Some Recent Advances in Computer Vision (near 2021)

Vision Transformers, Vision-Language models and NeRF

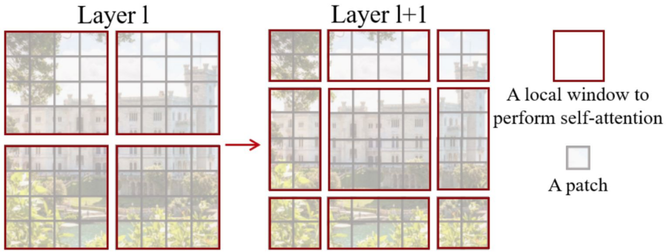
zsc@megvii.com

Oct. 2021

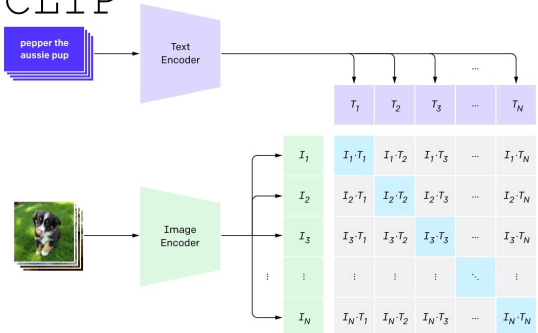
Overview

- Computer Vision meets Natural Language Processing
  - **Vision Transformers: Detection, Classification and Segmentation**
  - Semi- and Self-Supervised Learning: Vision-Language models
- Computer Vision meets Computer Graphics
  - Differential Rendering and Analysis by Synthesis
  - Neural Radiance Field, with applications to SLAM, AR/VR

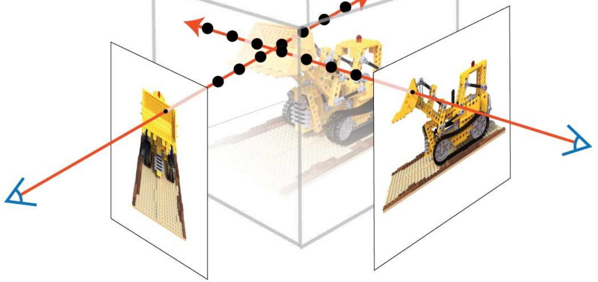
Swin Transformer



CLIP



NeRF



Breakthrough in NLP Language Model

- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (2018)
- GPT-1: Improving Language Understanding by Generative Pre-Training (2018)
- GPT-2: Language Models are Unsupervised Multitask Learners (2019)
- GPT-3: Language Models are Few-Shot Learners (2020)

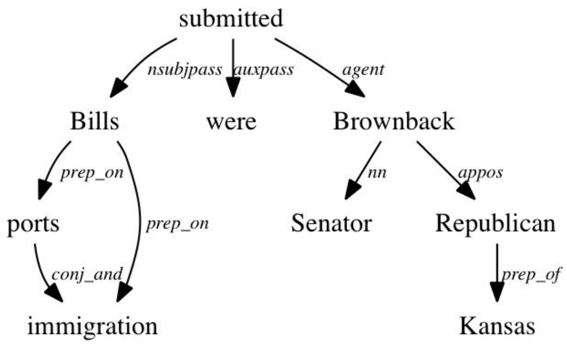
Breakthrough in NLP Language Model

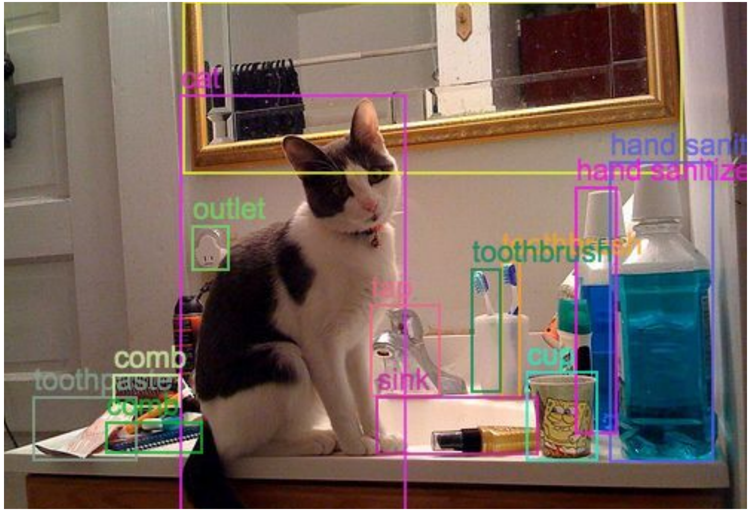
- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (2018)
- GPT-1: Improving Language Understanding by Generative Pre-Training (2018)
- GPT-2: Language Models are Unsupervised Multitask Learners (2019)
- GPT-3: Language Models are Few-Shot Learners (2020)

Two Ingredients: Transformer + Self/Semi-SL

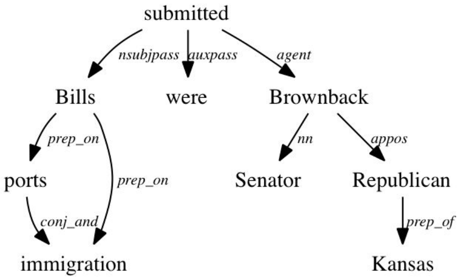
NLP vs. Computer Vision

- Natural Language is
  - naturally tokenized
  - 1D with tree hierarchy
  - "Digital signal"
  - prone to spelling errors
- Vision is
  - continuous: fuzzy spatial relationships, and in scale space
  - 2D (or 3D)
  - "Analog signal": ISP problems, AWB/AE, sensor noise etc.
  - prone to occlusions

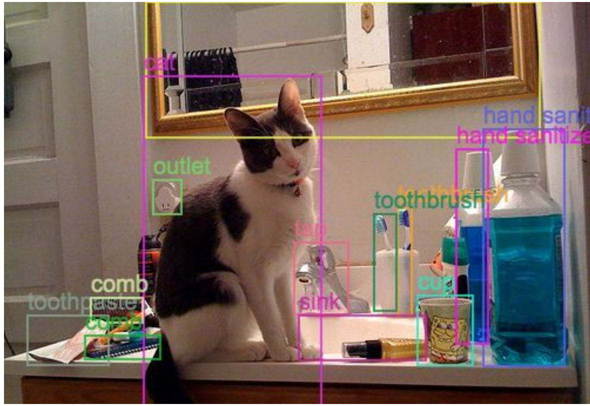




NLP vs. Computer Vision



- Natural Language is
  - naturally tokenized
  - 1D with tree hierarchy
  - "Digital signal"
  - prone to spelling errors
- Vision is
  - continuous: fuzzy spatial relationships, and in scale space
  - 2D (or 3D)
  - "Analog signal": ISP problems, AWB/AE, sensor noise etc.
  - prone to occlusions

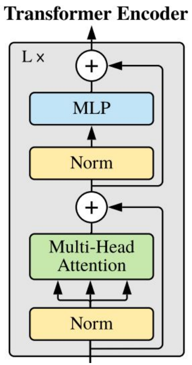


- ***Use Image Patches***
- ***2D attention***
- ***Image Augmentations***
- ***Image Augmentations***

If we can build tree structure out of an image, we can reduce Vision to NLP!

4 years to unleash the power of Vision Transformer

Slide courtesy of Hu Han (modified)



DETR
CNN+Transformer for Detection

VIT
Pure Transformer for Classification

SWIN
A Transformer Backbone

Reason I: General modeling capability

Reason II: Complement convolution

2019.4

2021.1

Reason V: Scalability

2017.06

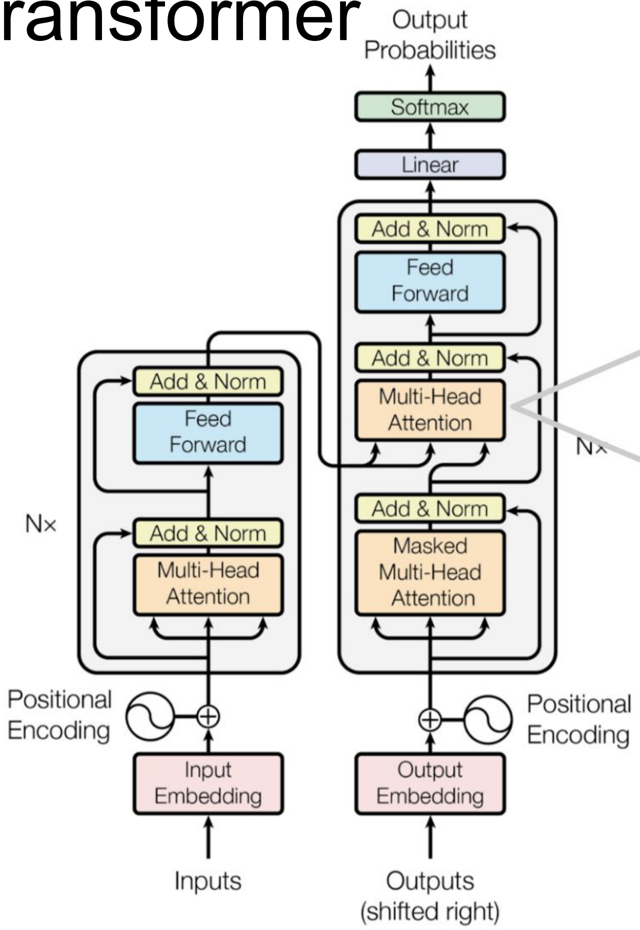
2017.11

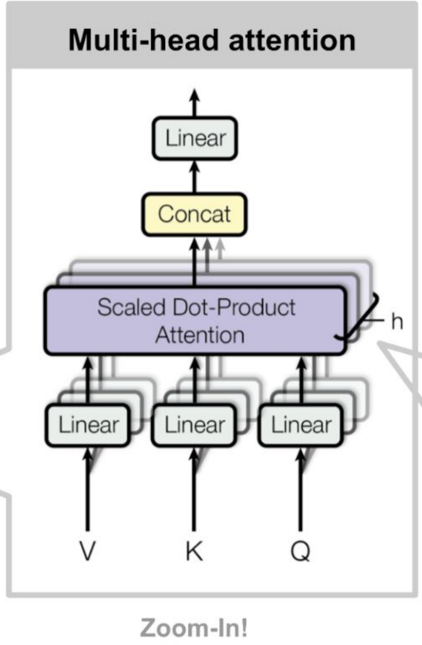
Reason III: Strong modeling power

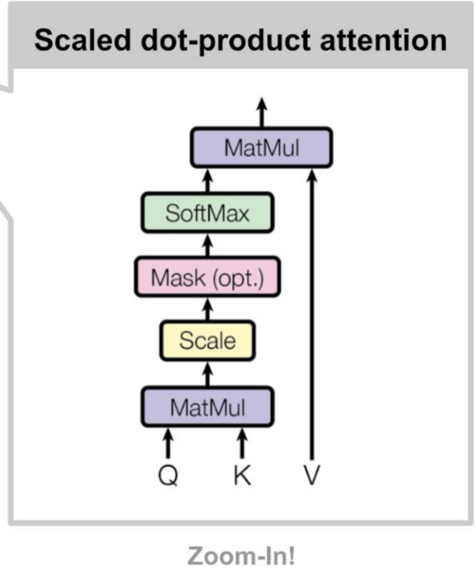
Reason IV: Better connect vision and language

