# CVPR 2022
## Review

**YeongHyeon Park**

**Department of Electrical and Computer Engineering**

**SungKyunKwan University**

# Experiences

**Good**
- Virtual session provision
- Discover other approaches (on research interests)
- Video recordings are provided (poster and oral)
- Motivation for research

**Lack**
- Errors in some sessions
- Speaker change (unannounced)
- Absence of real-time oral sessions

# Tutorial & Workshop


# Tutorial & Workshop

- Denoising Diffusion-based Generative Modeling: Foundations and Applications
- Machine Learning with Synthetic Data (SyntML) → Almost product promotion
- Beyond Convolutional Neural Networks

# Tutorial 1

## Denoising Diffusion-based Generative Modeling: Foundations and Applications

June 19, 08:30 AM - 12:10 PM (CDT)
**June 19, 10:30 PM - 02:10 AM (KST)**

Arash Vahdat

NVIDIA

**Ruiqi Gao**

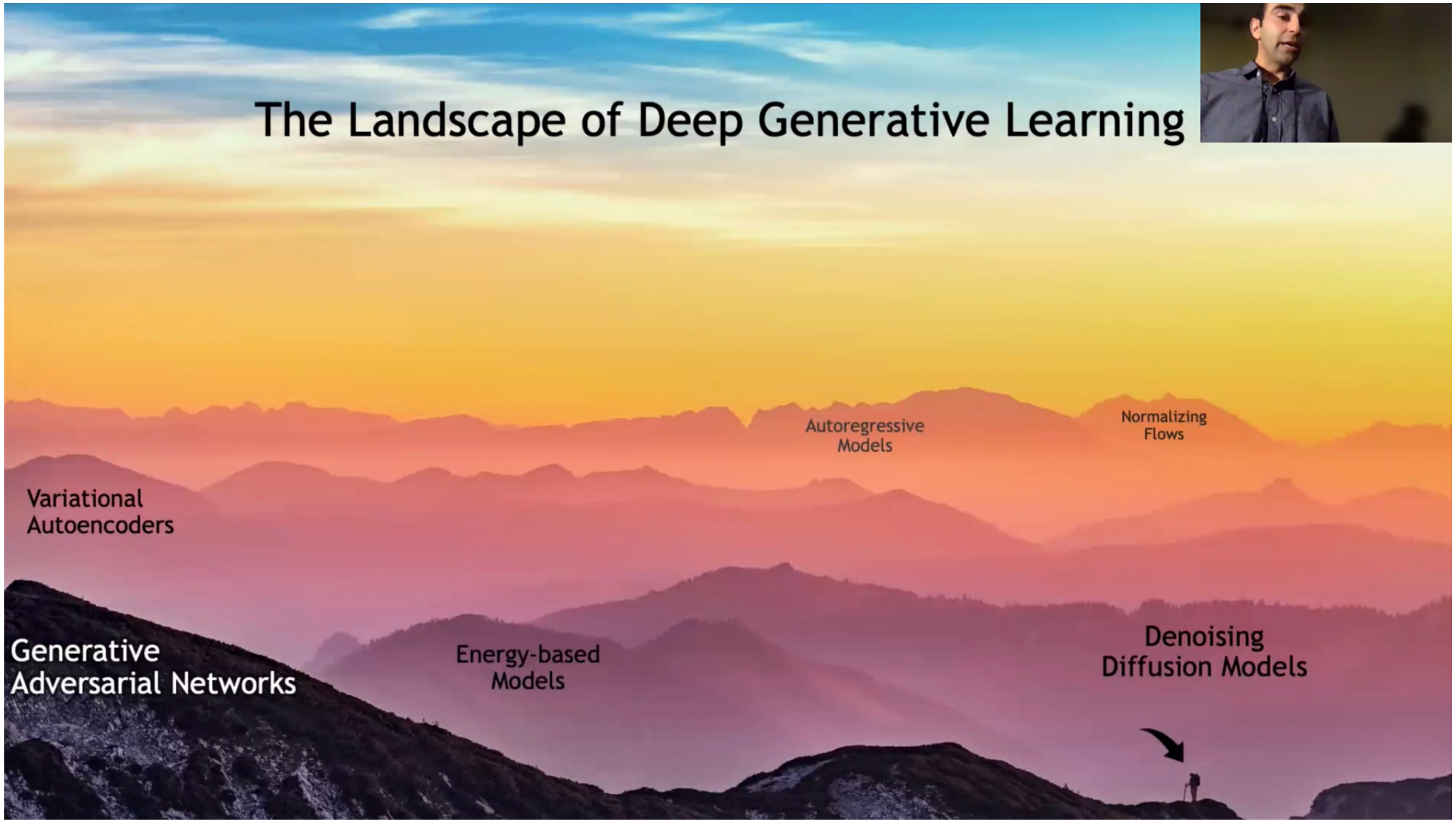
Google

**Karsten Kreis**

NVIDIA

Connection Error

# Denoising Diffusion-based Generative Modeling: Foundations and Applications



Saharia, Chitwan, et al. "Image super-resolution via iterative refinement." arXiv preprint. 2021.

# Denoising Diffusion-based Generative Modeling - Concept

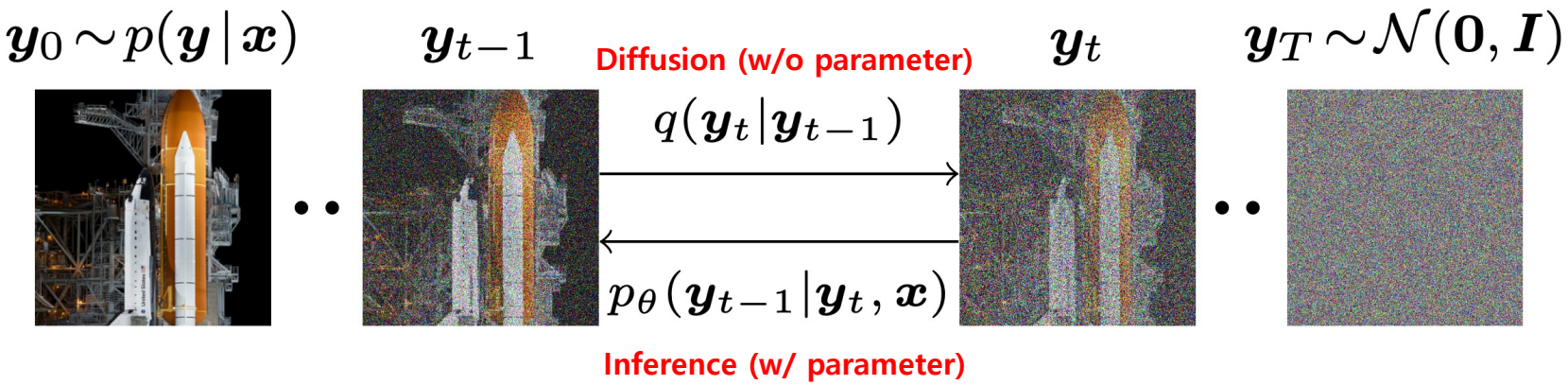



**Figure 2:** The forward diffusion process $q$ (left to right) gradually adds Gaussian noise to the target image. The reverse inference process $p$ (right to left) iteratively denoises the target image conditioned on a source image $\boldsymbol{x}$. Source image $\boldsymbol{x}$ is not shown here.

# Denoising Diffusion-based Generative Modeling - Diffusion Process

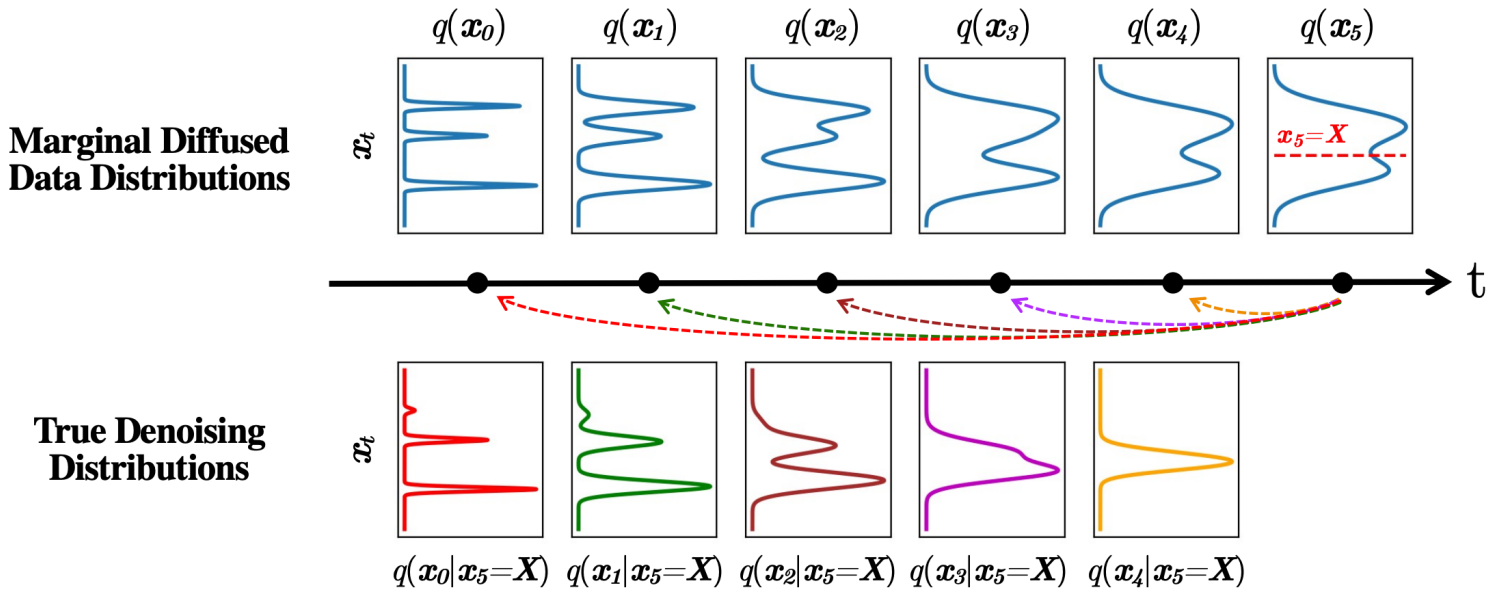



Figure 2: **Top**: The evolution of 1D data distribution $q(\mathbf{x}_0)$ through the diffusion process. **Bottom**:, The visualization of the true denoising distribution for varying step sizes conditioned on a fixed $\mathbf{x}_5$. The true denoising distribution for a small step size (i.e., $q(\mathbf{x}_4|\mathbf{x}_5 = X)$) is close to a Gaussian distribution. However, it becomes more complex and multimodal as the step size increases.

