Masked Autoencoders As Spatiotemporal Learners

Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, Kaiming He

NeurIPS 2022

01. Introduction

최근 딥러닝 분야에서는 다른 영역의 문제 해결을 위한 방법론을 통합하는 추세임

Transformer 아키텍처는 CV 분야의 성공적인 도입과 Language/Vision 분야에 일반적인 구조로 확립됨

Self-Supervised Representation Learning

BERT의 Denoising/Masked Autoencoding 방법은 이미지에서 효율적으로 Representation 학습이 가능함

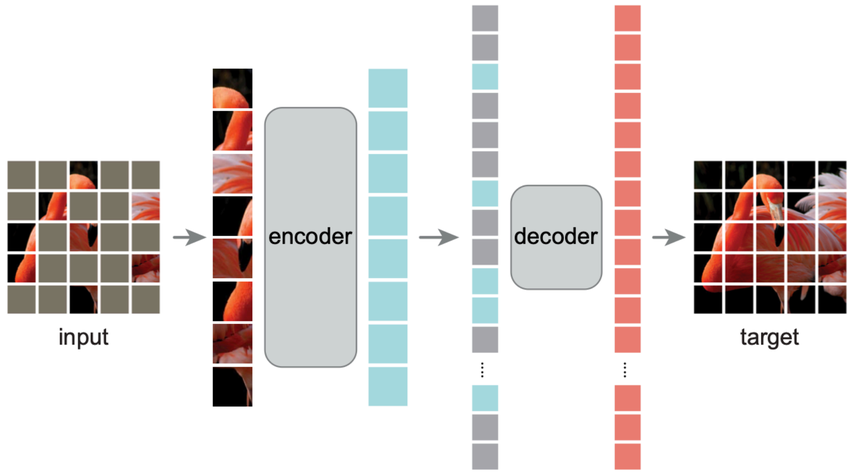
Towards Unifying Methodologies

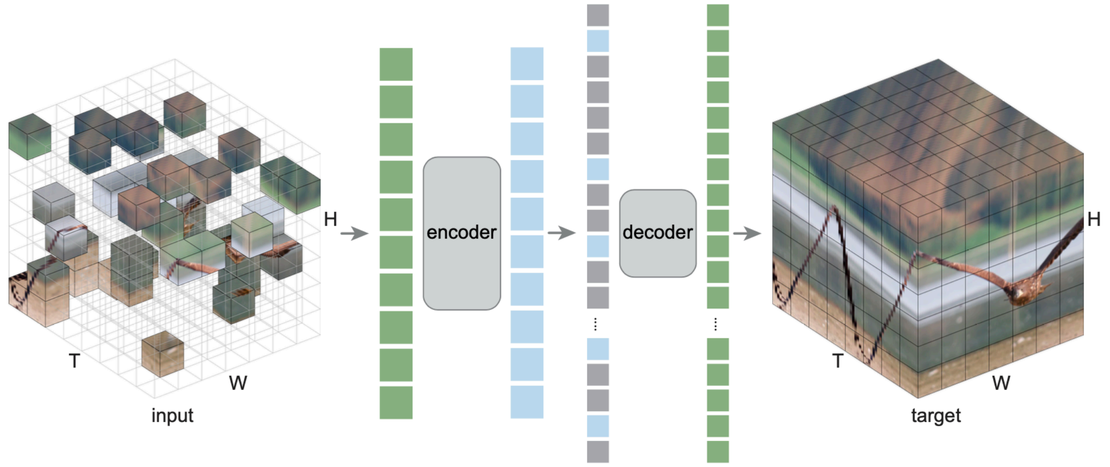
Less Domain Knowledge(Fewer Inductive Biases)는 데이터로부터 거의 순수하고 사용가능한 지식을 배움

01. Introduction

An Extension of Masked Autoencoders(MAE)

이전 배경에 따라 MAE의 Spatiotemporal Representation Learning으로의 확장





01. Introduction

Methods

Video 데이터에서 Spacetime Patches의 Randomly Masking

Autoencoder는 Pixel 영역에서 Reconstruct 방법을 학습

Minimal Domain Knowledge

Embedding Pathes와 Positions에 Spacetime-specific Bias만 존재함

Higher Masking Ratio (Percentage of Removed Tokens)