2021.6

Transformer

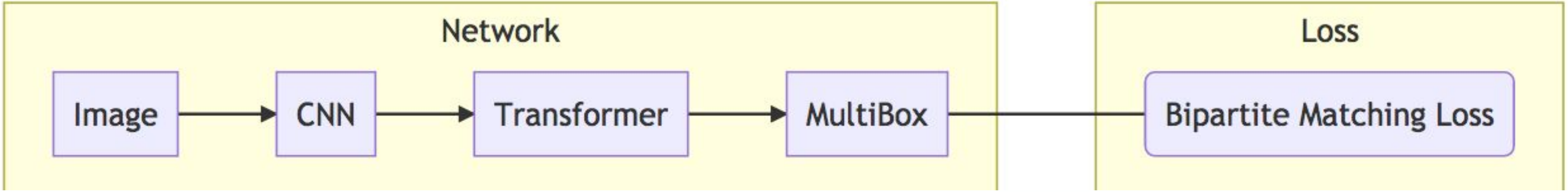


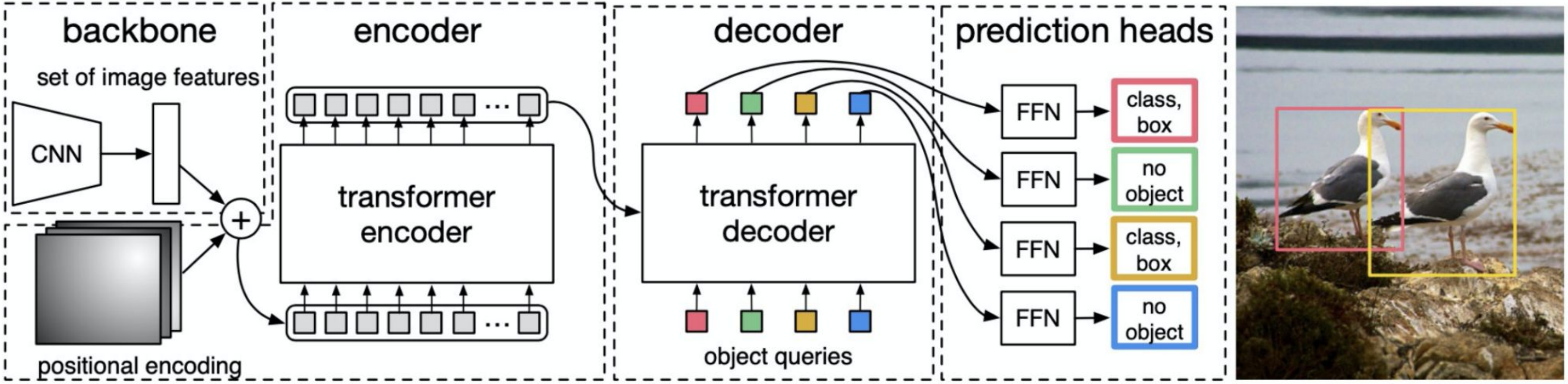




DETR: End-to-End Object Detection with Transformers (2005.12872)

- Draws heavily from MultiBox (*Scalable Object Detection using Deep Neural Networks, CVPR'14*)





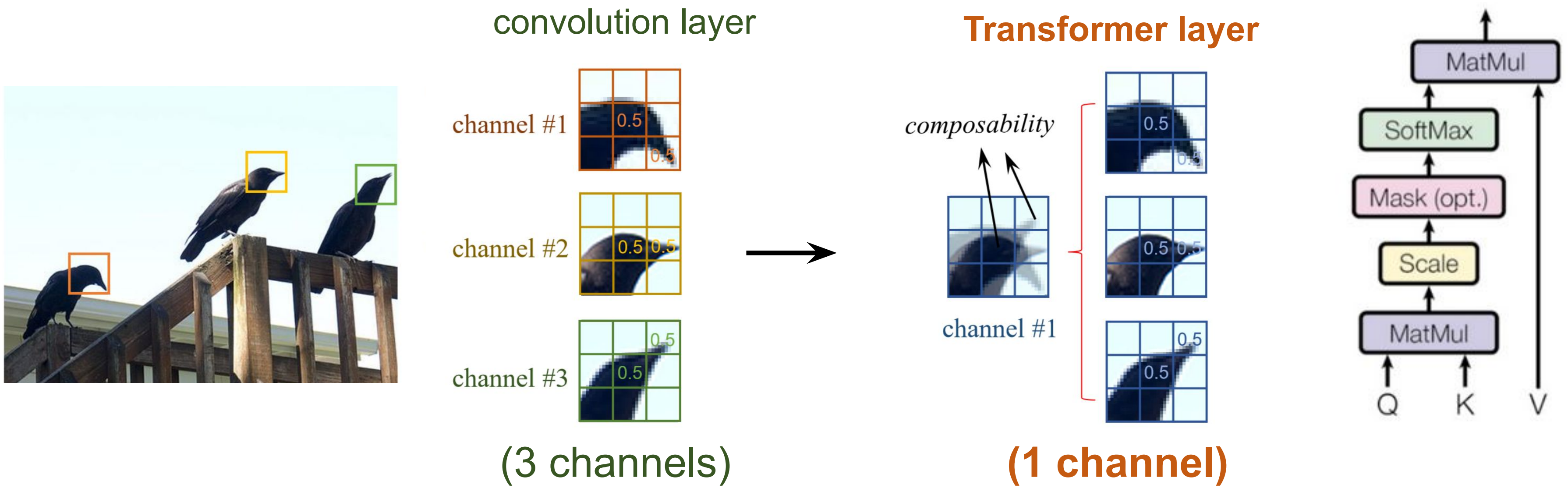
Discussion: Pro's and Con's of DETR

- End-to-end training is often preferred
  - Less tweaking, put gradient backpropagation at work
  - Ease the GT definition burden for immediate steps
- Single Feature
  - More compact representation
  - Thanks to Transformer's QKV attention and mixed-scale representation
- No NMS
  - Transformer serves as decoder: directly outputs a sequence
- Downside
  - Still relies on CNN for Image Prior
  - Slower to train

More Compact Representation thanks to QKV

Slide courtesy of Hu Han (modified)

- Powerful due to <u>adaptive computation</u>
  - "Convolution is exponentially inefficient!"

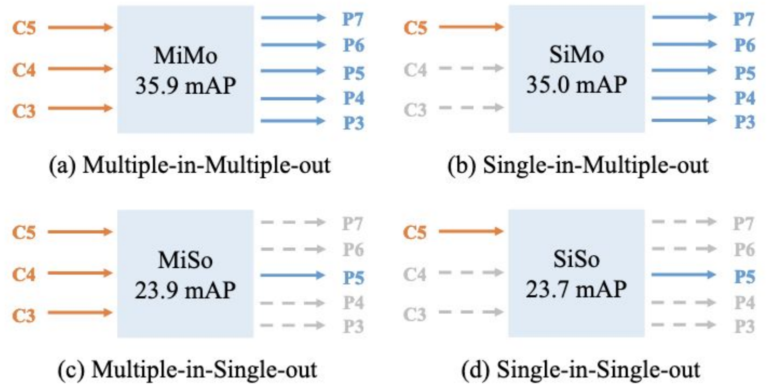


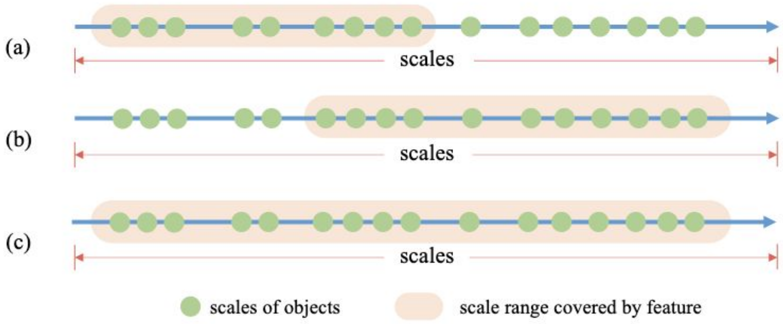
Discussion: Pro's and Con's of DETR

- End-to-end training is often preferred
  - Less tweaking, put gradient backpropagation at work
  - Ease the GT definition burden for immediate steps
- Single Feature
  - More compact representation
  - Thanks to Transformer's QKV attention and mixed-scale representation
- No NMS
  - Transformer serves as decoder: directly outputs a sequence

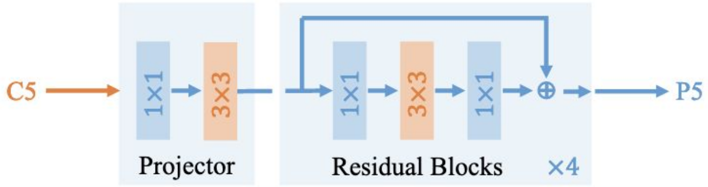
These are not exclusively for Transformer.

