	88.3	85.1	96.0	26.1	33.0	75.2
	ViTAD [65]†	90.5	91.7	86.3	98.2	36.6	41.1	85.1
	MambaAD [17]†	94.3	94.5	89.4	98.5	39.4	44.0	91.0
	Dinomaly (Ours)	**98.7**	**98.9**	**96.2**	**98.7**	**53.2**	**55.7**	**94.5**
Real-IAD [54]	RD4AD [10]	82.4	79.0	73.9	97.3	25.0	32.7	89.6
	SimpleNet [34]	57.2	53.4	61.5	75.7	2.8	6.5	39.0
	DeSTSeg [67]	82.3	79.2	73.2	94.6	37.9	41.7	40.6
	UniAD [60]†	83.0	80.9	74.3	97.3	21.1	29.2	86.7
	ReContrast [14]†	86.4	84.2	77.4	97.8	31.6	38.2	91.8
	DiAD [18]†	75.6	66.4	69.9	88.0	2.9	7.1	58.1
	ViTAD [65]†	82.3	79.4	73.4	96.9	26.7	34.9	84.9
	MambaAD [17]†	86.3	84.6	77.0	98.5	33.0	38.7	90.5
	Dinomaly (Ours)	**89.3**	**86.8**	**80.2**	**98.8**	**42.8**	**47.1**	**93.9**

Method	MVTec-AD [3]			VisA [70]			Real-IAD [54]		
	I-AUROC	P-AUROC	P-AUPRO	I-AUROC	P-AUROC	P-AUPRO	I-AUROC	P-AUROC	P-AUPRO
***Dinomaly** (MUAD)*	*99.6*	*98.4*	*94.8*	*98.7*	*98.7*	*94.5*	*89.3*	*98.8*	*93.9*
Dinomaly	**99.7**	**99.9**	**95.0**	**98.9**	**98.9**	**95.1**	**92.0**	**99.1**	**95.1**
RD4AD [10]	98.5	97.8	93.9	96.0	90.1	70.9	87.1	n/a	93.8
PatchCore [45]	99.1	98.1	93.5	94.7	98.5	91.8	89.4	n/a	91.5
SimpleNet [34]	99.6	98.1	90.0	97.1	98.2	90.7	88.5	n/a	84.6

Dinomaly 실험

Model scalability

모델별로 성능이 잘 나오는 backbone이 존재하는 것은 이전의 RD4AD, ViTAD에서 확인 하였는데 Dinomaly에서는 ViT-small에서 SOTA 성능이 나왔고, ViT-Large에서는 더 좋은 성능을 보였다.

Arch.	Params	MACs	Im/s	Image-level			Pixel-level			
				AUROC	AP	F_1-max	AUROC	AP	F_1-max	AUPRO
ViT-Small	37.4M	26.3G	153.6	99.26	99.67	98.72	98.07	68.29	67.78	94.36
ViT-Base†	148.0M	104.7G	58.1	99.60	99.78	99.04	98.35	69.29	69.17	94.79
ViT-Large	275.3M	413.5G	24.2	**99.77**	**99.92**	**99.45**	**98.54**	**70.53**	**70.04**	**95.09**

input scalability

이미지 크기가 커지면 성능이 저하되는 다른 모델과 다르게 dinomaly는 이미지 크기가 변해도 비슷하게 좋은 성능을 보여준다

Method	Input Size	Image-Level	Pixel-Level
RD4AD	256^2†	**94.6**/96.5/**96.1**	**96.1**/48.6/53.8/**91.1**
	320^2	93.2/**96.9**/95.6	95.7/**55.1**/**57.5**/**91.1**
	384^2	91.9/96.2/95.0	94.9/52.1/55.3/90.8
ReContrast	256^2†	**98.3**/**99.4**/**97.6**	**97.1**/60.2/61.5/93.2
	320^2	98.2/99.2/97.5	96.8/**61.8**/**62.6**/**93.3**
	384^2	95.2/98.0/96.4	96.5/57.7/59.5/92.6
Dinomaly	280^2	**99.6**/**99.8**/**99.3**	98.2/65.2/66.3/93.6
	336^2	**99.6**/**99.8**/99.2	98.3/67.2/67.8/94.2
	392^2†	**99.6**/**99.8**/99.0	**98.4**/**69.3**/**69.2**/**94.8**

Dinomaly 실험

Attention vs convolution

Attention대신 convolution을 사용하였을 경우 성능비교
Decoder의 attention 대신 convolution으로 대체하였고, inverted bottleneck block을 사용하였다.

Spatial Mixer	Image-level			Pixel-level			
	AUROC	AP	F_1-max	AUROC	AP	F_1-max	AUPRO
ConvBlock 3×3	99.45	99.63	98.64	98.05	65.35	68.07	94.17
ConvBlock 5×5	99.41	99.62	98.86	97.99	66.64	67.47	94.24
ConvBlock 7×7	99.42	99.65	98.86	98.01	67.57	67.94	94.45
Softmax Attention	99.52	99.73	98.92	98.20	68.25	68.34	94.17
Softmax Attention w/ Neighbour-Mask $n = 1$	99.51	99.71	98.90	98.17	67.86	67.92	94.27
Softmax Attention w/ Neighbour-Mask $n = 3$	99.56	99.76	99.05	98.28	69.26	68.17	94.50
Linear Attention	**99.60**	99.78	99.04	98.35	69.29	69.17	**94.79**
Linear Attention w/ Neighbour-Mask $n = 1$	**99.60**	99.78	99.04	98.32	68.77	68.72	94.75
Linear Attention w/ Neighbour-Mask $n = 3$	**99.60**	**99.80**	**99.14**	**98.38**	**69.65**	**69.38**	94.70

Convolution보다 attentio을 사용하였을 때 성능이 좋았다. 하지만 covolution을 이용한 모델도 SOTA성능을 보였다.

참고자료link

Dinomaly-paper

UniAD(A Unified Model for Multi-class Anomaly Detection)-paper

Reconstruction based anomaly detection image

Reconstruction based anomaly detection image