Masking Ratio는 Problem의 정보 중복성(Information Redundancy)과 관련되어 있음

높은 Masking Ratio를 통한 Computation과 Complexity 개선

01. Introduction

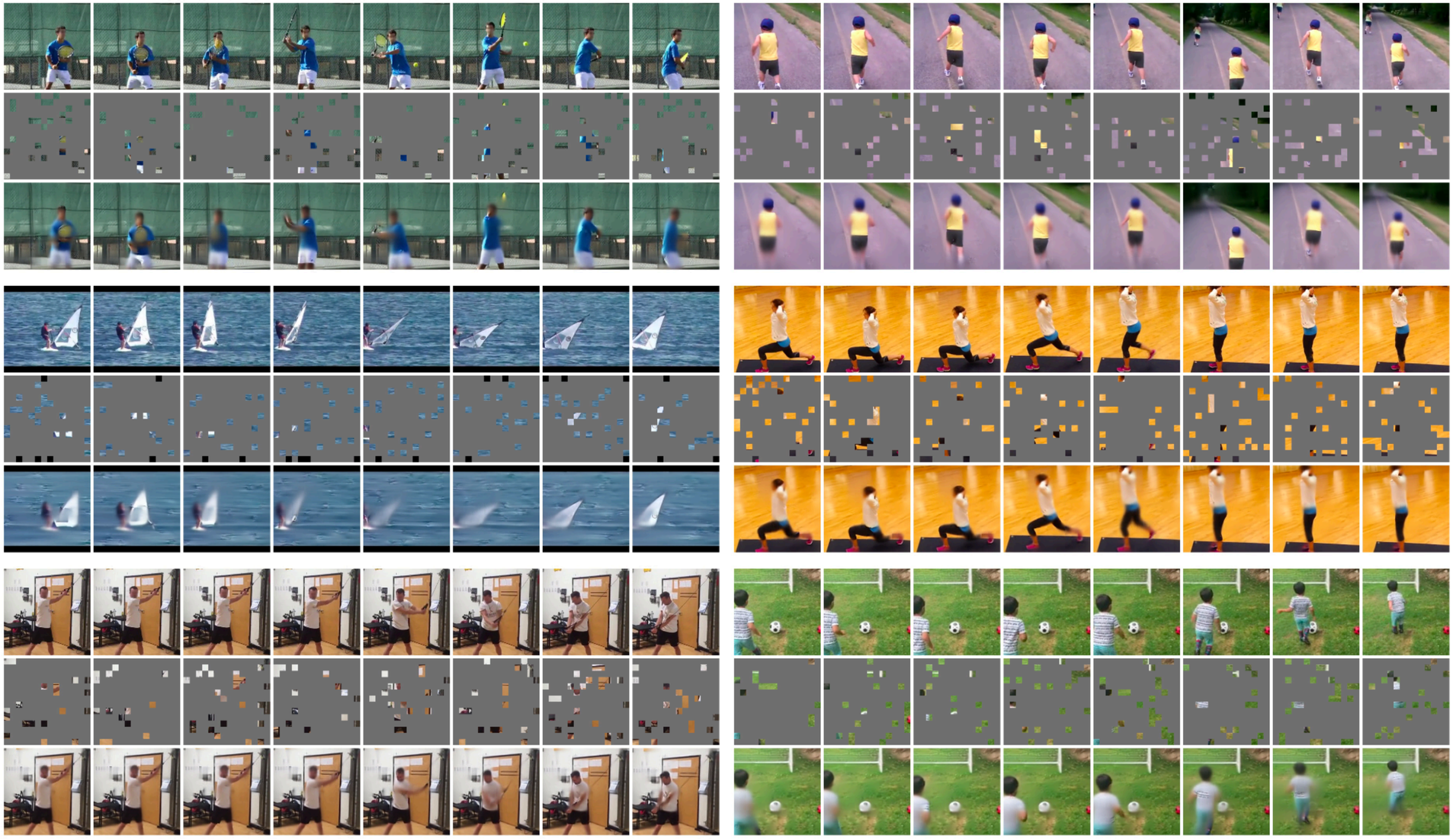


Figure 2: Visualizations on the Kinetics-400 [35] validation set (masking ratio **90%**). We show the original video (top), masked video (middle), and MAE output (bottom) for each sample. This model reconstructs the original pixels. The video size is $16\times224\times224$ and the spacetime patch size is $2\times16\times16$ (the temporal patch size of 2 is not visualized here). Each sample has $8\times14\times14=1568$ tokens with 156 being visible. For better visualizations, the known patches in the output are from the original input. Fig. 7 shows more examples.