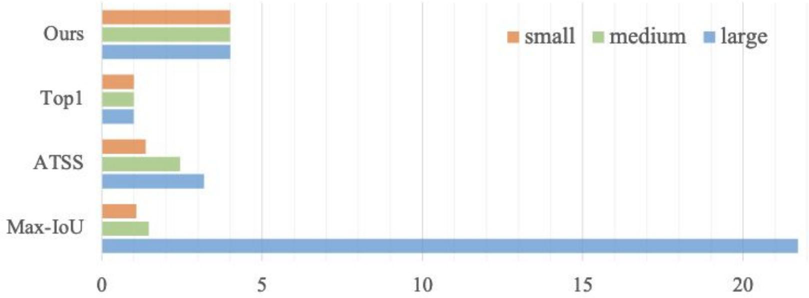
YoloF: You Only Look One-level Feature (2103.09460)





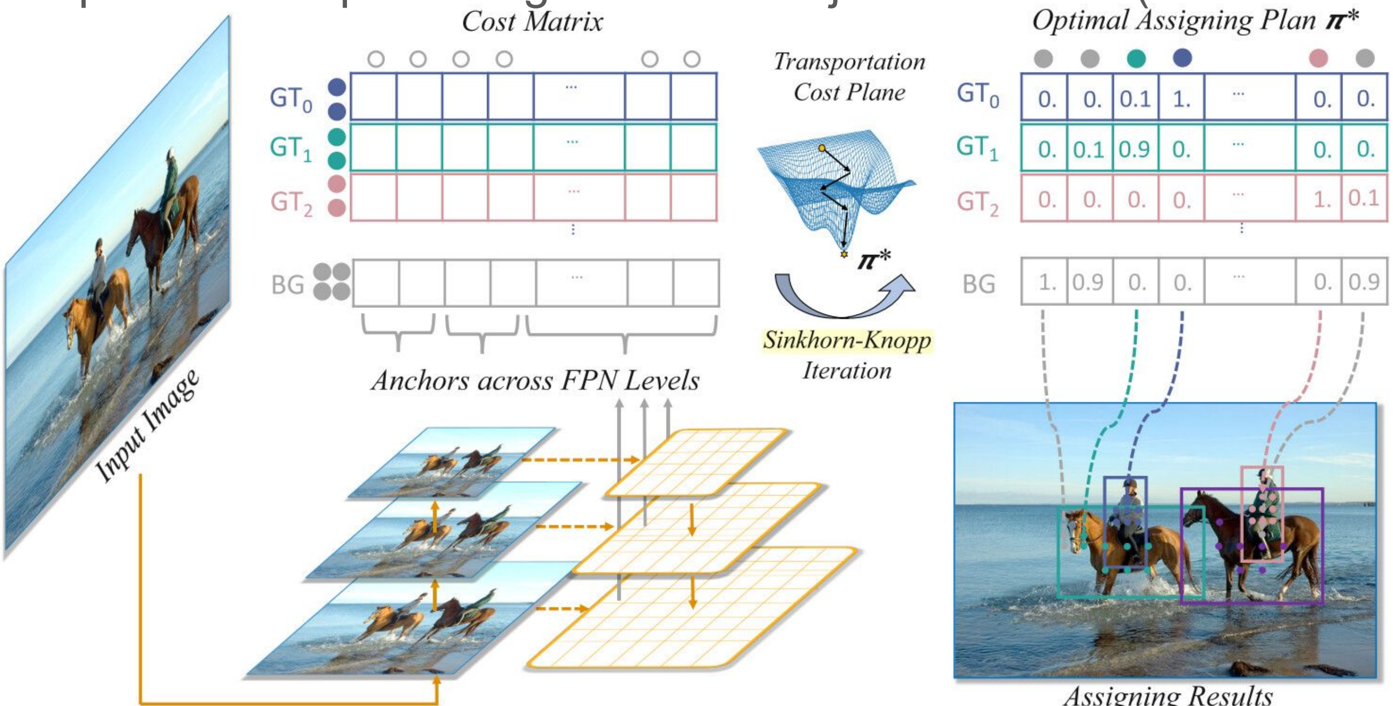
First one-stage single-feature ***realtime*** detector





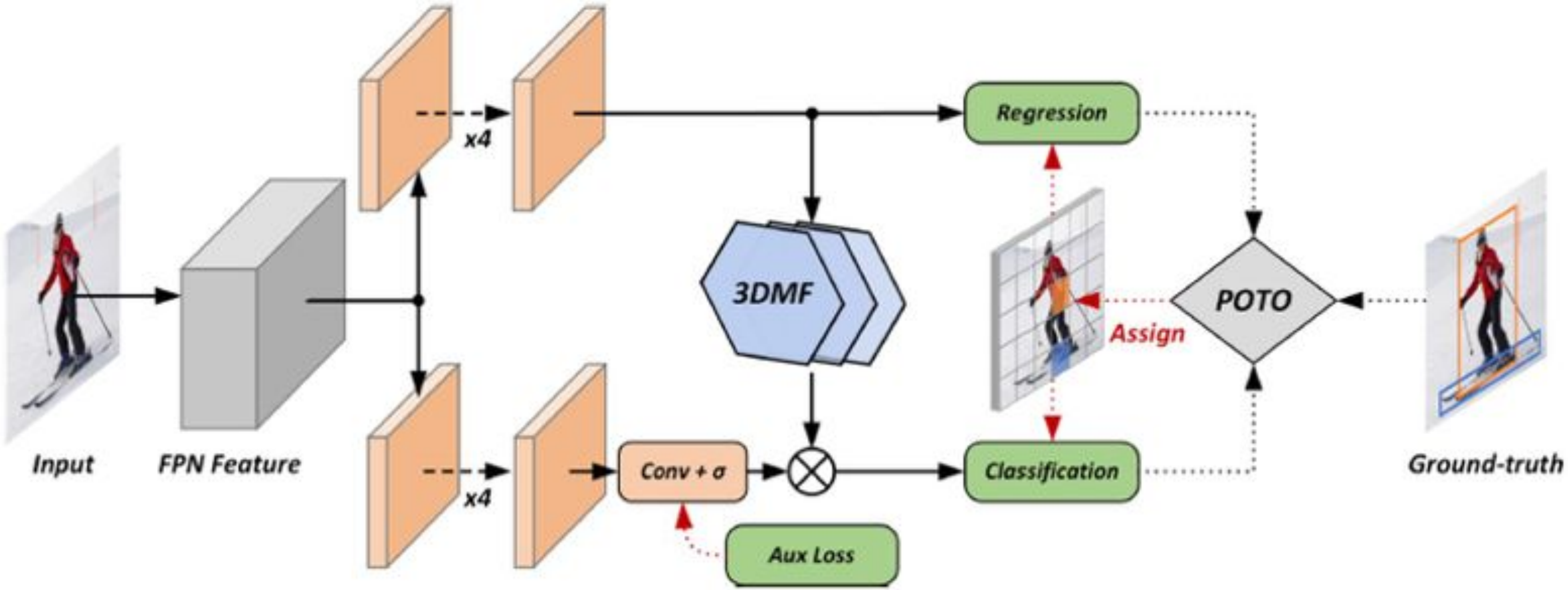
Assignment Problem: match GT boxes against predicted

- BML uses Hungarian method (non-differential) for assignment
- OTA: Optimal Transport Assignment for Object Detection (2103.14259)



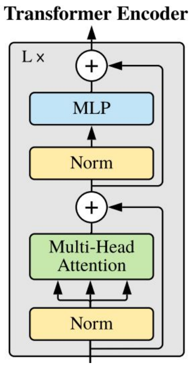
E2E Object Detection with Fully Convolutional Network (2012.03544)

- CNN based End to End Detection
  - One-to-One label-assignment
  - 3D-max-filter for sharp-feature (suppress spatial blurriness caused by sliding window)



4 years to unleash the power of Vision Transformer

Slide courtesy of Hu Han (modified)



2017.06

Reason I: General modeling capability

Reason II: Complement convolution

2017.11

DETR
CNN+Transformer for Detection

2019.4

Reason III: Strong modeling power

VIT
Pure Transformer for Classification

2021.1

Reason IV: Better connect vision and language

SWIN
A Transformer Backbone

Reason V: Scalability

2021.6

ViT: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale (2010.11929)

- Effort for "conv-free"

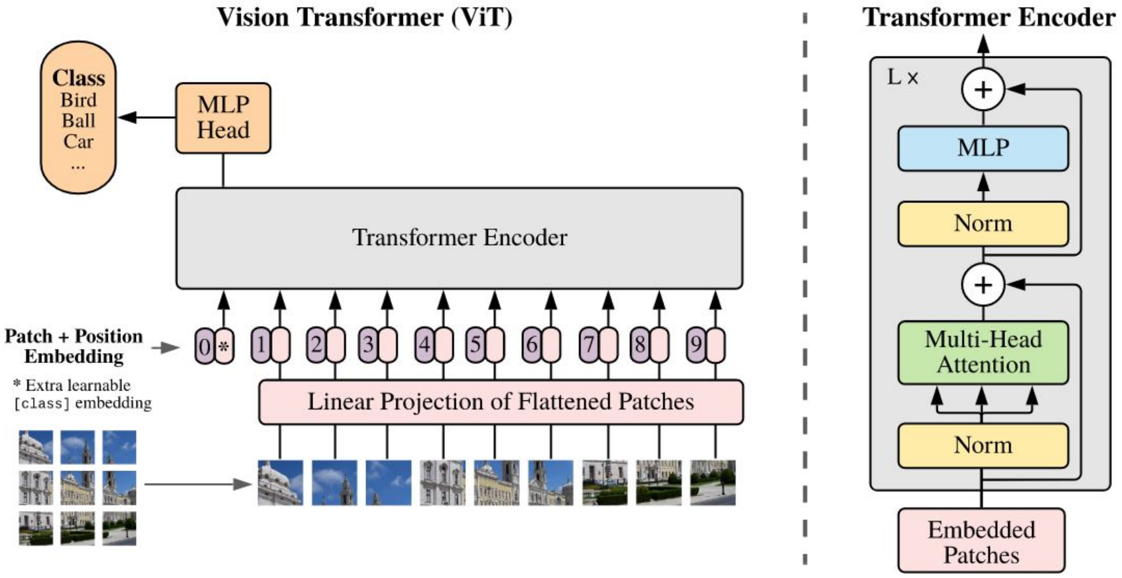
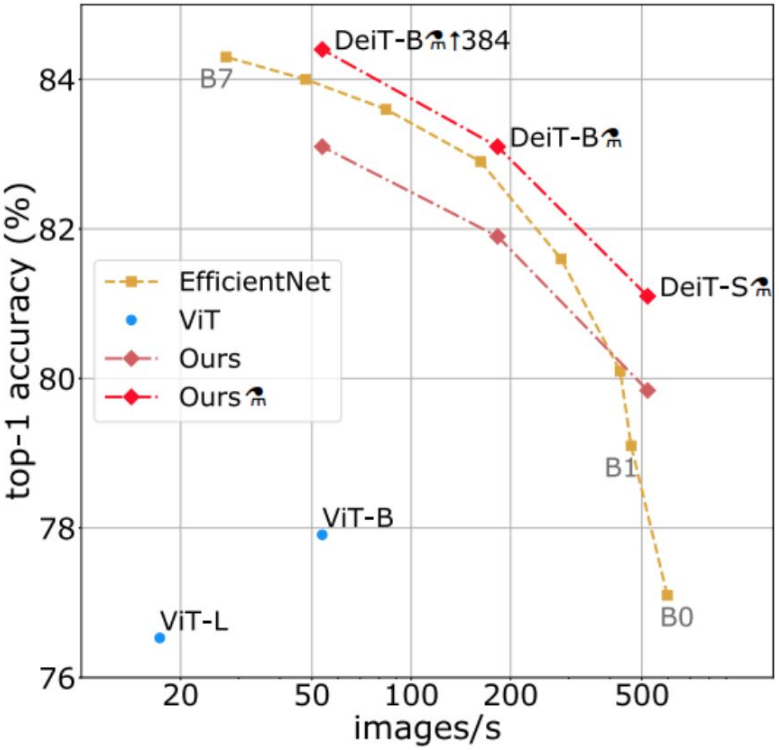


Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings to the resulting sequence of vectors, and feed the patches to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable "classification token" to the sequence. The illustration of the Transformer encoder was inspired by Vaswani et al. (2017).