# Denoising Diffusion-based Generative Modeling - Diffusion on Image

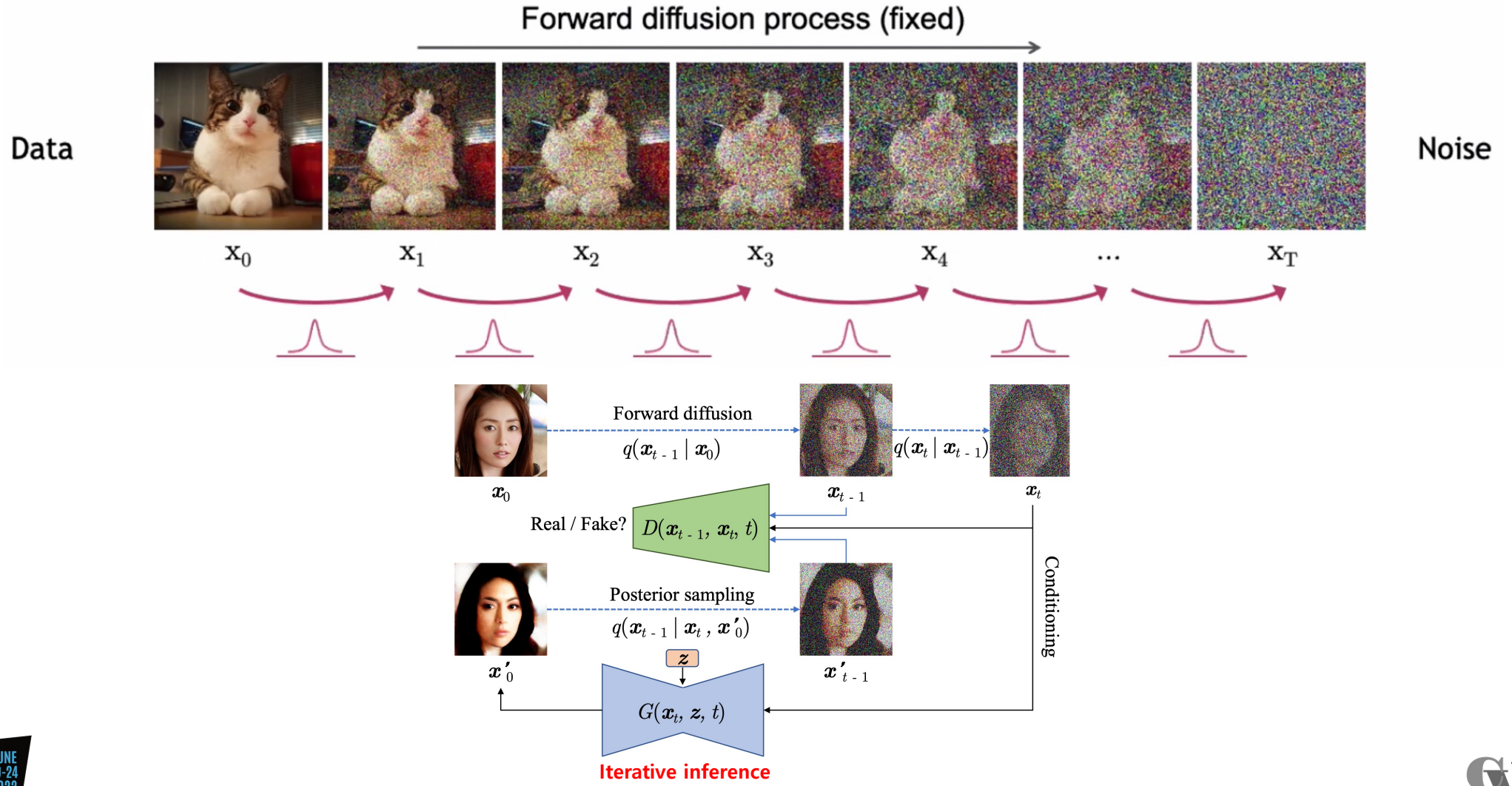



Figure 3: The training process of denoising diffusion GAN.

- Red: strong recommendation
- Blue: light recommendation

# Tutorial 2
## Summary

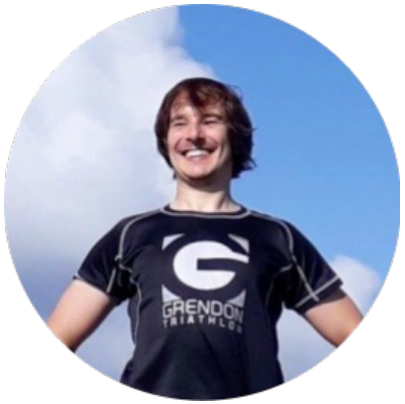
## Beyond Convolutional Neural Networks

June 20, 08:25 AM - 12:30 PM (CDT)
**June 20, 10:25 PM - 02:30 AM (KST)**

**Alexander Kolesnikov**

Google

Another perspective (w/ MLP)



**Neil Houlsby**

Google

MLP is good, Use MLP
(also recommend hybrid form)

**Alexey Dosovitskiy**

Google

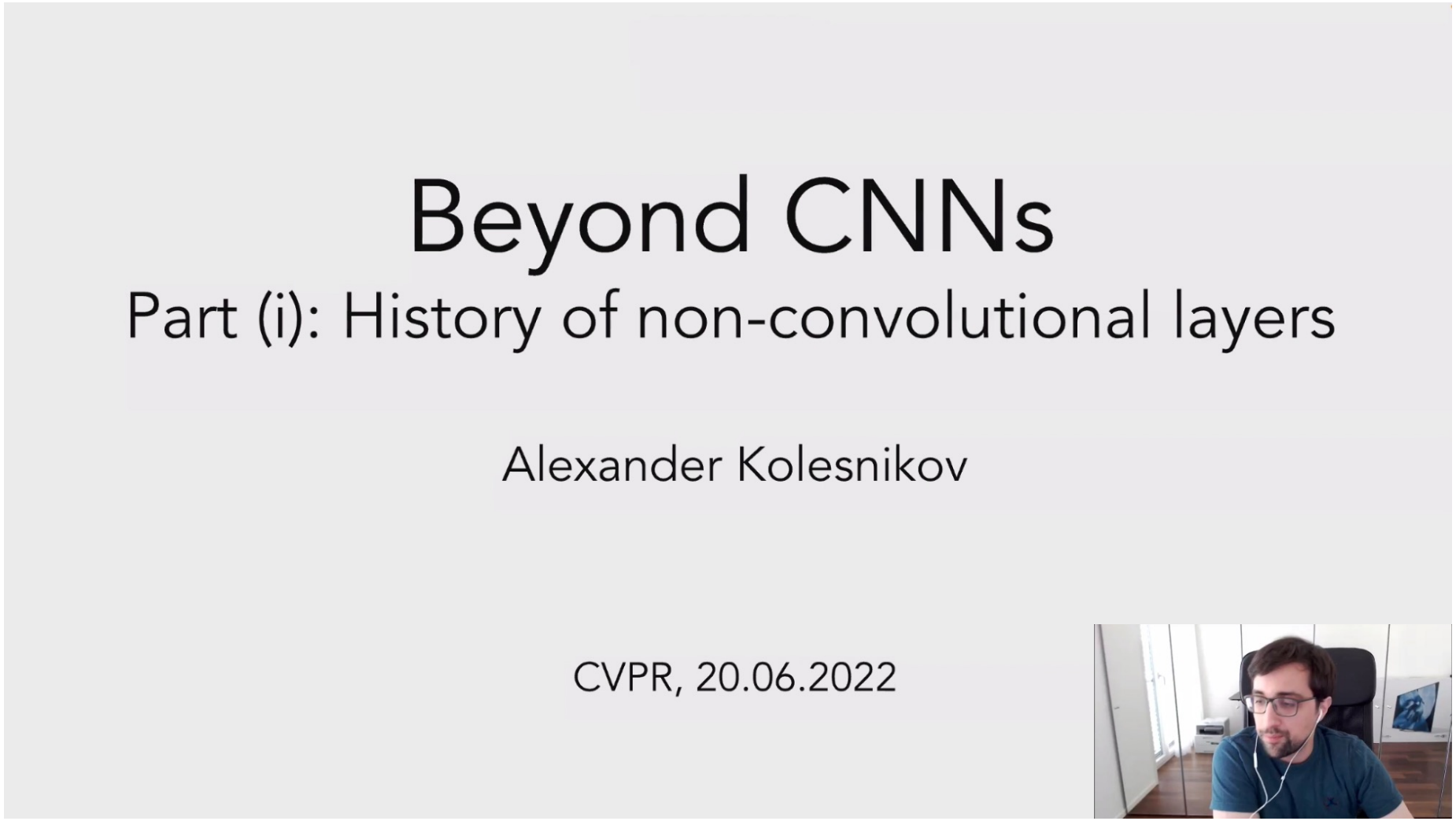
Append or mix Transformer
to CNN

**Xiaohua Zhai**

Google

Transformer can be a bridge
for multimodal data

# Beyond CNNs
## Part 1




Alexander Kolesnikov

# Beyond CNNs
## Part 1

**Local CNNs**
- LeNet
- AlexNet
- VGGNet
- ResNet
- ResNeXt
- MobileNet
- Efficientnet

**Non-Local CNNs**
- Squeeze-and-Excitation (Channel Excitation)
- Gather-Excite (Spatial Excitation)
- BAM: Bottleneck Attention Module (Channel and Spatial Excitation)

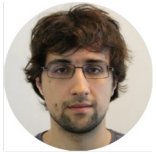
**Pros**: High computational efficiency
**Cons**: Process local-feature in general

---

**MLP Revisit (Non-Local Neural Networks : Global Processing)**
- Vision Transformer (ViT)
- MLP-Mixer
- Axial Attention (Stand-alone attention module)

**Pros**: Process global feature (global attention)
**Cons**: Low efficiency (if naïve) and easy to overfit

Tolstikhin, Ilya O., et al. "Mlp-mixer: An all-mlp architecture for vision." *Advances in Neural Information Processing Systems (Neurips)*. 2021.
Wang, Huiyu, et al. "Axial-deeplab: Stand-alone axial-attention for panoptic segmentation." *European Conference on Computer Vision (ECCV)*, 2020.



Alexander Kolesnikov

# Beyond CNNs
## Part 1

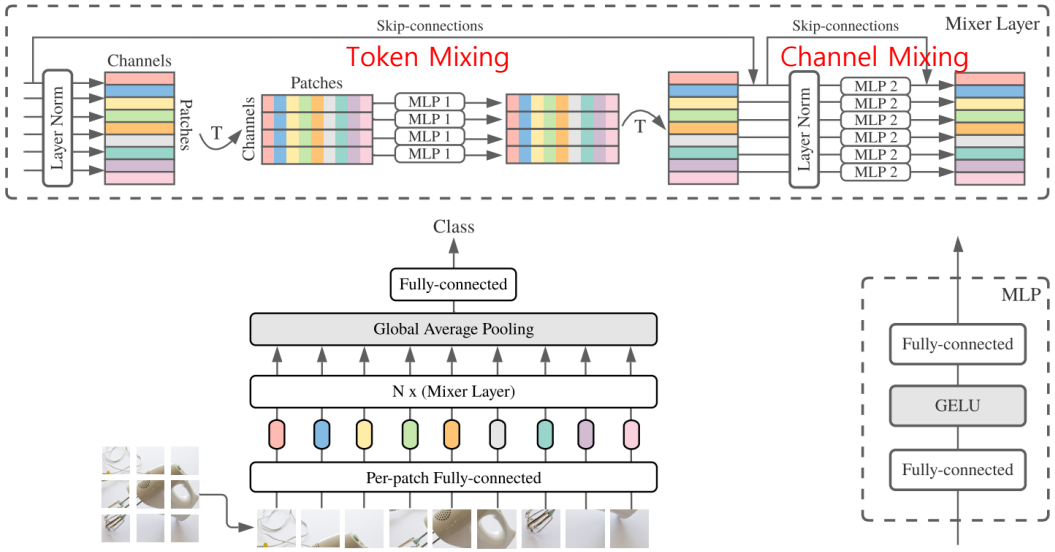



Figure 1: MLP-Mixer consists of per-patch linear embeddings, Mixer layers, and a classifier head. Mixer layers contain one token-mixing MLP and one channel-mixing MLP, each consisting of two fully-connected layers and a GELU nonlinearity. Other components include: skip-connections, dropout, and layer norm on the channels.