01. Introduction

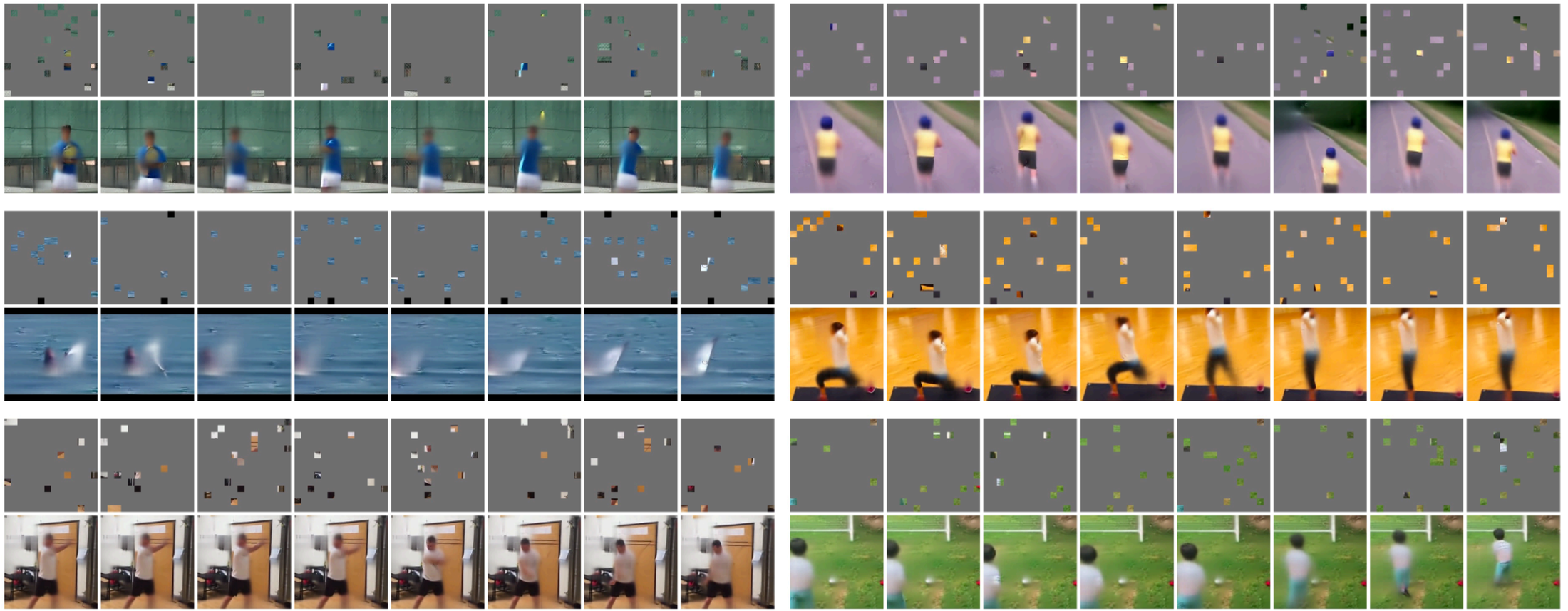
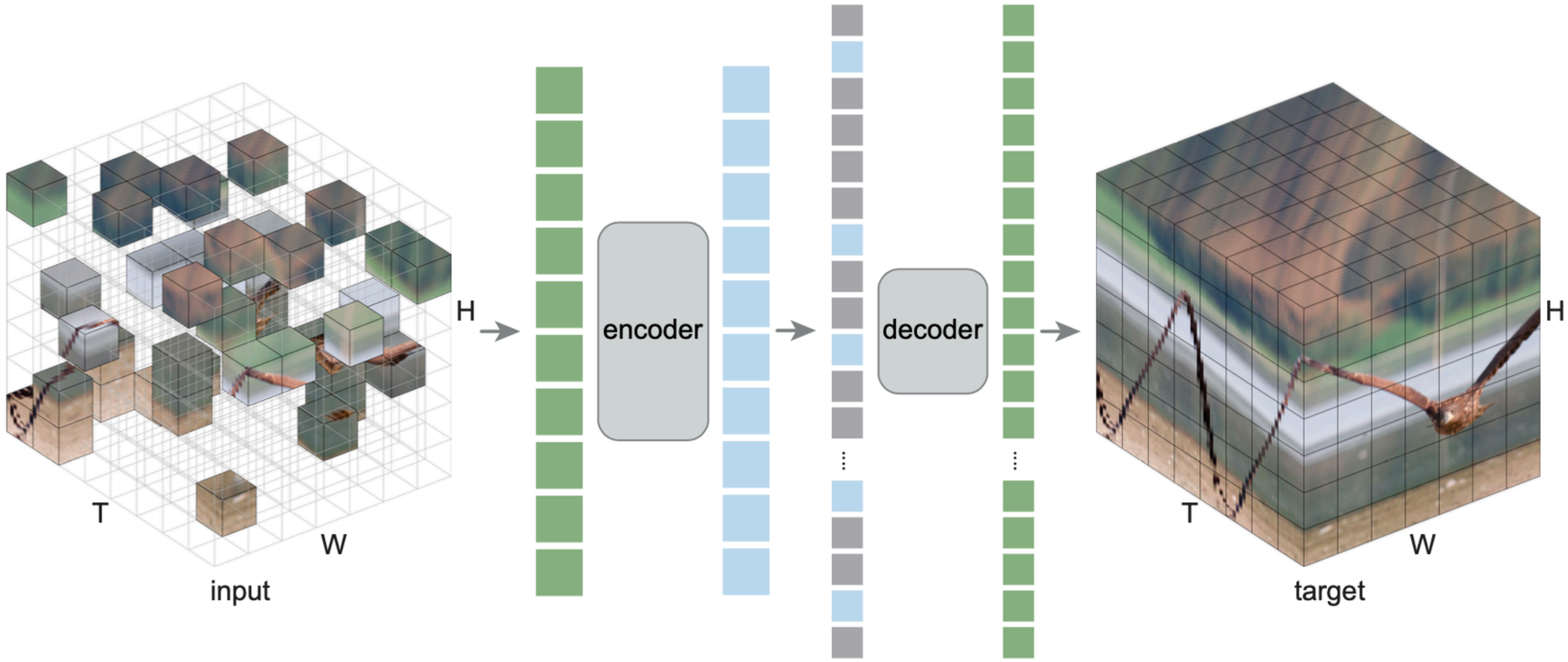