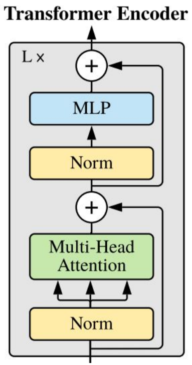
DeiT: Training data-efficient image transformers & distillation through attention (2012.12877)

- Eliminate ViT's reliance on training on ImageNet-21k / JFT300M
- DeiT-B = ViT-B/16
- Raise accuracy by tweaking optimizer, data augmentation and regularization.



4 years to unleash the power of Vision Transformer

Slide courtesy of Hu Han (modified)



DETR
CNN+Transformer for Detection

VIT
Pure Transformer for Classification

SWIN
A Transformer Backbone

Reason I: General modeling capability

Reason II: Complement convolution

2019.4

2021.1

Reason V: Scalability

2017.06

2017.11

Reason III: Strong modeling power

Reason IV: Better connect vision and language

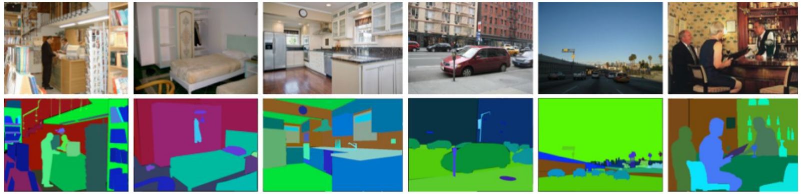
2021.6

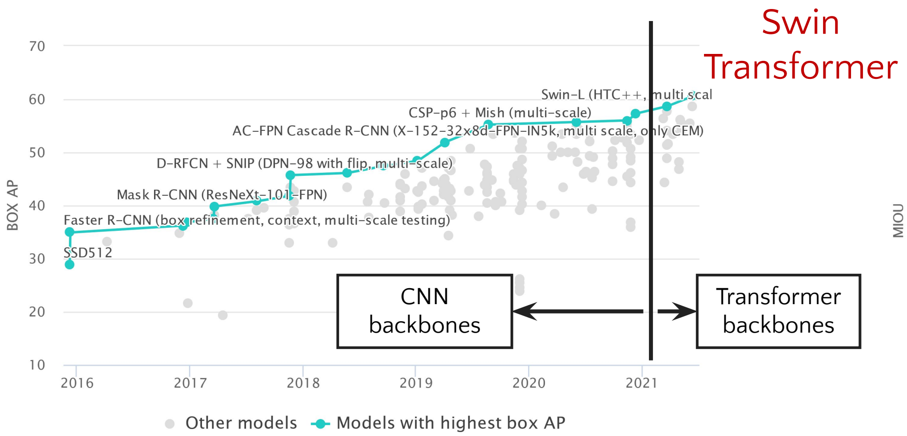
Swin Transformer: General Purpose Backbone

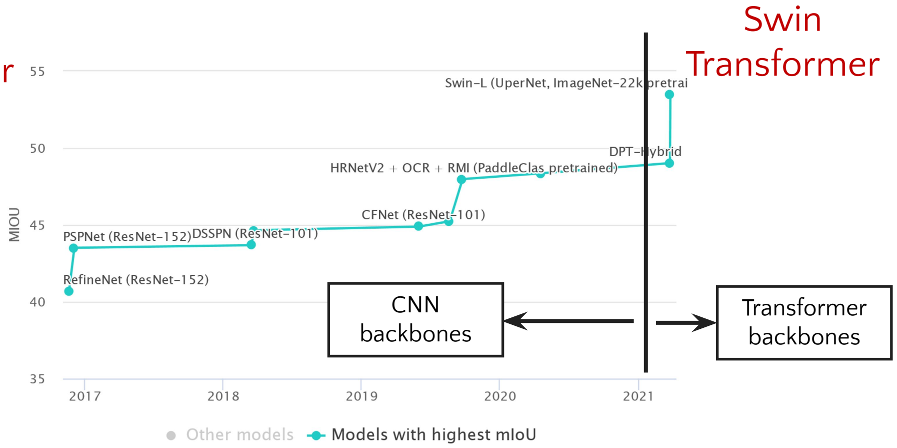
Slide courtesy of Hu Han (modified)

COCO object detection

ADE20K semantic segmentation



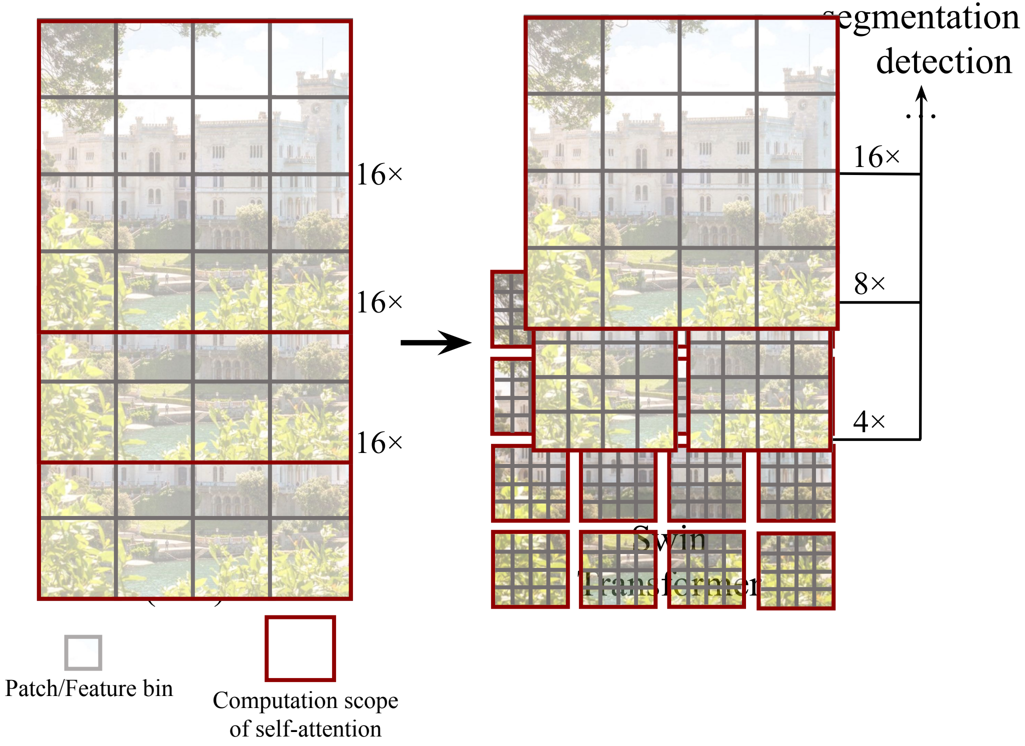




Swin Transformer =

Slide courtesy of Hu Han (modified)

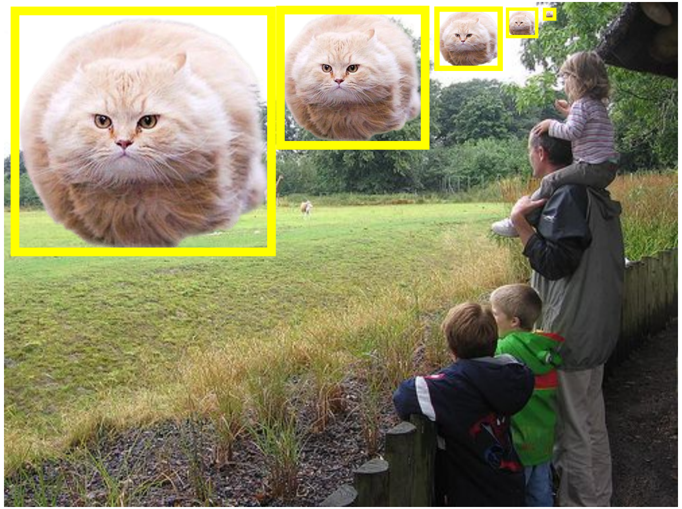
- Transformer
  - Strong modeling power
- + good priors for visual modeling
  - Hierarchy
  - Locality
  - Translational invariance

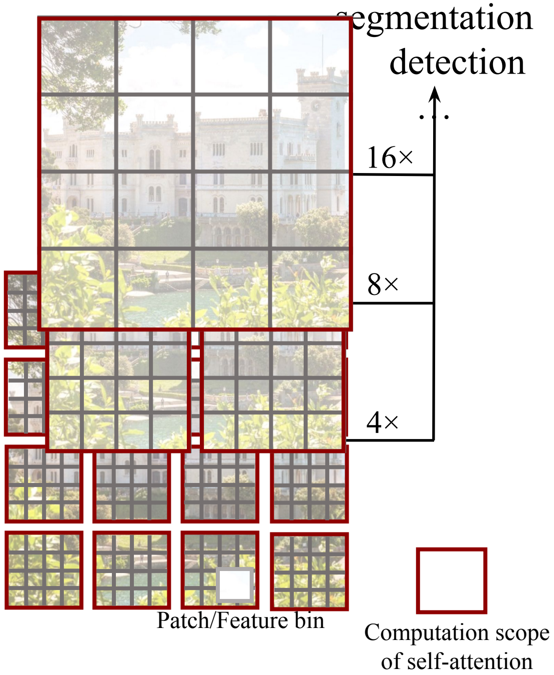


Hierarchy

Slide courtesy of Hu Han (modified)

- Processing objects of different scales



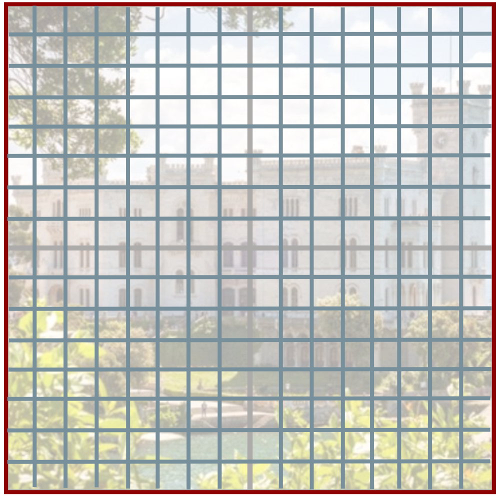


Left figure credit by Ross

Slide courtesy of Hu Han (modified)