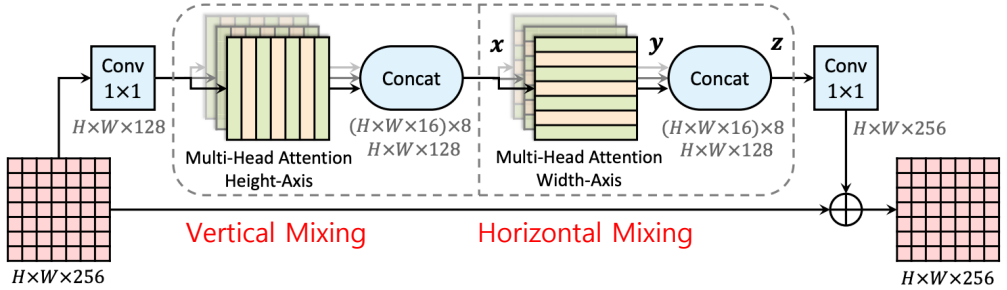
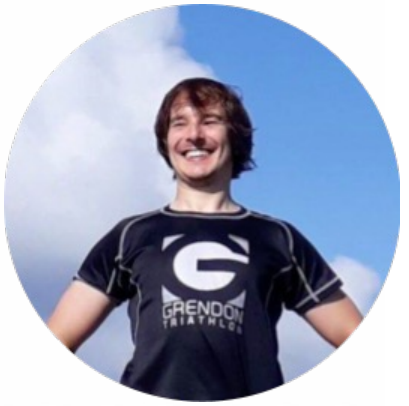



**Fig. 2.** An axial-attention block, which consists of two axial-attention layers operating along height- and width-axis sequentially. The channels $d_{in} = 128$, $d_{out} = 16$ is what we use in the first stage of ResNet after 'stem'. We employ $N = 8$ attention heads

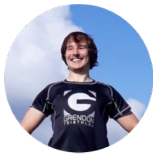
# Beyond CNNs
Part 2

## Beyond CNNs
## Part (ii): New Architecture Designs

Neil Houlsby

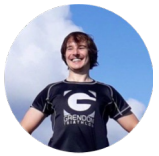


CVPR 20.06.2022


Neil Houlsby

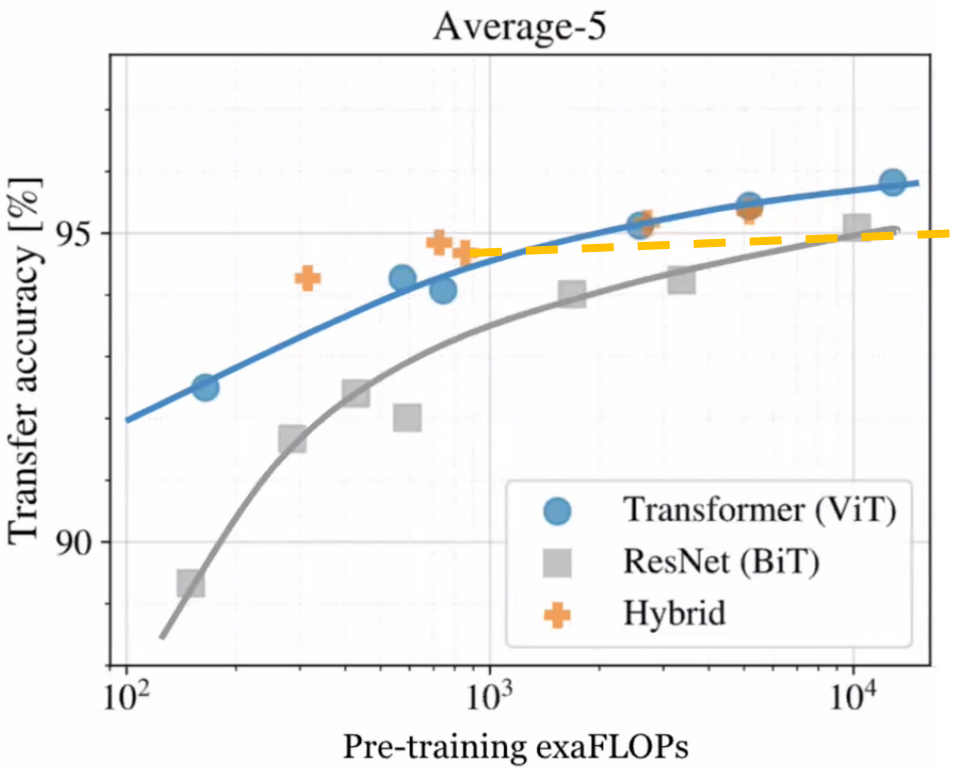
# Beyond CNNs
## Part 2

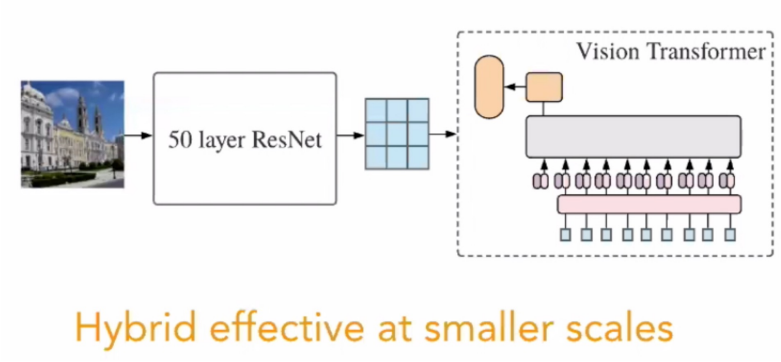
| Characteristics | Convolution | MLP |
|---|---|---|
| Shift invariance | Yes | No |
| Permutation invariance | No | Yes (at the patch level) |
| Spatial processing | Local convolution | Partial local (intra-patch)<br>Partial global (inter-patch) |
| Parameter sharing | Available | Available |
| Feature map (through depth) | Pyramid style | Non-pyramidal |



Neil Houlsby

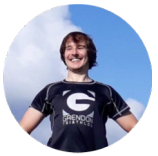
# Beyond CNNs
## Part 2




Hybrid effective at smaller scales

Tolstikhin, Ilya O., et al. "Mlp-mixer: An all-mlp architecture for vision." *Advances in Neural Information Processing Systems (Neurips)*. 2021.



Neil Houlsby

# Beyond CNNs
## Part 2

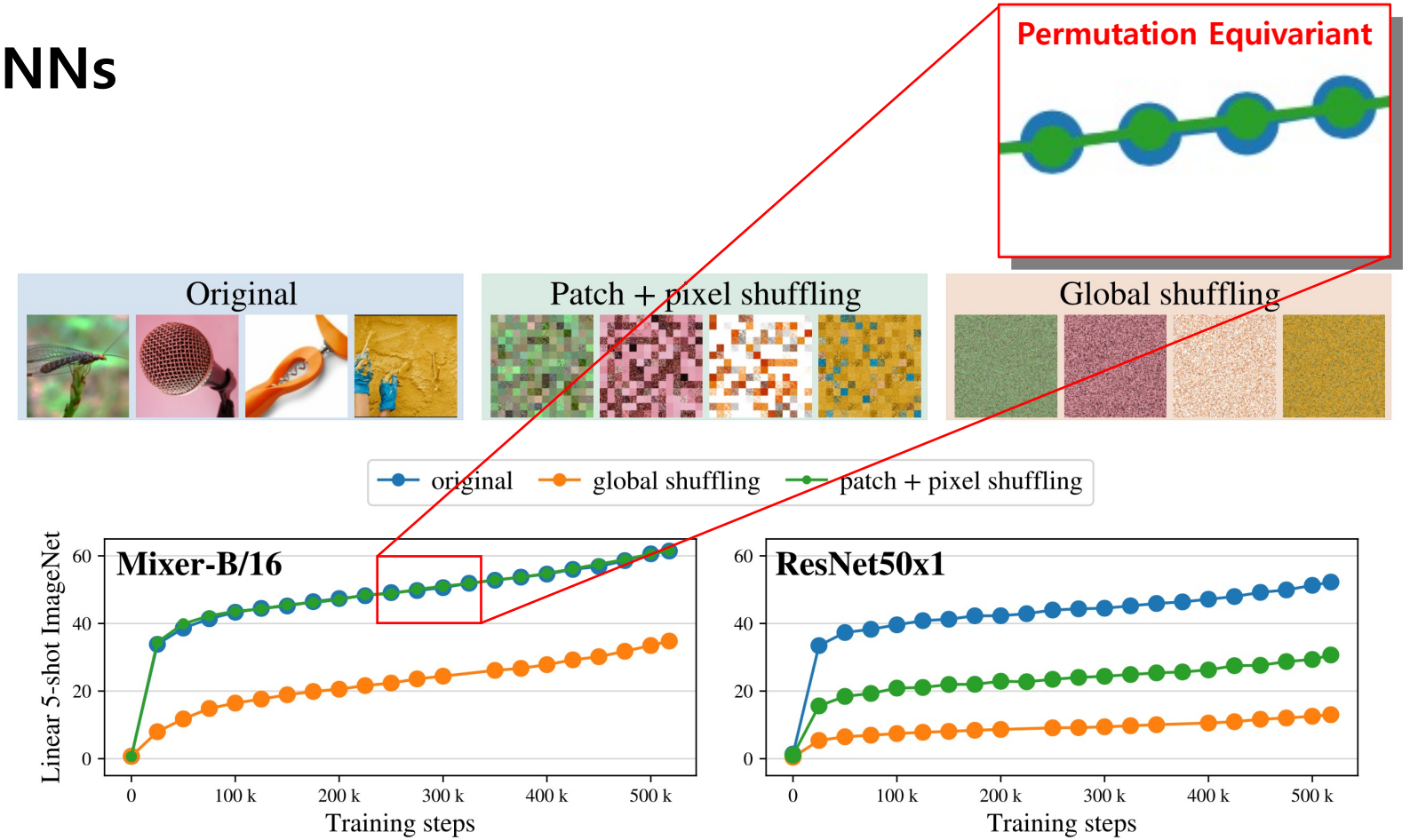
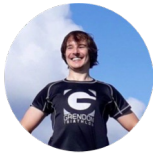



Figure 4

Touvron, Hug, et al. "Deit iii: Revenge of the vit." *arXiv preprint*. 2022.
You, Yang, et al. "Large batch optimization for deep learning: Training bert in 76 minutes." *arXiv preprint*. 2019.