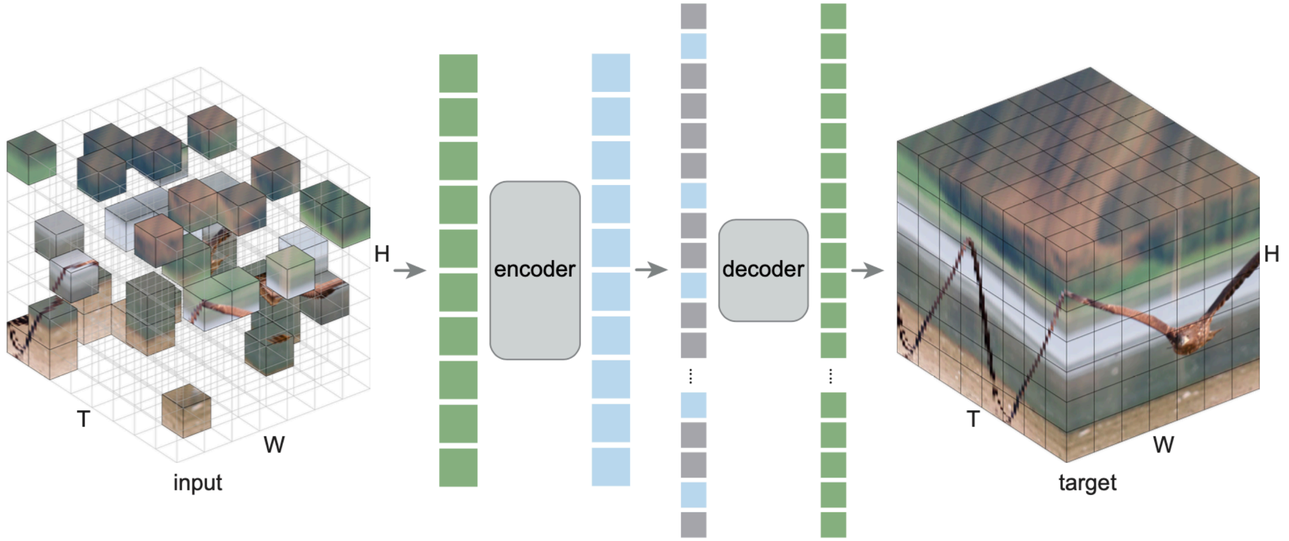
Figure 3: Visualizations of the same pre-trained model in Fig. 2 but with a masking ratio of **95%**.