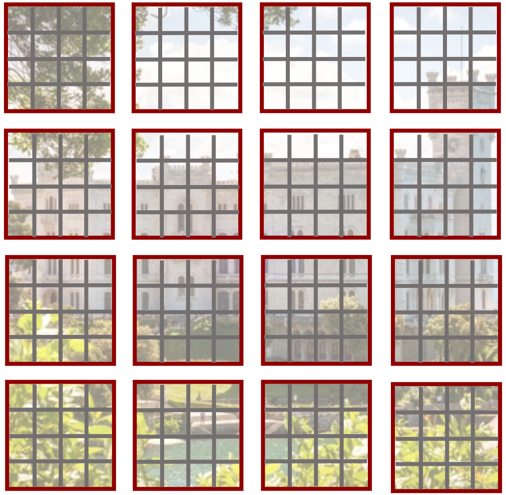
Locality by non-overlapped windows

- Proves beneficial in modeling the high correlation in visual signals (Yann LeCun)
- Linear complexity with increasing image resolution: from $O(n^2)$ to $O(n)$



ViT: 256^2=65536 (Global)

16x less computation

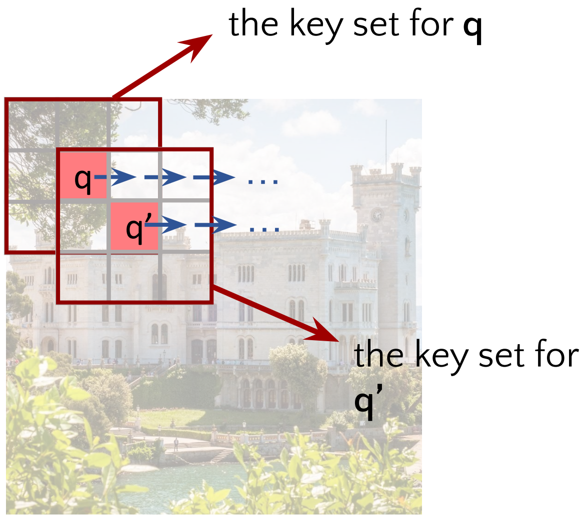


Swin Transformer: 16×16^2=4096 (Local)

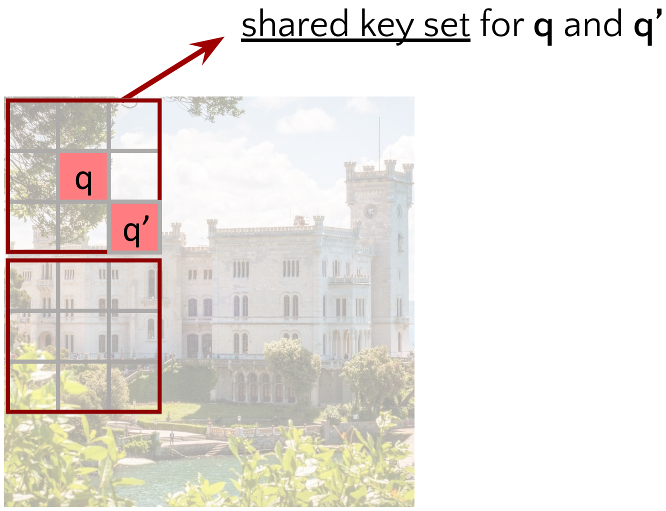
Locality by non-overlapped windows

Slide courtesy of Hu Han (modified)

- Compared to sliding window (LR-Net)
  - Shared key set enables friendly memory access and is thus good for speed (larger than 3x)



sliding window (LR-Net)

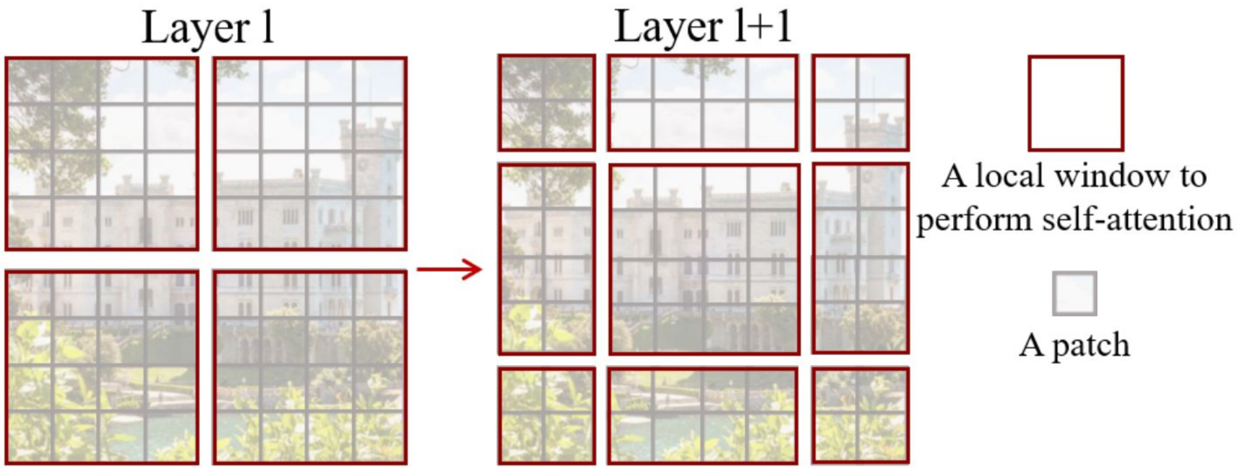


Non-overlapped window (Swin Transformer)

Shifted non-overlapped windows

Slide courtesy of Hu Han (modified)

- Enable cross-window connection
  - Non-overlapped windows will result in no connection between windows
  - Performs as effective or even slightly better than the sliding window approach, due to regularization effects

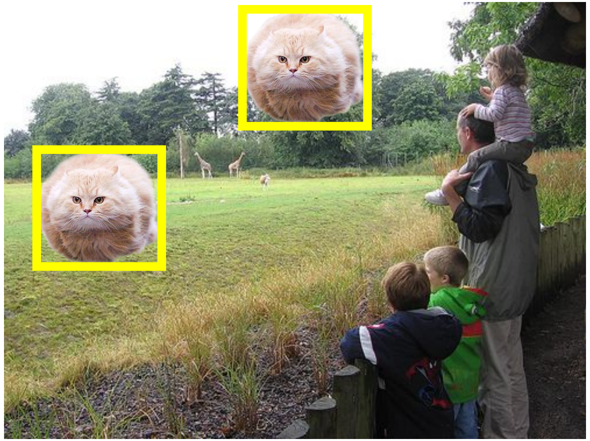


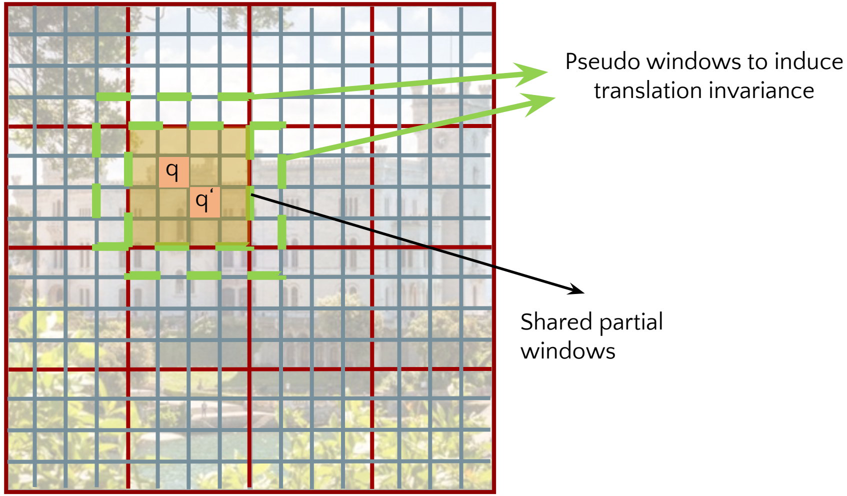
Translational semi-invariance

Slide courtesy of Hu Han (modified)

- Relative position bias plays a more important role in vision than in NLP

$$\text{Attention}(Q, K, V) = \text{SoftMax}(QK^T/\sqrt{d} + \boxed{B})V,$$



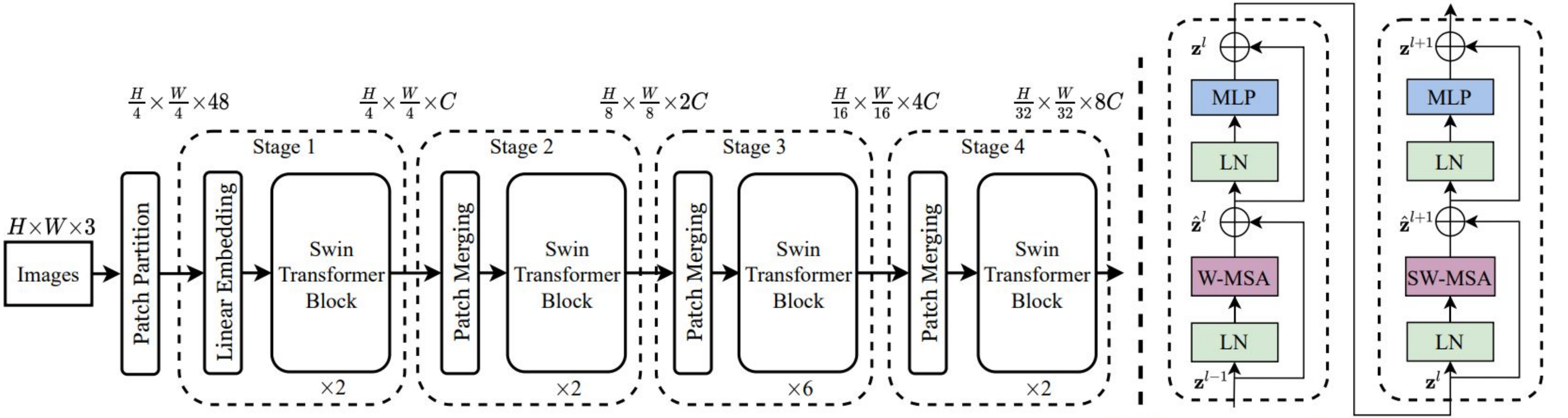


<u>semi-invariance</u> is as effective as full-invariance in our experiments

SWIN: Architecture instantiations

Slide courtesy of Hu Han (modified)

- Resolution of each stage is set similar as ResNet, to facilitate application to down-stream tasks