Neil Houlsby

# Beyond CNNs
## Part 2

**LAMB:** Layer-wise **A**daptive **M**oments optimizer for **B**atch training

Table 1: Summary of our training procedures with ImageNet-1k and ImageNet-21k. We also provide DeiT [48], Wightman et al [57] and Steiner et al. [42] baselines for reference. Adapt. means the hparams is adapted to the size of the model. For finetuning to higher resolution with model pre-trained on ImageNet-1k only we use the finetuning procedure from DeiT see section A for more details.

| | Previous approaches | | | | Ours | | |
|---|---|---|---|---|---|---|---|
| Procedure → | ViT | Steiner | DeiT | Wightman | ImNet-1k | ImNet-21k | |
| Reference | [13] | et al. [42] | [48] | et al. [57] | | Pretrain. | Finetune. |
| Batch size | 4096 | 4096 | 1024 | 2048 | 2048 | 2048 | 2048 |
| Optimizer | AdamW | AdamW | AdamW | LAMB | LAMB | LAMB | LAMB |
| LR | $3.10^{-3}$ | $3.10^{-3}$ | $1.10^{-3}$ | $5.10^{-3}$ | $3.10^{-3}$ | $3.10^{-3}$ | $3.10^{-4}$ |
| LR decay | cosine | cosine | cosine | cosine | cosine | cosine | cosine |
| Weight decay | 0.1 | 0.3 | 0.05 | 0.02 | 0.02 | 0.02 | 0.02 |
| Warmup epochs | 3.4 | 3.4 | 5 | 5 | 5 | 5 | 5 |
| Label smoothing $\varepsilon$ | 0.1 | 0.1 | 0.1 | ✗ | ✗ | 0.1 | 0.1 |
| Dropout | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| Stoch. Depth | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Repeated Aug | ✗ | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ |
| Gradient Clip. | 1.0 | 1.0 | ✗ | 1.0 | 1.0 | 1.0 | 1.0 |
| H. flip | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RRC | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ |
| Rand Augment | ✗ | Adapt. | 9/0.5 | 7/0.5 | ✗ | ✗ | ✗ |
| 3 Augment (ours) | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ |
| LayerScale | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ |
| Mixup alpha | ✗ | Adapt. | 0.8 | 0.2 | 0.8 | ✗ | ✗ |
| Cutmix alpha | ✗ | ✗ | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| Erasing prob. | ✗ | ✗ | 0.25 | ✗ | ✗ | ✗ | ✗ |
| ColorJitter | ✗ | ✗ | ✗ | ✗ | 0.3 | 0.3 | 0.3 |
| Test crop ratio | 0.875 | 0.875 | 0.875 | 0.95 | 1.0 | 1.0 | 1.0 |
| Loss | CE | CE | CE | BCE | BCE | CE | CE |

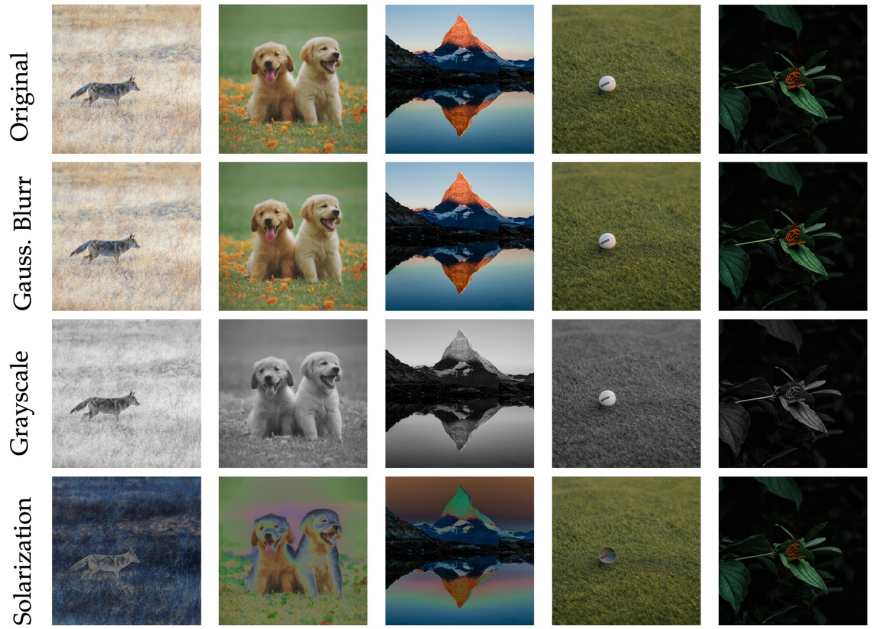
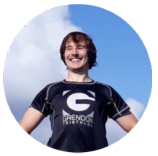



Figure 2: Illustration of the 3 types of data-augmentations used in 3-Augment.

Zhai, Xiaohua, et al. "Scaling vision transformers." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022.



Neil Houlsby

# Beyond CNNs
## Part 2

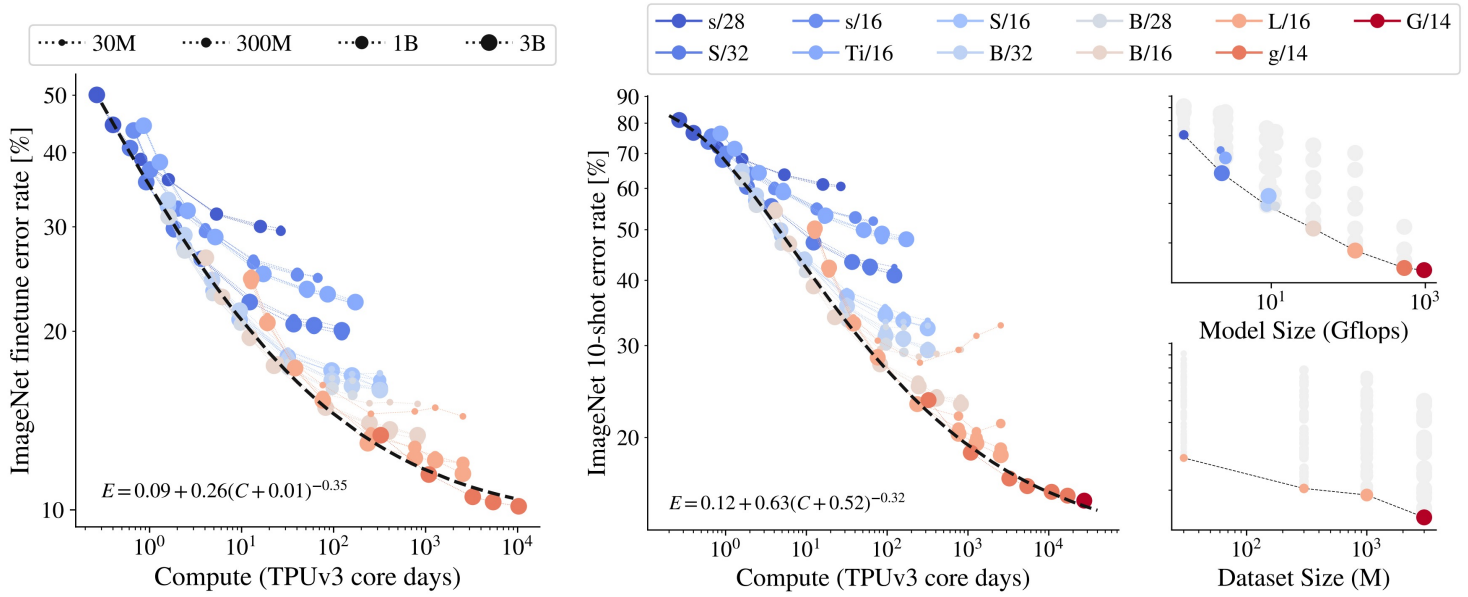



Figure 2. **Left/Center**: Representation quality, measured as ImageNet finetune and linear 10-shot error rate, as a function of total training compute. A saturating power-law approximates the Pareto frontier fairly accurately. Note that smaller models (blue shading), or models trained on fewer images (smaller markers), saturate and fall off the frontier when trained for longer. **Top right**: Representation quality when bottlenecked by model size. For each model size, a large dataset and amount of compute is used, so model capacity is the main bottleneck. Faintly-shaded markers depict sub-optimal runs of each model. **Bottom Right**: Representation quality by datasets size. For each dataset size, the model with an optimal size and amount of compute is highlighted, so dataset size is the main bottleneck.

**Use MLP** if you have
- Huge dataset
- Huge model (w/ sufficient equipment)

**It not**, the hybrid (MLP-CNN) architecture is recommended.

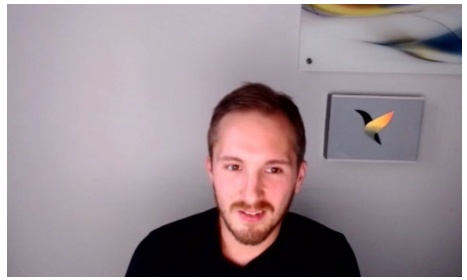

# Beyond CNNs
Part 3

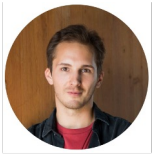
# Beyond CNNs

## Part (iii): Beyond Image Classification

Alexey Dosovitskiy

CVPR, New Orleans, 20.06.2022





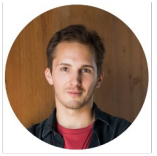
Alexey Dosovitskiy

# Beyond CNNs
## Part 3

**End-to-end instance-level tasks with transformers**

- Backbones for object detection / tracking
- Video processing
- 3D processing (mesh generation or etc.)

Carion, Nicolas, et al. "End-to-end object detection with transformers." *European conference on computer vision (ECCV)*. 2020.



Alexey Dosovitskiy

# Beyond CNNs
## Part 3

DETR: DEtection TRansformer

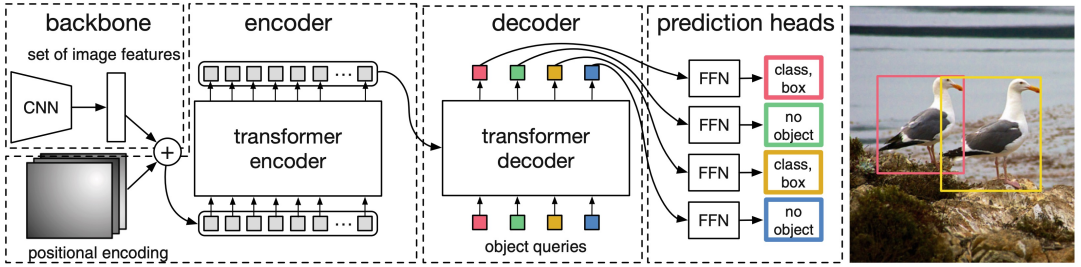



**Fig. 2.** DETR uses a conventional CNN backbone to learn a 2D representation of an input image. The model flattens it and supplements it with a positional encoding before passing it into a transformer encoder. A transformer decoder then takes as input a small fixed number of learned positional embeddings, which we call *object queries*, and additionally attends to the encoder output. We pass each output embedding of the decoder to a shared feed forward network (FFN) that predicts either a detection (class and bounding box) or a "`no object`" class.