02. Method

Patch Embedding / Masking / Autoencoding



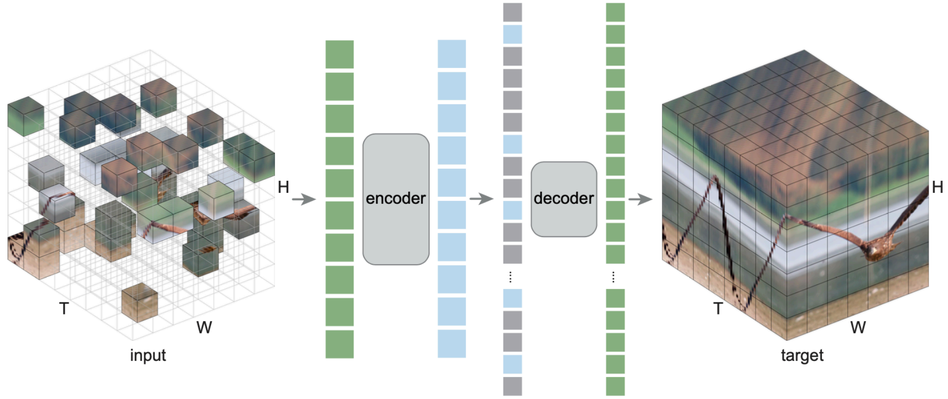
02. Method



Patch Embedding

- Original ViT와 동일한 방법을 사용
- Video Clip을 Spacetime에서 Non-Overlapping Patches로 나눔
- Patches는 Faltten 되어 Linear Projection을 통해 Embedding
- Embedded Patches에 Positional Embedding 추가
- Patch와 Positional Embedding 과정이 유일한 Spacetime-aware 과정임

02. Method



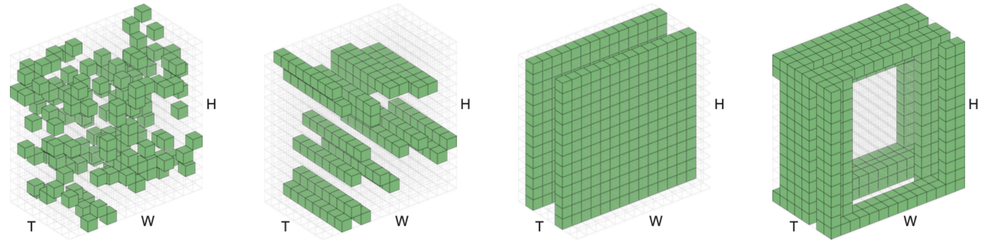
Masking

- Embedding Patches에서 랜덤하게 샘플링 (w/o replacement)
- Random Sampling은 Spacetime Structure에 Agnositc함

Optimal Masking Ratio

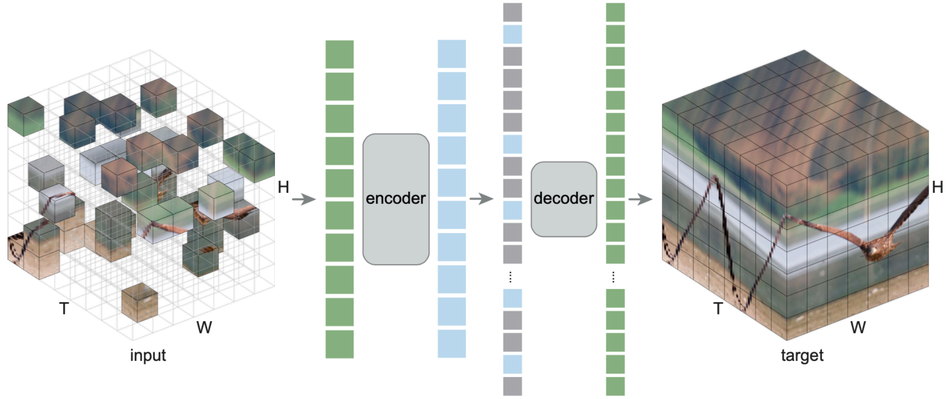
- 데이터의 정보 중복성(Information Redundancy)과 관련되어 있음
- BERT는 Language에 15%, MAE는 Image에 75%가 적용됨에 따라 Image가 정보 중복성이 높음
- Temporal Coherence 때문에 Video는 Image보다 정보 중복성이 높아 실험적으로 90%를 사용함