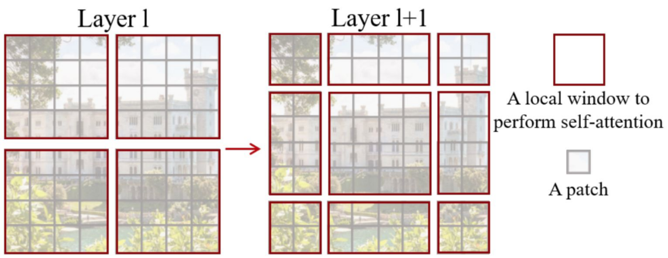
(a) Architecture

(b) Two Successive Swin Transformer Blocks

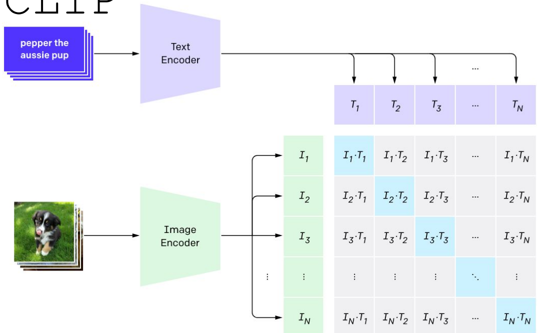
Overview

- Computer Vision meets Natural Language Processing
    - Vision Transformers: Detection, Classification and Segmentation
    - **Semi- and Self-Supervised Learning: Vision-Language models**
- Computer Vision meets Computer Graphics
    - Differential Rendering and Analysis by Synthesis
    - Neural Radiance Field, with applications to SLAM, AR/VR

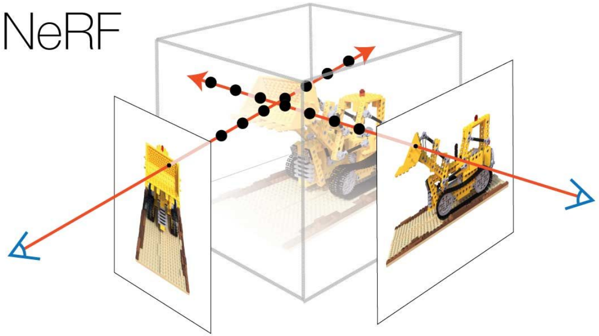
Swin Transformer



CLIP



NeRF



"Training Trilogy": Self-SL + SL + Semi-SL

- Self-Supervised Learning
  - Billion-scale dataset: JFT-300M, Instagram-940M
  - Large models like ResNeXt
- Supervised-finetuning
- Semi-Supervised Learning

"Training Trilogy": Self-SL + SL + Semi-SL

Large-scale Universal Self-SL as a common infrastructure

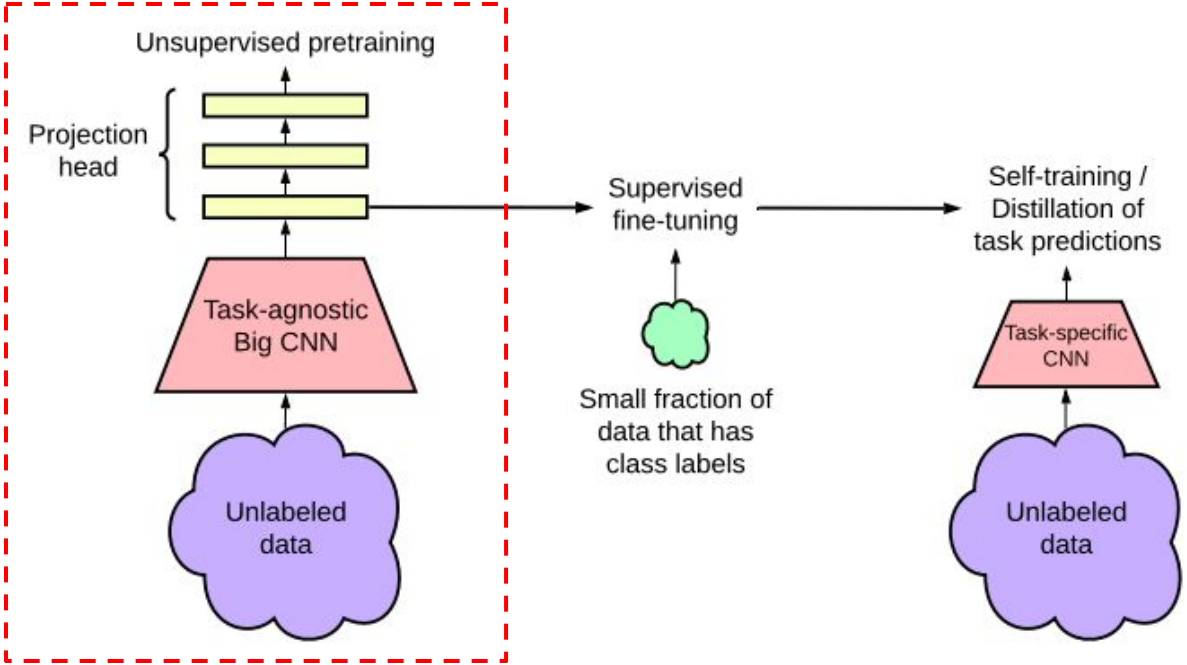


Figure 3: The proposed semi-supervised learning framework leverages unlabeled data in two ways: (1) task-agnostic use in unsupervised pretraining, and (2) task-specific use in self-training / distillation.

SimCLR V2 '20

"Training Trilogy": Self-SL + SL + Semi-SL

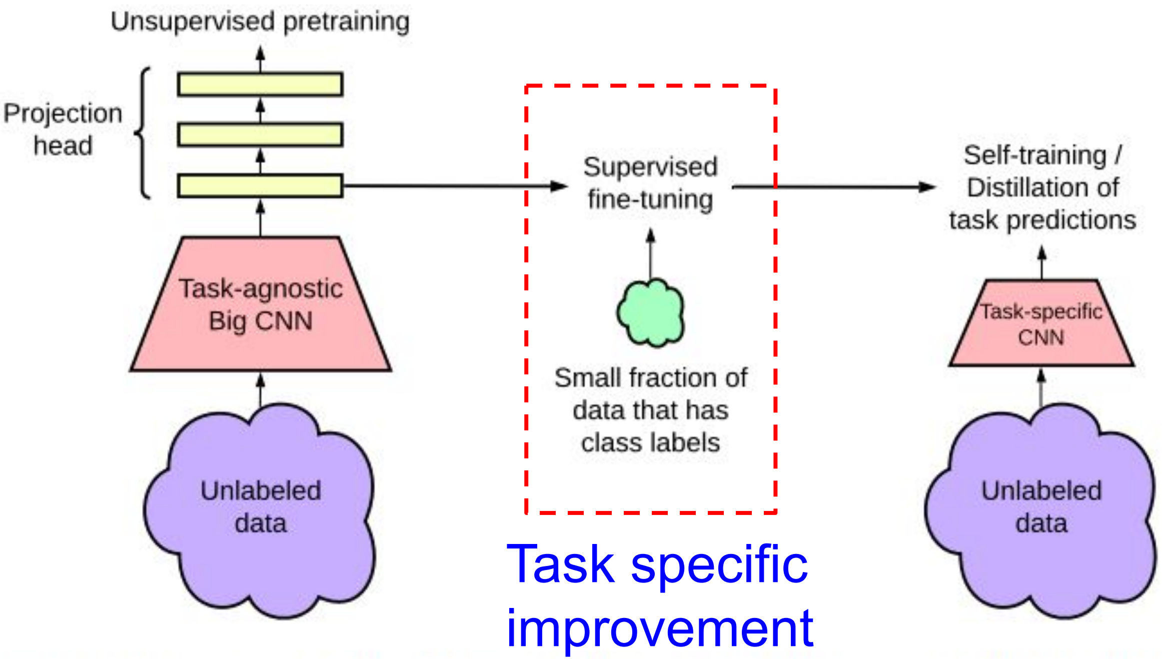
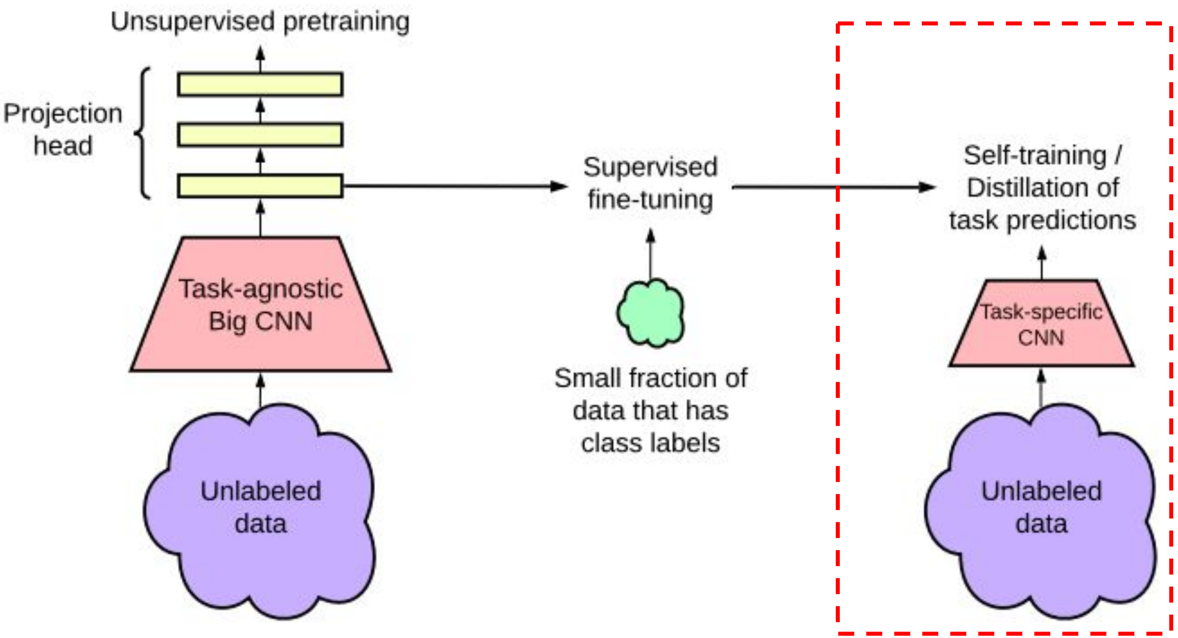


Figure 3: The proposed semi-supervised learning framework leverages unlabeled data in two ways: (1) task-agnostic use in unsupervised pretraining, and (2) task-specific use in self-training / distillation.