**Table 1.** Comparison with RetinaNet and Faster R-CNN with a ResNet-50 and ResNet-101 backbones on the COCO validation set. The top section shows results for models in Detectron2 [49], the middle section shows results for models with GIoU [37], random crops train-time augmentation, and the long **9x** training schedule. DETR models achieve comparable results to heavily tuned Faster R-CNN baselines, having lower $AP_S$ but greatly improved $AP_L$. We use torchscript models to measure FLOPS and FPS. Results without R101 in the name correspond to ResNet-50.

| Model | GFLOPS/FPS | #params | AP | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ |
|---|---|---|---|---|---|---|---|---|
| RetinaNet | 205/18 | 38M | 38.7 | 58.0 | 41.5 | 23.3 | 42.3 | 50.3 |
| Faster RCNN-DC5 | 320/16 | 166M | 39.0 | 60.5 | 42.3 | 21.4 | 43.5 | 52.5 |
| Faster RCNN-FPN | 180/26 | 42M | 40.2 | 61.0 | 43.8 | 24.2 | 43.5 | 52.0 |
| Faster RCNN-R101-FPN | 246/20 | 60M | 42.0 | 62.5 | 45.9 | 25.2 | 45.6 | 54.6 |
| RetinaNet+ | 205/18 | 38M | 41.1 | 60.4 | 43.7 | 25.6 | 44.8 | 53.6 |
| Faster RCNN-DC5+ | 320/16 | 166M | 41.1 | 61.4 | 44.3 | 22.9 | 45.9 | 55.0 |
| Faster RCNN-FPN+ | 180/26 | 42M | 42.0 | 62.1 | 45.5 | 26.6 | 45.4 | 53.4 |
| Faster RCNN-R101-FPN+ | 246/20 | 60M | 44.0 | 63.9 | **47.8** | **27.2** | 48.1 | 56.0 |
| DETR | 86/28 | 41M | 42.0 | 62.4 | 44.2 | 20.5 | 45.8 | 61.1 |
| DETR-DC5 | 187/12 | 41M | 43.3 | 63.1 | 45.9 | 22.5 | 47.3 | 61.1 |
| DETR-R101 | 152/20 | 60M | 43.5 | 63.8 | 46.4 | 21.9 | 48.0 | 61.8 |
| DETR-DC5-R101 | 253/10 | 60M | **44.9** | **64.7** | 47.7 | 23.7 | **49.5** | **62.3** |

Zhang, Hao, et al. "Dino: Detr with improved denoising anchor boxes for end-to-end object detection." *arXiv preprint*. 2022.

Alexey Dosovitskiy

# Beyond CNNs
## Part 3

**DINO**: **D**ETR with **I**mproved de**N**oising anch**O**r boxes

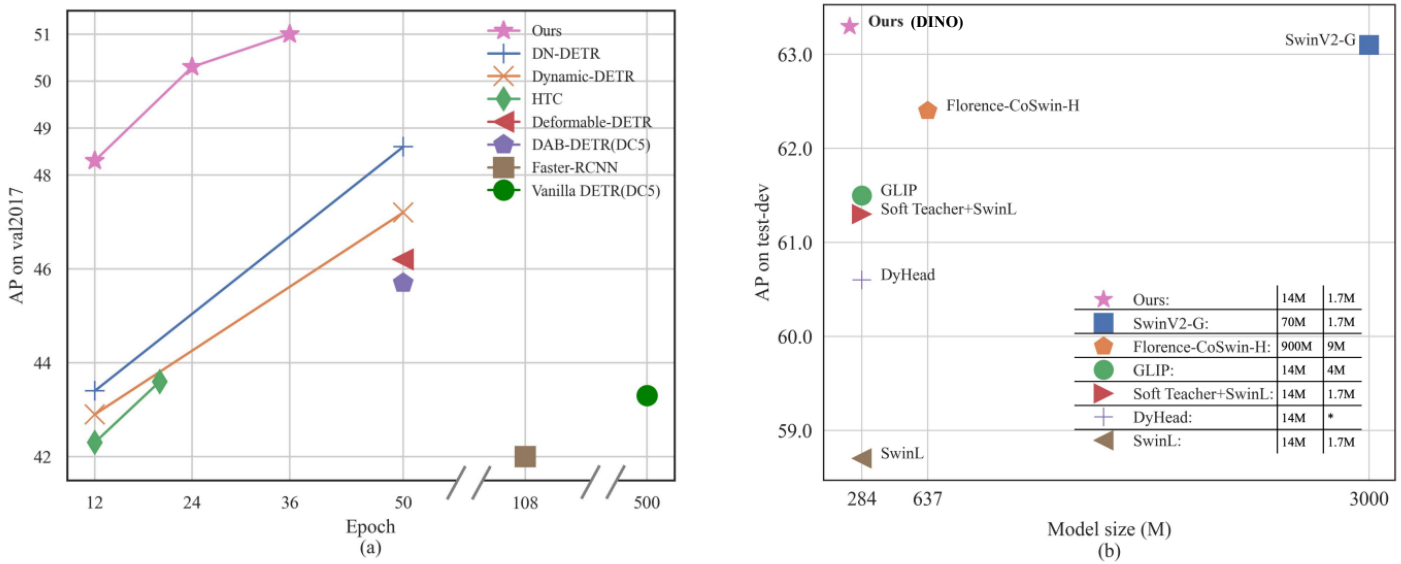
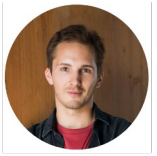



**Fig. 1.** AP on COCO compared with other detection models. (a) Comparison to models with a ResNet-50 backbone w.r.t. training epochs. Models marked with DC5 use a dilated larger resolution feature map. Other models use multi-scale features. (b) Comparison to SOTA models w.r.t. pre-training data size and model size. SOTA models are from the COCO `test-dev` leaderboard. In the legend we list the backbone pre-training data size (first number) and detection pre-training data size (second number). $*$ means the data size is not disclosed.

Meinhardt, Tim, et al. "Trackformer: Multi-object tracking with transformers." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022.


Alexey Dosovitskiy

# Beyond CNNs
## Part 3

**CNN:** Feature extractor
**RNN:** Object tracking
**Transformer:** Object detection

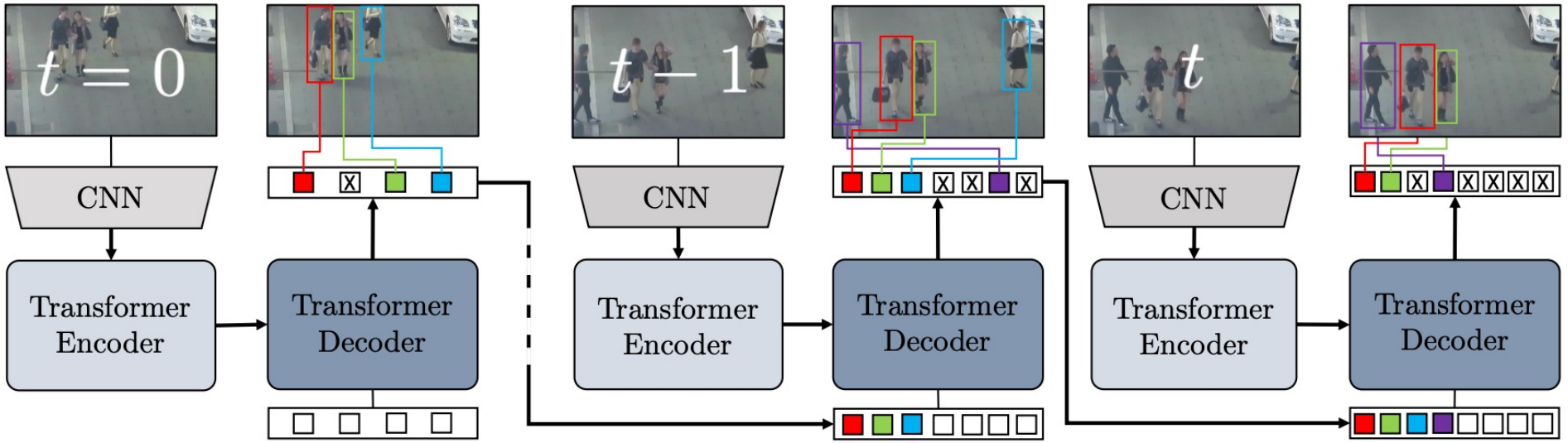
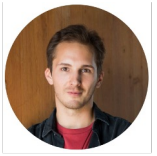



Figure 2. **TrackFormer** casts multi-object tracking as a set prediction problem performing joint detection and **tracking-by-attention**. The architecture consists of a CNN for image feature extraction, a Transformer [50] encoder for image feature encoding and a Transformer decoder which applies self- and encoder-decoder attention to produce output embeddings with bounding box and class information. At frame $t = 0$, the decoder transforms $N_{\text{object}}$ object queries (white) to output embeddings either initializing new autoregressive **track queries** or predicting the background class (crossed). On subsequent frames, the decoder processes the joint set of $N_{\text{object}} + N_{\text{track}}$ queries to follow or remove (blue) existing tracks as well as initialize new tracks (purple).

Lin, Kevin, et al. "End-to-end human pose and mesh reconstruction with transformers." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021.



Alexey Dosovitskiy

# Beyond CNNs
## Part 3

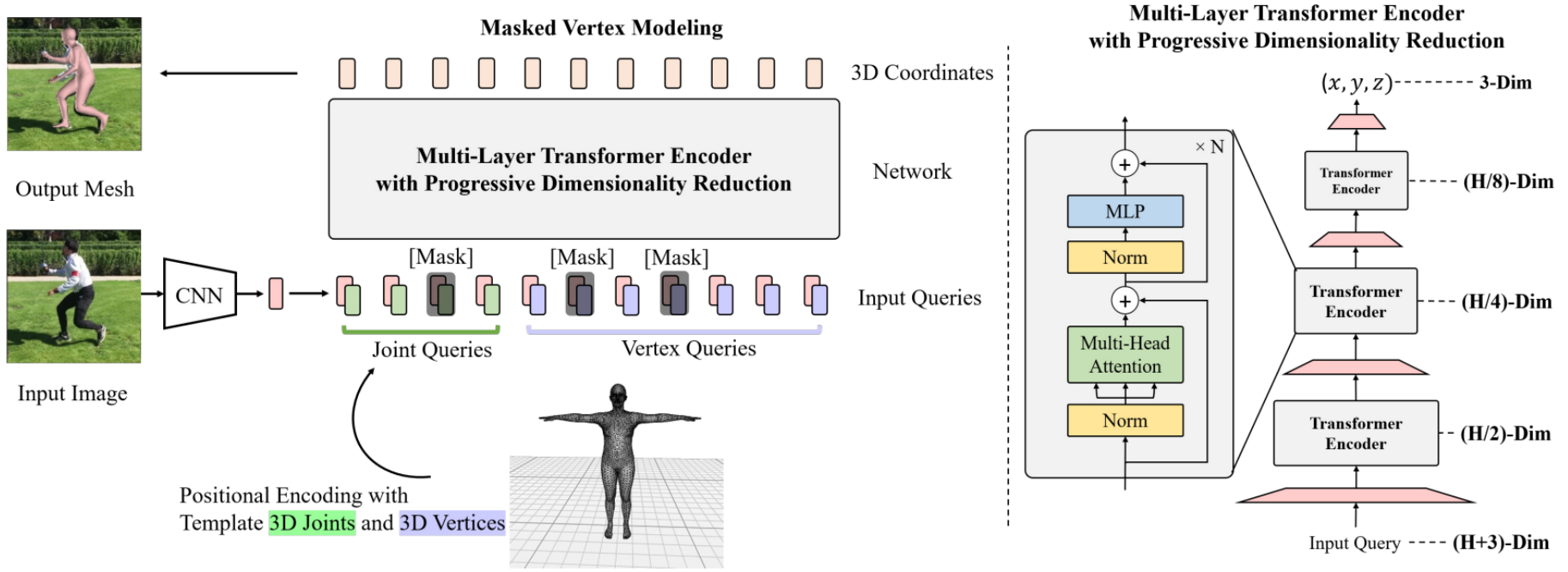



Figure 2: **Overview of the proposed framework.** Given an input image, we extract an image feature vector using a convolutional neural network (CNN). We perform position encoding by adding a template human mesh to the image feature vector by concatenating the image feature with the 3D coordinates $(x_i, y_i, z_i)$ of every body joint $i$, and 3D coordinates $(x_j, y_j, z_j)$ of every vertex $j$. Given a set of joint queries and vertex queries, we perform self-attentions through multiple layers of a transformer encoder, and regress the 3D coordinates of body joints and mesh vertices in parallel. We use a progressive dimensionality reduction architecture (right) to gradually reduce the hidden embedding dimensions from layer to layer. Each token in the final layer outputs 3D coordinates of a joint or mesh vertex. Each encoder block has 4 layers and 4 attention heads. $H$ denotes the dimension of an image feature vector.