Spacetime-agnostic vs Structure-aware Sampling



(a) agnostic, 90% (b) space-only, 90% (c) time-only, 75% (d) block, 75%

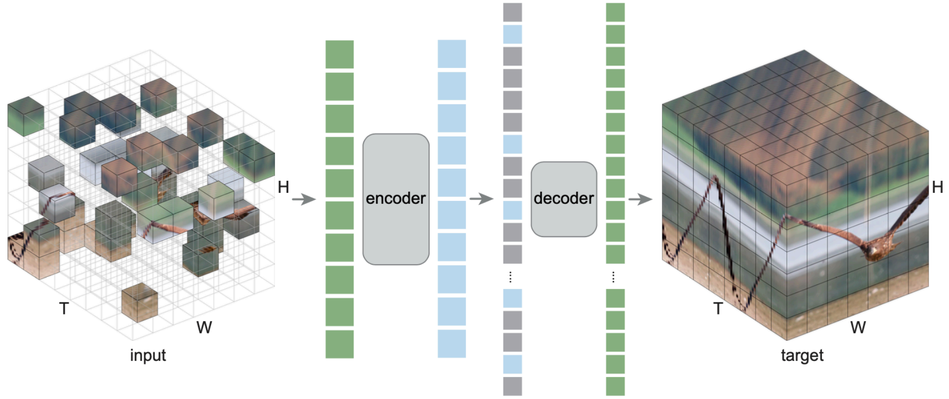
02. Method



Autoencoding - Encoder

- Vanilla ViT는 Visible Embedded Patches만 입력으로 활용함
- Time/Memory Compleixty 개선 (Masking Ratio 90%: Encoder Complexity to 1<10)
- Encoder에는 Space/Time 두 개의 분리된 Positional Embedding이 적용됨

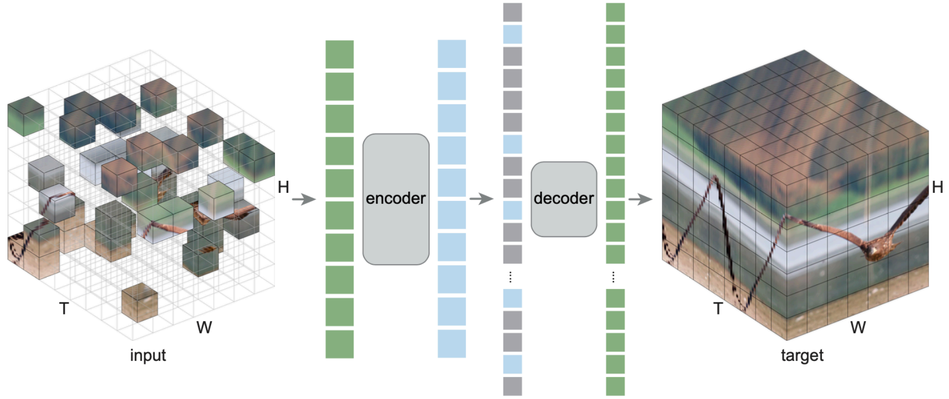
02. Method



Autoencoding - Decoder

- Another Vanilla ViT는 Encoded Patches와 Mask Tokens를 입력으로 사용함
- Decoder-specific Positional Embedding이 적용됨
- Decoder는 Encoder보다 Depth/Dimension이 작게 구성되어 Complexity Reduction이 7.7배
- Decoder는 Pixel 영역에서 Patches를 예측함 (Original Pixel or Per-Patch Normalized Values)

02. Method



Autoencoding - Encoder/Decoder

- Encoder/Decoder는 Spacetime Structure에 Agnostic함
- No Hierarchy or Spacetime Factorization
- Global Self-Attention을 통해 데이터로부터 사용가능한 Knowledge를 학습함

03. Implementation

Data Pre-Processing

- MAE Pre-Training을 위한 입력 크기: 16x224x224
- Raw Video의 랜덤한 시작 위치에서 Temporal Stride 크기 4로 16 Frames를 샘플링함: 16x4
- (Spatial) Data Augmentation: Only Random Resized Cropping [0.5, 1] and Horizontal Flipping

  *Temporal Data는 자연적으로 Augmentation을 제공함 (e.g. View Points, Motion, Deformation, Occlusion)

  *Random Temporal Sampling을 통해 자연적인 Augmentation이 통합되어짐

Bottleneck: Data Loading

- 빠른 Computation을 가진 MAE Pre-Training에 따라 Data Loading의 병목 현상
- Repeated Sampling: 하나의 Raw Video가 Loaded/Decompressed 될 때, Multiple Sampling을 수행