SimCLR V2 '20

"Training Trilogy": Self-SL + SL + Semi-SL

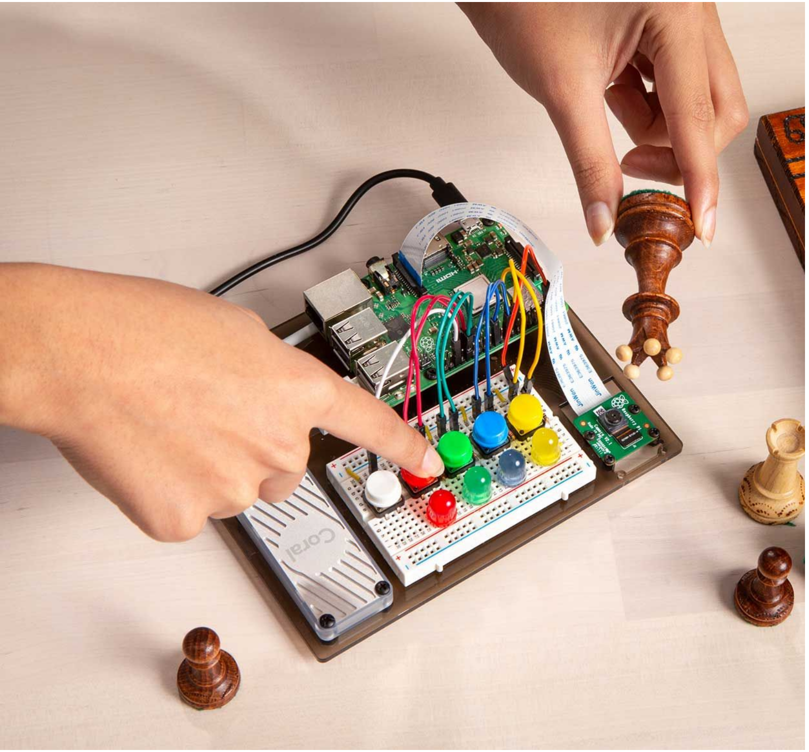


Tuning with real data, maybe on Edge

Figure 3: The proposed semi-supervised learning framework leverages unlabeled data in two ways: (1) task-agnostic use in unsupervised pretraining, and (2) task-specific use in self-training / distillation.

SimCLR V2 '20

Low-Shot Learning with Imprinted Weights (1712.07136)



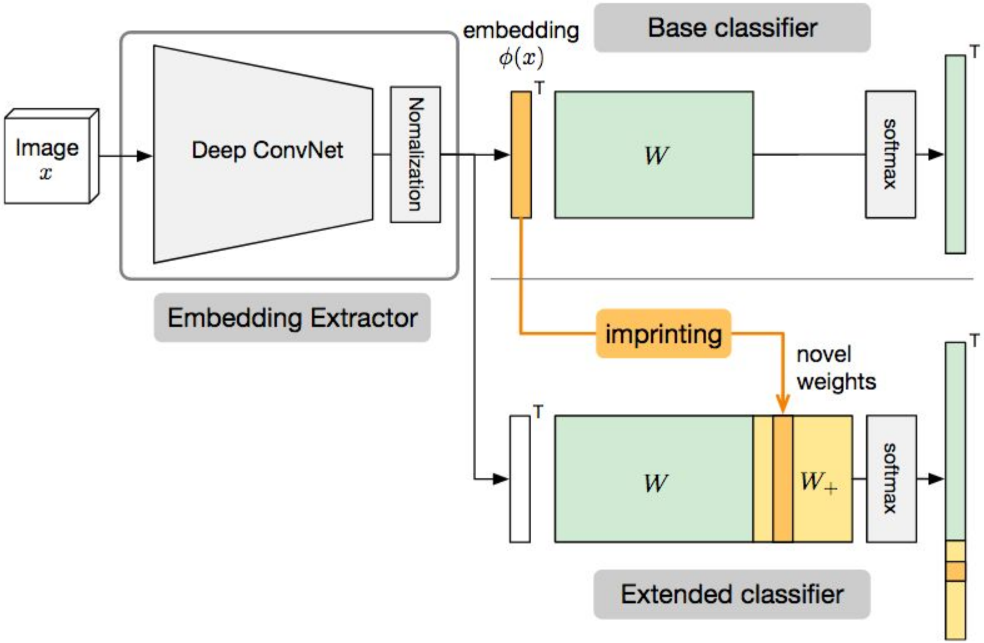
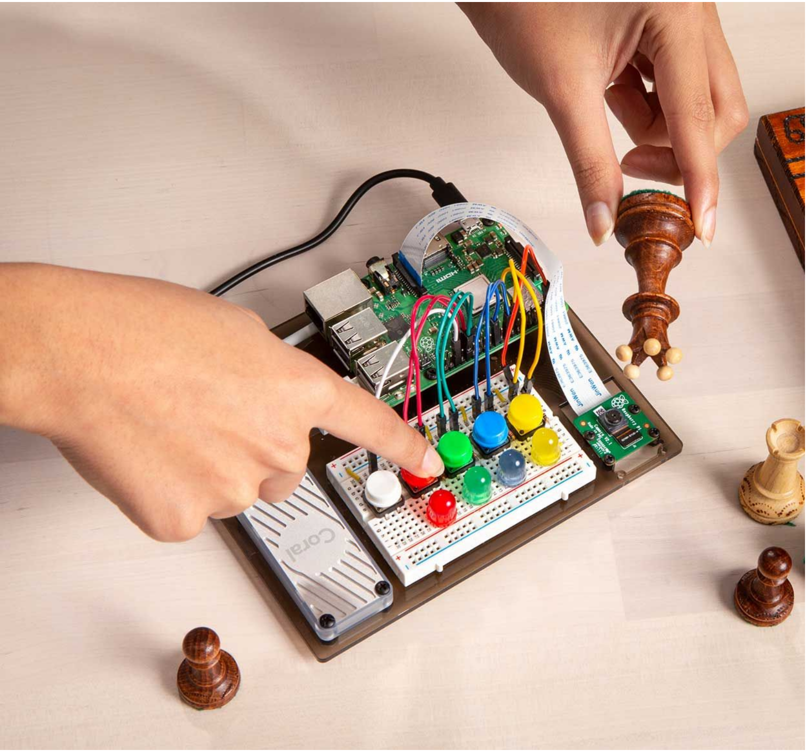
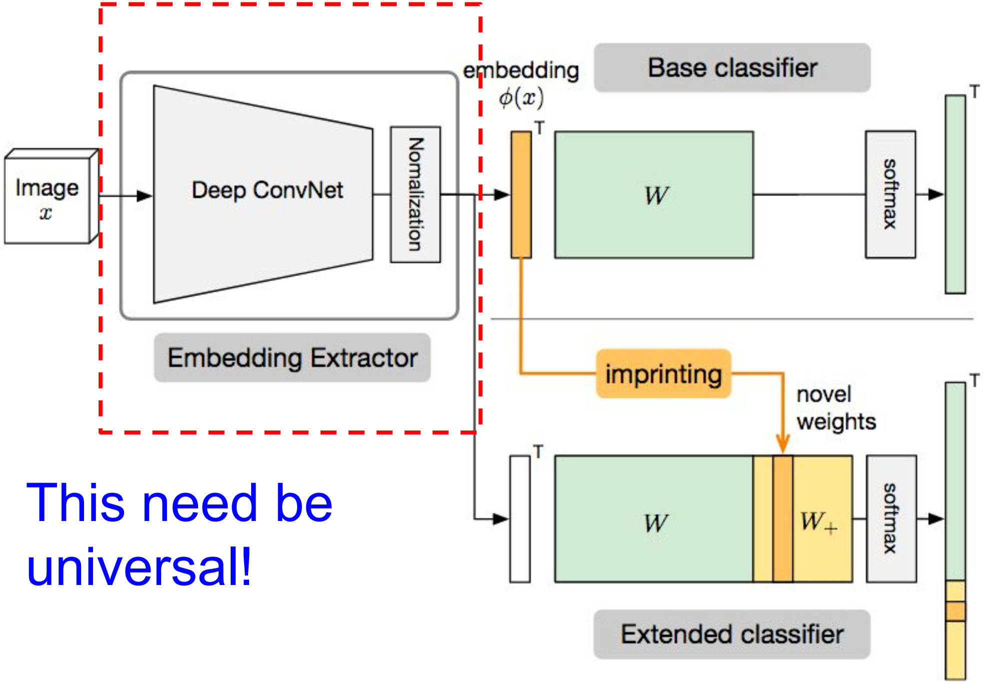


Figure 1. The overall architecture of imprinting. After a base classifier is trained, the embedding vectors of new low-shot examples are used to imprint weights for new classes in the extended classifier.

Low-Shot Learning with Imprinted Weights (1712.07136)





This need be universal!

Figure 1. The overall architecture of imprinting. After a base classifier is trained, the embedding vectors of new low-shot examples are used to imprint weights for new classes in the extended classifier.