# Beyond CNNs
Part 4

## Beyond CNNs
### Part (iv): Multimodal and Self-supervised Learning

Xiaohua Zhai

CVPR, 20.06.2022

Xiaohua Zhai

# Beyond CNNs

## Part 4

- Multimodal classification
- Self-supervised Learning

Radford, Alec, et al. "Learning transferable visual models from natural language supervision." *International Conference on Machine Learning (ICML).* 2021.

Xiaohua Zhai

# Beyond CNNs
## Part 4

**CLIP**: **C**ontrastive **L**anguage-**I**mage **P**re-Training

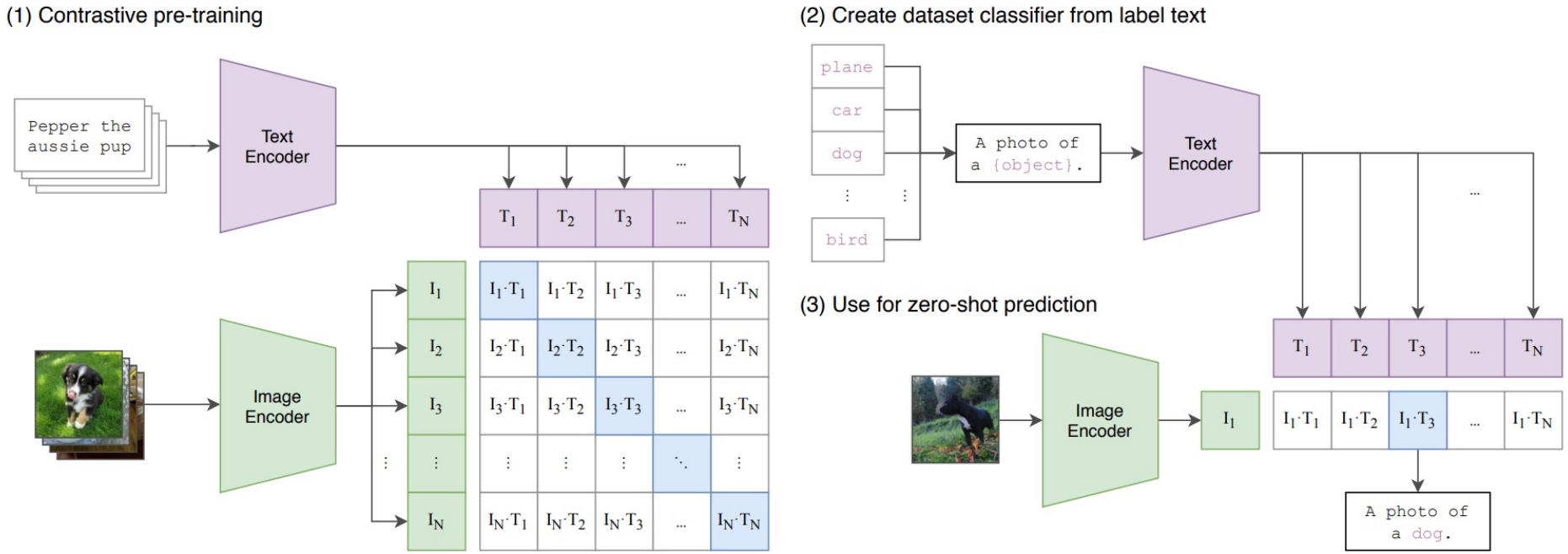



*Figure 1.* Summary of our approach. While standard image models jointly train an image feature extractor and a linear classifier to predict some label, CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples. At test time the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset's classes.

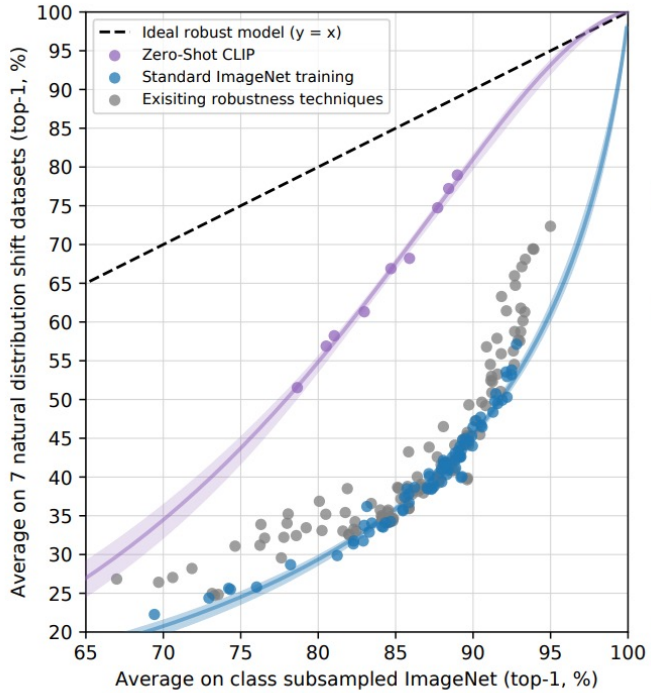
Public: CC12M YFCC100m dataset

Radford, Alec, et al. "Learning transferable visual models from natural language supervision." *International Conference on Machine Learning (ICML).* 2021.

Xiaohua Zhai

# Beyond CNNs
## Part 4

**CLIP**: **C**ontrastive **L**anguage-**I**mage **P**re-Training



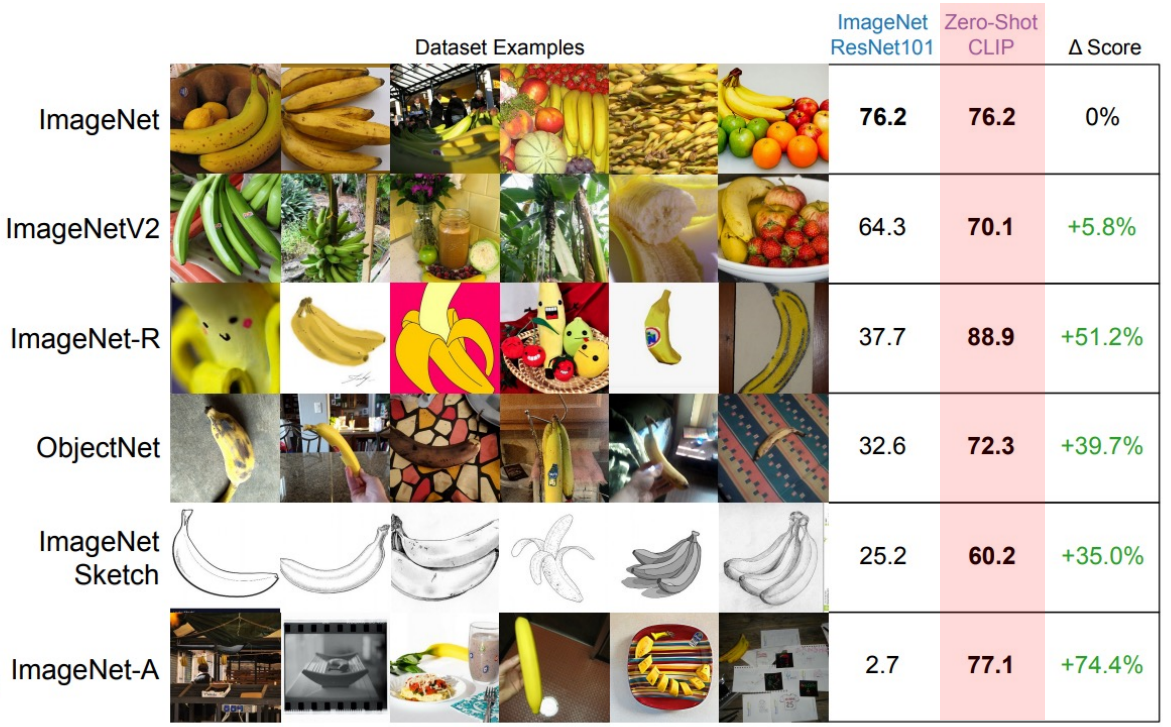




*Figure 7.* **Zero-shot CLIP is much more robust to distribution shift than standard ImageNet models.** (Left) An ideal robust model (dashed line) performs equally well on the ImageNet distribution and on other natural image distributions. Zero-shot CLIP models shrink this "robustness gap" by up to 75%. Linear fits on logit transformed values are shown with bootstrap estimated 95% confidence intervals. (Right) Visualizing distribution shift for bananas, a class shared across 5 of the 7 natural distribution shift datasets. The performance of the best zero-shot CLIP model is compared with a model that has the same performance on the ImageNet validation set, ResNet101.

CVPR JUNE 19-24 2022 NEW ORLEANS • LOUISIANA

GVL SUNG KYUN KWAN UNIVERSITY 1398 성균관대학교

Radford, Alec, et al. "Learning transferable visual models from natural language supervision." *International Conference on Machine Learning (ICML).* 2021.