03. Implementation

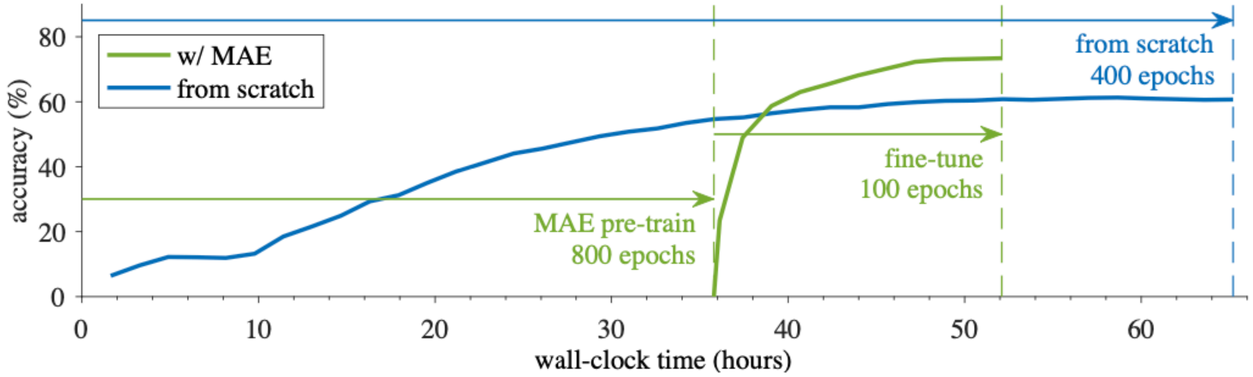
Architecture

- Encoder/Decoder는 Vanilla ViT Architecture를 활용함
- Temporal Patch Size 2와 Spatial Patch Size 16x16: 2x16x16

  e.g. 16x224x224 Input에서 8x14x14 Patches가 생성 가능함

04. Experiments

Performance



	scratch	MAE
1-view	60.7	**73.4** (+12.7)
multi-view	71.4	**84.4** (+13.0)

Figure 5: MAE pre-training plus fine-tuning is *much more accurate* and *faster* than training from scratch. Here the x-axis is the wall-clock training time (128 A100 GPUs), and the y-axis is the 1-view accuracy on Kinetics-400 validation. The table shows the final accuracy. The model is ViT-L.

04. Experiments

Performance

MAE w/	acc.	FLOPs	compute	load+compute
encoder w/ [M]	84.3	627.5 G	141.1 hr	147.5 hr
encoder w/o [M]	84.4	81.0 G	24.5 hr	35.8 hr
gain		7.7×	5.8×	4.1×

Table 1: **Training time comparison** between a dense encoder (w/ [M]) and a sparse encoder (w/o [M]) in MAE. The encoder is ViT-L (1024-d, 24-block); the decoder is our default (512-d, 4-block). With a masking ratio of 90%, the sparse variant reduces FLOPs by 7.7×. This reduces computation time by 5.8×. In our infra, computation is so fast that data loading becomes a bottleneck, which leads to an actual speedup of 4.1×. Profiling is with synchronized SGD over 16 nodes, each with 8 A100 GPUs and 80 CPU cores. The training length is 800 epochs.