Xiaohua Zhai

# Beyond CNNs

## Part 4

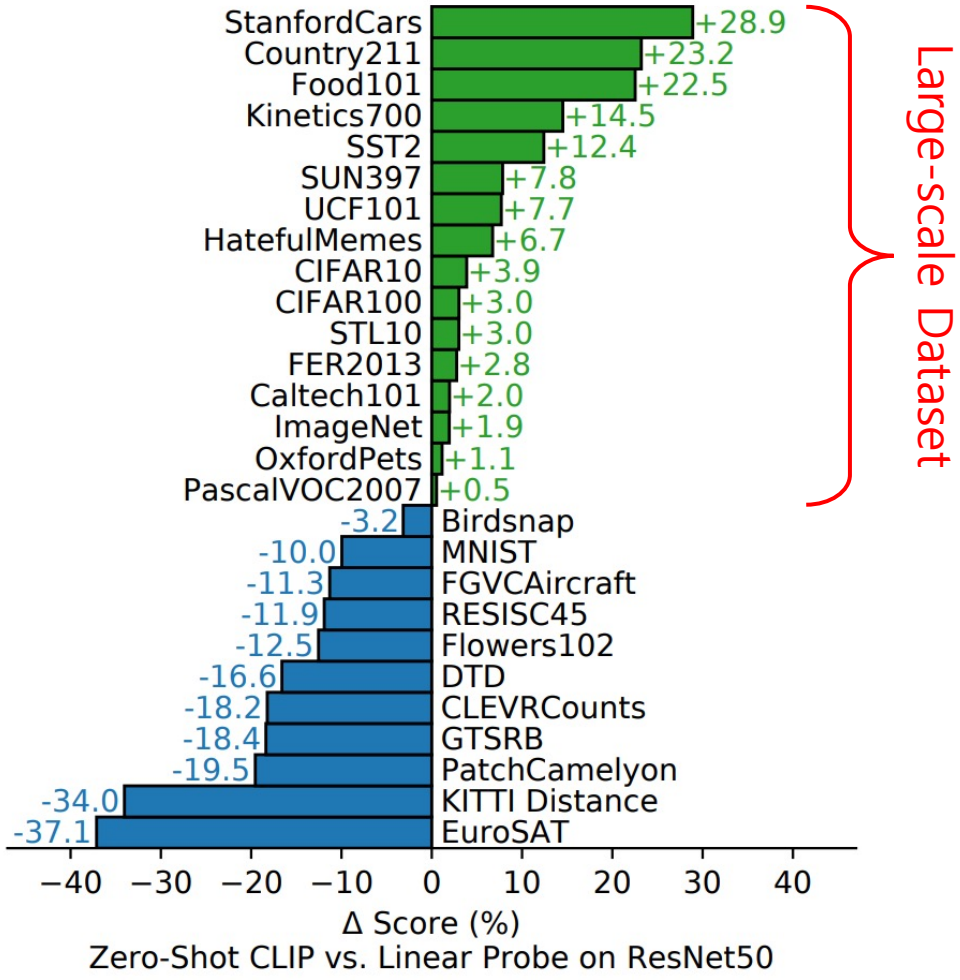



*Figure 4.* **Zero-shot CLIP is competitive with a fully supervised baseline.** Across a 27 dataset eval suite, a zero-shot CLIP classifier outperforms a fully supervised linear classifier fitted on ResNet50 features on 16 datasets, including ImageNet.

Zhai, Xiaohua, et al. "Lit: Zero-shot transfer with locked-image text tuning." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022.

Xiaohua Zhai

# Beyond CNNs

## Part 4

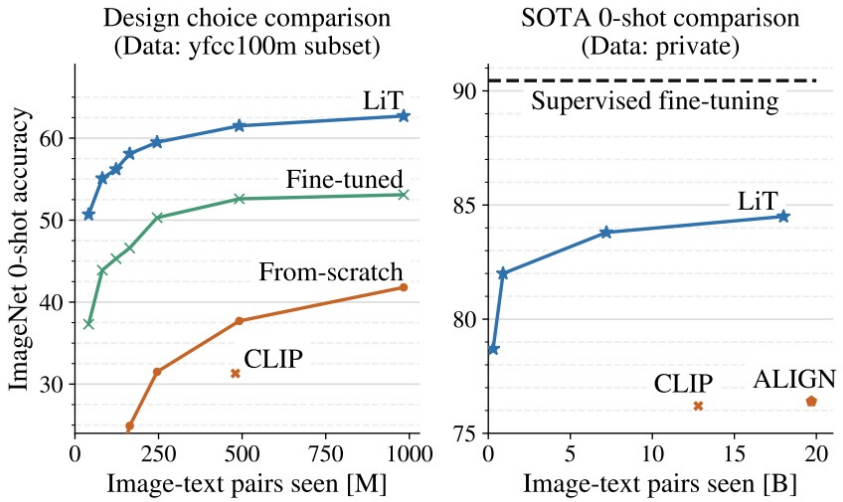



Figure 1. Comparison to the previous SOTA methods. **Left**: results on public YFCC100m subset, with from-scratch, fine-tuned from a pre-trained image model, and LiT with a pre-trained image model. The proposed LiT improves over 30% ImageNet zero-shot transfer accuracy on YFCC100m subset. **Right**: results on privately gathered data, LiT halves the gap between previous from-scratch methods CLIP [45], ALIGN [30] and supervised fine-tuning [12, 68].

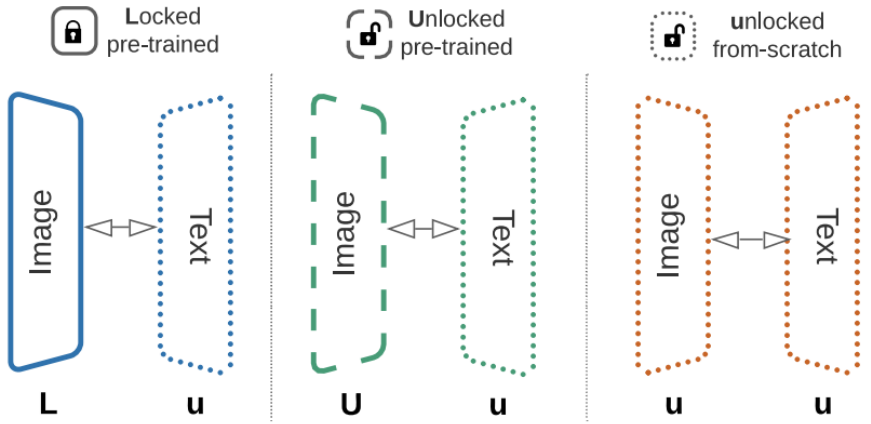



Figure 2. Design choices for contrastive-tuning on image-text data. Two letters are introduced to represent the image tower and text tower setups. `L` stands for locked variables and initialized from a pre-trained model, `U` stands for unlocked and initialized from a pre-trained model, `u` stands for unlocked and randomly initialized. `Lu` is named as "Locked-image Tuning" (LiT).

# Orals & Posters

Oral presentation

Good to understand

# Orals & Posters

**Anomaly Detection**

- Rethinking Reconstruction Autoencoder-Based Out-of-Distribution Detection
- Catching Both Gray and Black Swans: Open-Set Supervised Anomaly Detection
- Anomaly Detection via Reverse Distillation From One-Class Embedding
- Towards Total Recall in Industrial Anomaly Detection

**Cross Domain Problems**

- OoD-Bench: Quantifying and Understanding Two Dimensions of Out-of-Distribution Generalization (no video)
- The Two Dimensions of Worst-Case Training and Their Integrated Effect for Out-of-Domain Generalization

**Others**

- Self-Supervised Predictive Convolutional Attentive Block for Anomaly Detection

CVPR JUNE 19-24 2022 NEW ORLEANS • LOUISIANA

GVL SUNG KYUN KWAN UNIVERSITY 1398 성균관대학교

Zhou, Yibo. "Rethinking Reconstruction Autoencoder-Based Out-of-Distribution Detection." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022.

# Rethinking Reconstruction Autoencoder-Based Out-of-Distribution Detection

**Rethinking Reconstruction Autoencoder-Based Out-of-Distribution Detection**

Yibo Zhou
Beihang University
ybzhou@impcas.ac.cn

Zhou, Yibo. "Rethinking Reconstruction Autoencoder-Based Out-of-Distribution Detection." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022.

# Rethinking Reconstruction Autoencoder-Based Out-of-Distribution Detection

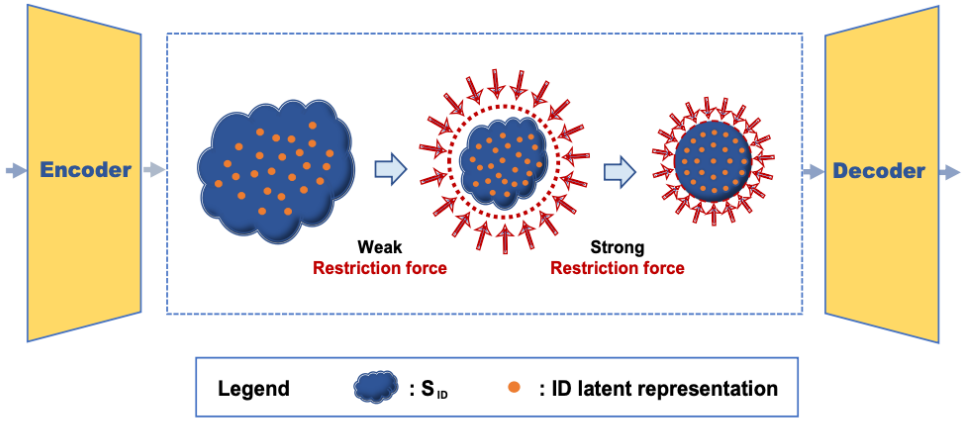



Figure 2. Illustration of the transition of $\mathcal{S}_{ID}$ when the restrictive power imposed over latent codes increases. During training, any deviation of the latent codes from the latent space (red-dot circle) would be penalized greatly. By tightening this space sufficiently, in principle it would be mostly utilized to satisfy the jointly learned reconstruction task. Thus, detecting the outlier of $\mathcal{S}_{ID}$ is approximately equal to identifying the feature outside this latent space.

- $NL2$: **N**ormalized **L2** distance
- $f$: input
- $\tilde{f}$: reconstruction of $f$

$$Dist(f, \tilde{f}) = NL2(f, \tilde{f}) = \| \frac{f}{\|f\|} - \frac{\tilde{f}}{\|f\|} \|, \qquad (6)$$

- For projection the $f$ and $\tilde{f}$ on the same surface.

**Good**: Performance improvement with simple loss transformation

**Bad**: Other loss functions such as L1 and SSIM are not handled

# Catching Both Gray and Black Swans: Open-set Supervised Anomaly Detection

**Catching Both Gray and Black Swans: Open-set Supervised Anomaly Detection***

Choubo Ding[1†], Guansong Pang[2†], Chunhua Shen[3]
[1] The University of Adelaide [2] Singapore Management University [3] Zhejiang University

**DRA**: **D**isentangled **R**epresentations of **A**bnormalities

Ding, Choubo, Guansong Pang, and Chunhua Shen. "Catching Both Gray and Black Swans: Open-set Supervised Anomaly Detection." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022.
Yun, Sangdoo, et al. "Cutmix: Regularization strategy to train strong classifiers with localizable features." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

# Catching Both Gray and Black Swans: Open-set Supervised Anomaly Detection

**DRA**: **D**isentangled **R**epresentations of **A**bnormalities

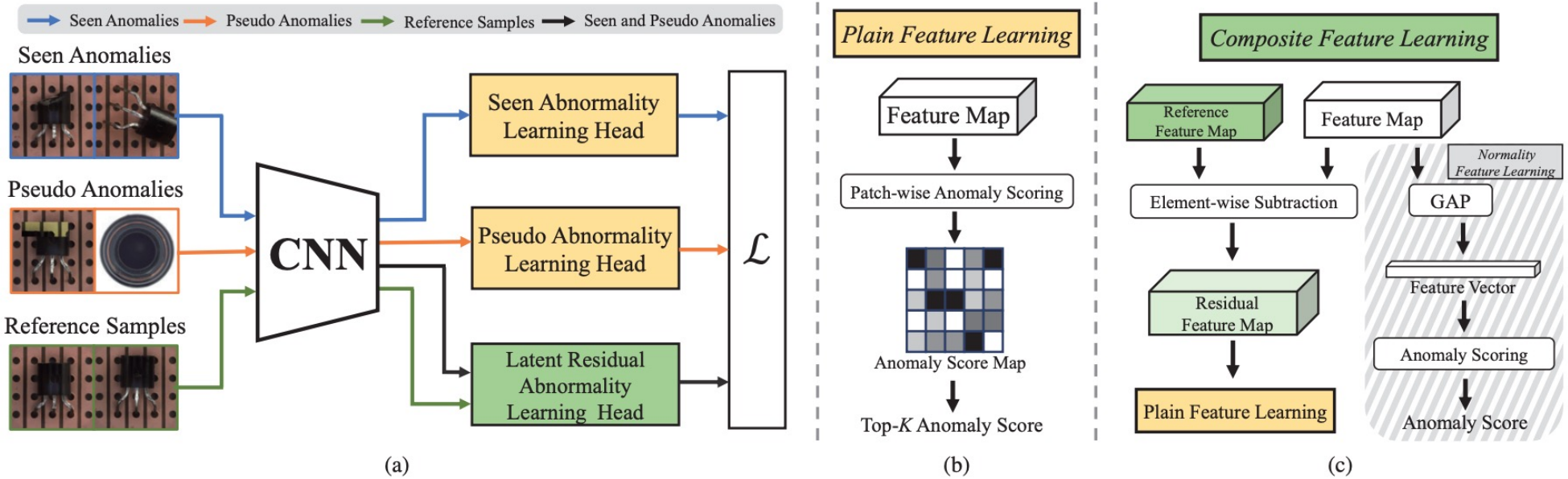
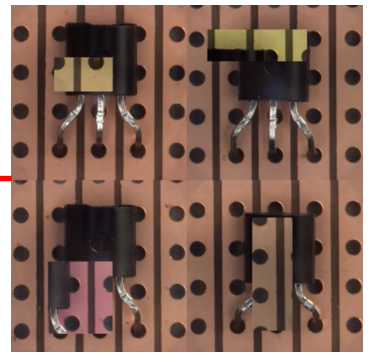



Figure 2. Overview of our proposed framework. (a) presents the high-level procedure of learning three disentangled abnormalities, (b) shows the abnormality feature learning in the plain (non-composite) feature space for the seen and pseudo abnormality learning heads, and (c) shows the framework of our proposed latent residual abnormality learning in a composite feature space.



CutMix

Tang, Youbao, et al. "A disentangled generative model for disease decomposition in chest x-rays via normal image synthesis." *Medical Image Analysis,* Elsevier. 2021. (16 citations)

# DGM
## Disentangled Generative Model

**Dataset**
- Chest Xray 8 (Public)

① Generate normal sample $y'$
② Restore input data $z''$
by combining latent information of $y'$ and $a$
③ Suppress abnormal information $a$

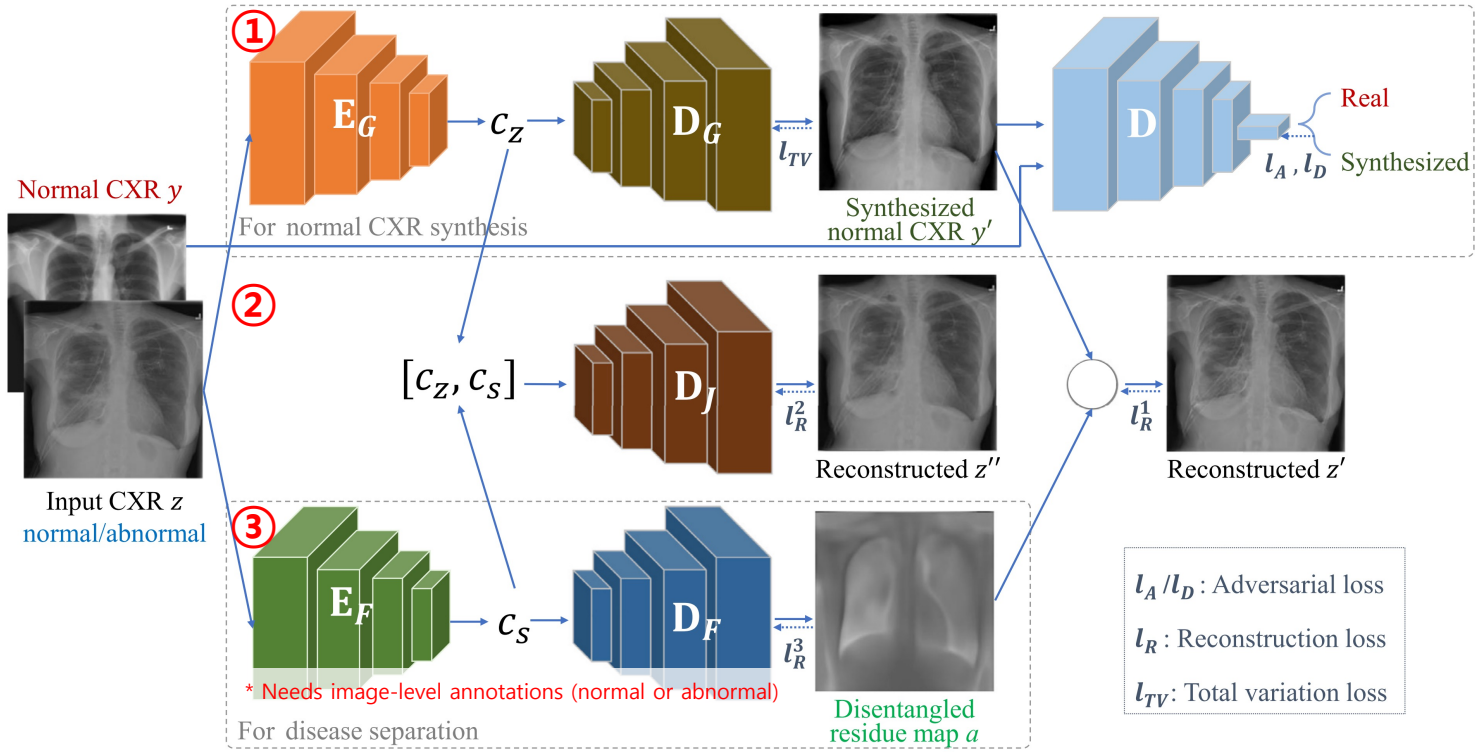



**Fig. 2.** Framework of the proposed disentangled generative model (DGM). An input chest X-ray image is decomposed into a normal CXR and a disease residue map by the DGM, which is trained in an end-to-end manner. The DGM training only requires image-level annotations of the input CXRs, *i.e.*, normal or abnormal. The input and output are shown in solid arrows, while the objective functions are shown in dashed arrows.

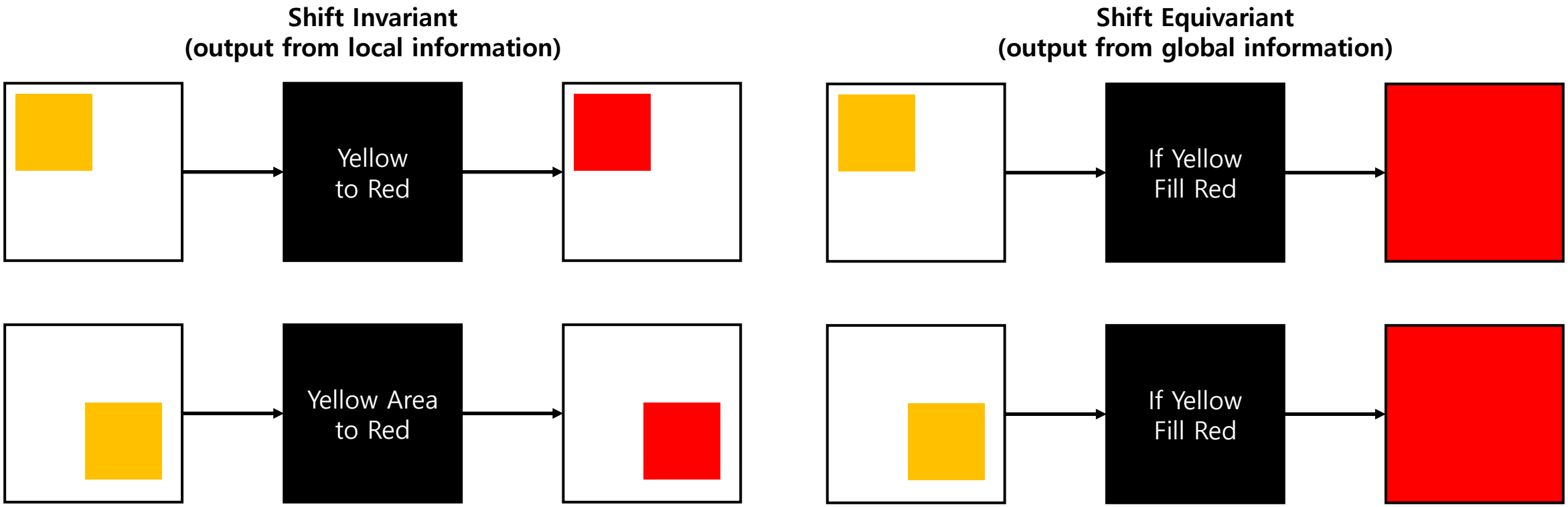
- **Synthesizing all the input data as normal style.**

# Appendix


CVPR
JUNE
19-24
2022
NEW ORLEANS • LOUISIANA

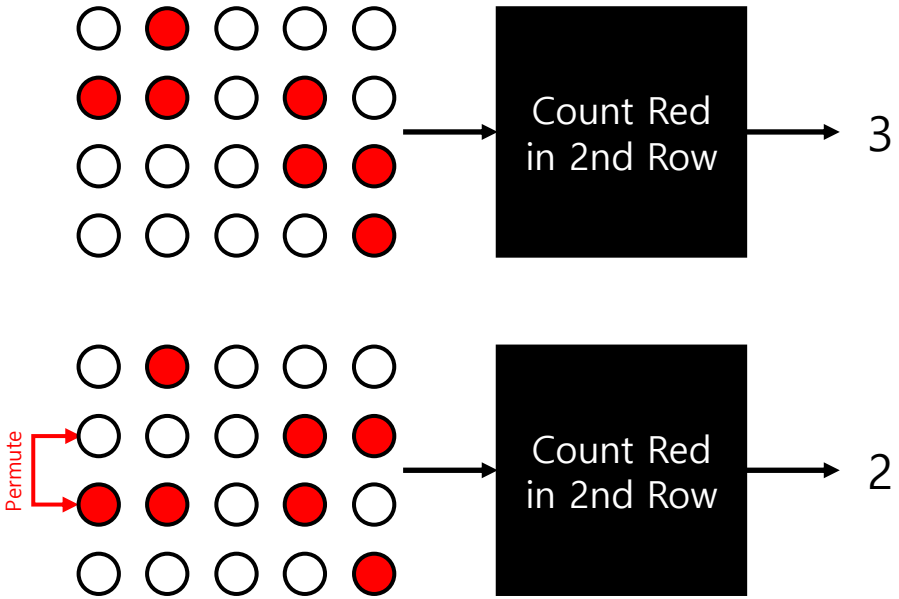

GVL
SUNG KYUN KWAN UNIVERSITY
1398
성균관대학교

# Shift In/Equivariant



# Permutation In/Equivariant

**Permutation Invariant**
**(output from local information)**




**Permutation Equivariant**
**(output from global information)**