04. Experiments

Ablation Experiments - Masking Ratio

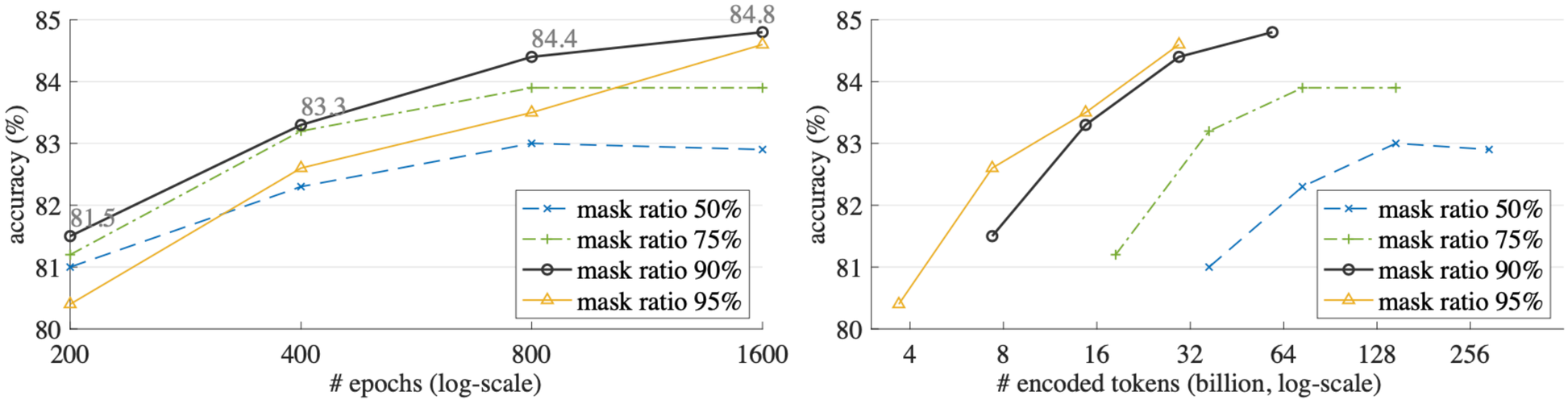


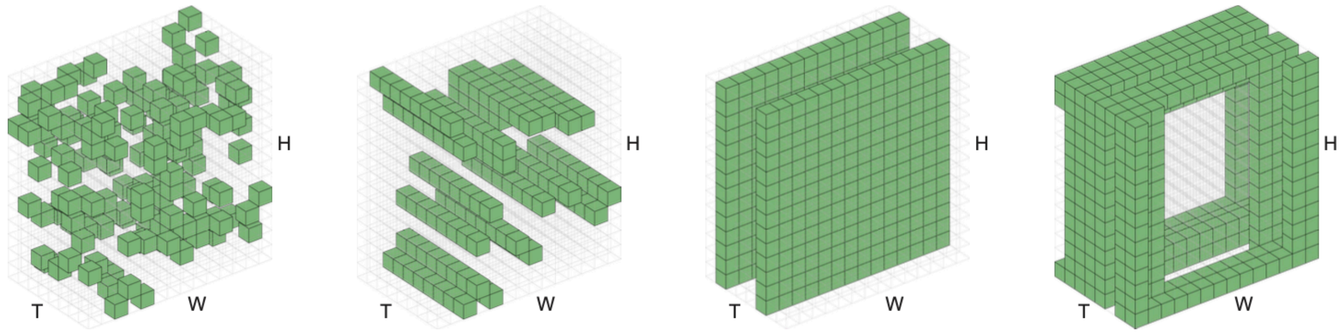
Figure 6: **Masking ratio**. Every point represents a single pre-training and fine-tuning experiment. **Left**: x-axis is the epochs (proportional to the number of *decoded* tokens). **Right**: x-axis is the number of *encoded* tokens.

04. Experiments

Ablation Experiments - Masking Strategy

case	ratio	acc.
agnostic	90	**84.4**
space-only	90	83.5
time-only	75	79.1
block	75	83.2

(a) **Mask sampling**. See also Fig. 4. Random sampling that is spacetime-*agnostic* works the best.



(a) agnostic, 90% (b) space-only, 90% (c) time-only, 75% (d) block, 75%

04. Experiments

Ablation Experiments - Reconstruction Target / Data Augmentation

case	acc.
pixel (w/o norm)	83.8
pixel (w/ norm)	**84.4**
HOG	84.0
dVAE token	83.8

(b) **Reconstruction target**. Pixels as reconstruction targets work well with no domain knowledge.

case	acc.
center crop	83.9
rand crop	**84.4**
rand crop (stronger)	83.4
rand crop + color jit	83.8

(c) **Data augmentation**. Strong augmentation is unnecessary.

04. Experiments

Ablation Experiments - Repeated Sampling / Decoder Capacity

rep.	acc.	speed
1	83.7	1.0×
2	84.3	1.8×
4	**84.4**	**3.0**×

(d) **Repeated sampling**. All entries see the same # samples. Data loading overhead is reduced.

dim	acc.
128	80.8
256	83.1
512	**84.4**
1024	83.7

(e) **Decoder width**. Unlike the image counterpart [31], an overly narrow decoder degrades accuracy noticeably.

blocks	acc.
1	83.2
2	83.6
4	**84.4**
8	84.3

(f) **Decoder depth**. Unlike the image counterpart [31], an overly shallow decoder degrades accuracy.

04. Experiments

Influence of Data

pre-train set	# pre-train data	pre-train method	K400	AVA	SSv2
-	-	none (from scratch)	71.4	-	-
IN1K	1.28M	supervised	78.6	17.8	50.2
IN1K	1.28M	MAE	82.3	27.2	65.6
K400	240k	supervised	-	22.2	55.7
K400	240k	MAE	84.8	32.3	72.1
K600	387k	MAE	**84.9**	33.7	73.0
K700	537k	MAE	n/a[†]	34.2	**73.6**
IG-uncurated	1M	MAE	84.4	**35.1**	**73.6**

Table 3: **Influence of pre-training data**, evaluated on K400, AVA, and SSv2 as the downstream tasks. The MAE pre-training length is 1600 epochs on K400/600/700 and IG-uncurated. No intermediate fine-tuning is used. The model is ViT-L. [†]: *The K700 training set has 13.9k duplicated videos with the K400 validation set (19.9k), so it is not legitimate to train on K700 to get K400 results.*

data	# videos	200-ep.	400-ep.	800-ep.
K400	240k	81.5	83.3	**84.4**
IG-curated	240k	79.0	81.6	83.2
IG-curated	512k	81.9	83.5	83.9
IG-curated	1M	**83.5**	84.1	84.2
IG-uncurated	1M	83.2	**84.5**	**84.4**

Table 4: **Real-world Instagram data** for MAE pre-training. We pre-train MAE on each individual set for 200, 400, and 800 epochs. We compare fine-tuning accuracy on K400. The model is ViT-L.