MEGVII 旷视

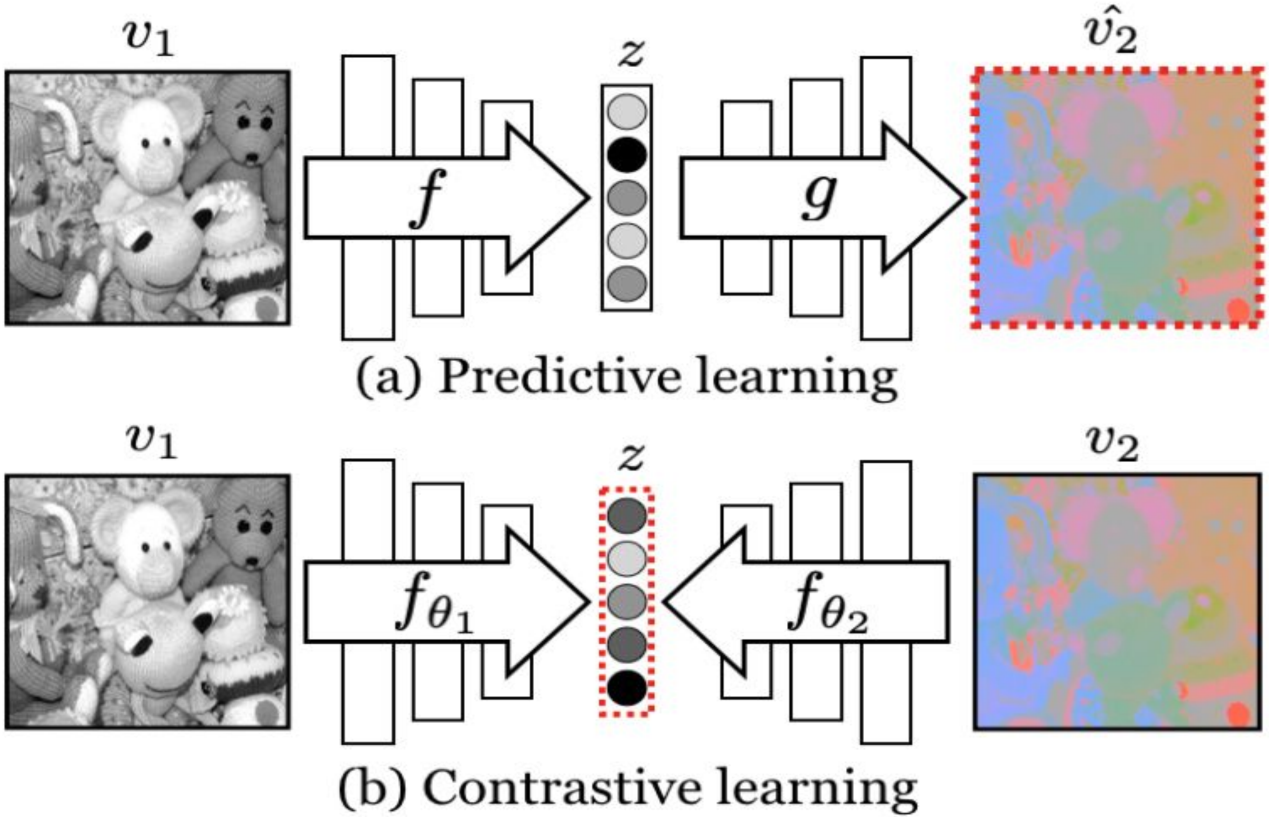
Self-SL by Auxiliary task: Inpainting

- Context Encoders: Feature Learning by Inpainting '16

(a) Input context | (b) Human artist | (c) Context Encoder ($L2$ loss) | (d) Context Encoder ($L2$ + Adversarial loss)

Predictive Learning vs. Contrastive Learning

- SimCLR
- MoCo
- BYOL



Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning (2006.07733)

- Free of Negative Samples
- Later works: having ***some*** differences between two branches is enough

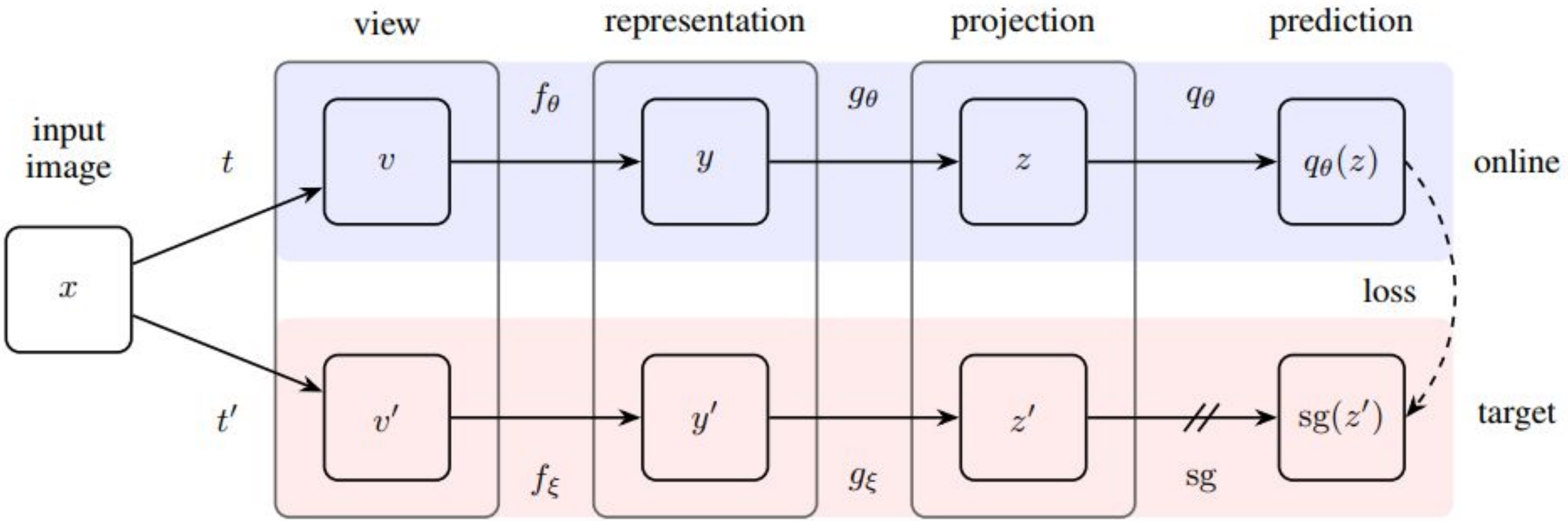


Figure 2: BYOL's architecture. BYOL minimizes a similarity loss between $q_\theta(z)$ and $\mathrm{sg}(z')$, where θ are the trained weights, ξ are an exponential moving average of θ and sg means stop-gradient. At the end of training, everything but f_θ is discarded and y is used as the image representation.

Self-SL by Generative Prior: Pixel-by-pixel Image Reconstruction (Jun. 17, 2020)

- Image GPT

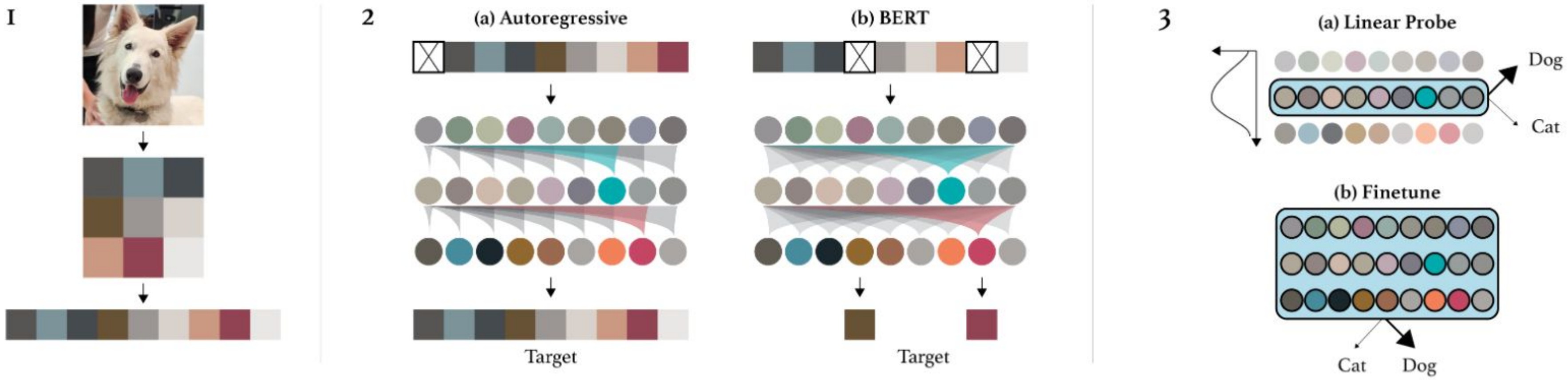


Figure 1. An overview of our approach. First, we pre-process raw images by resizing to a low resolution and reshaping into a 1D sequence. We then chose one of two pre-training objectives, auto-regressive next pixel prediction or masked pixel prediction. Finally, we evaluate the representations learned by these objectives with linear probes or fine-tuning.

Self-SL by Generative Prior: Pixel-by-pixel Image Reconstruction

- Image GPT

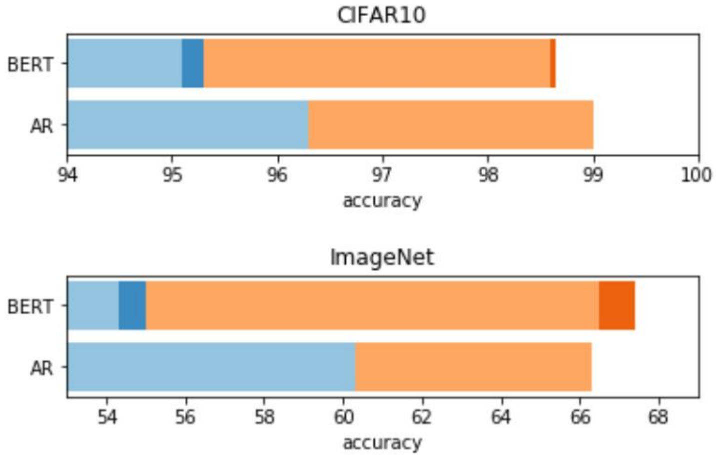
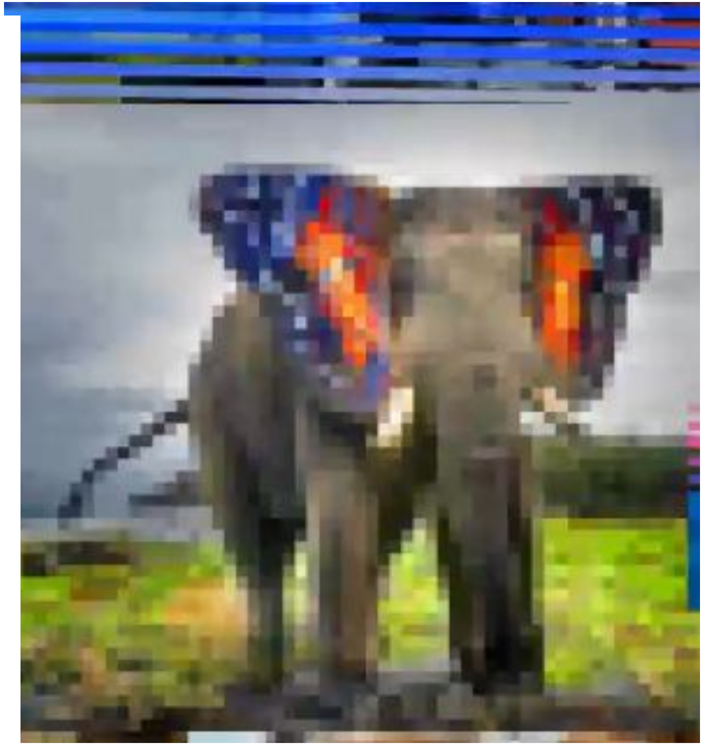


Figure 4. Comparison of auto-regressive pre-training with BERT pre-training using iGPT-L at an input resolution of $32^2 \times 3$. Blue bars display linear probe accuracy and orange bars display fine-tune accuracy. Bold colors show the performance boost from ensembling BERT masks. We see that auto-regressive models produce much better features than BERT models after pre-training, but BERT models catch up after fine-tuning.

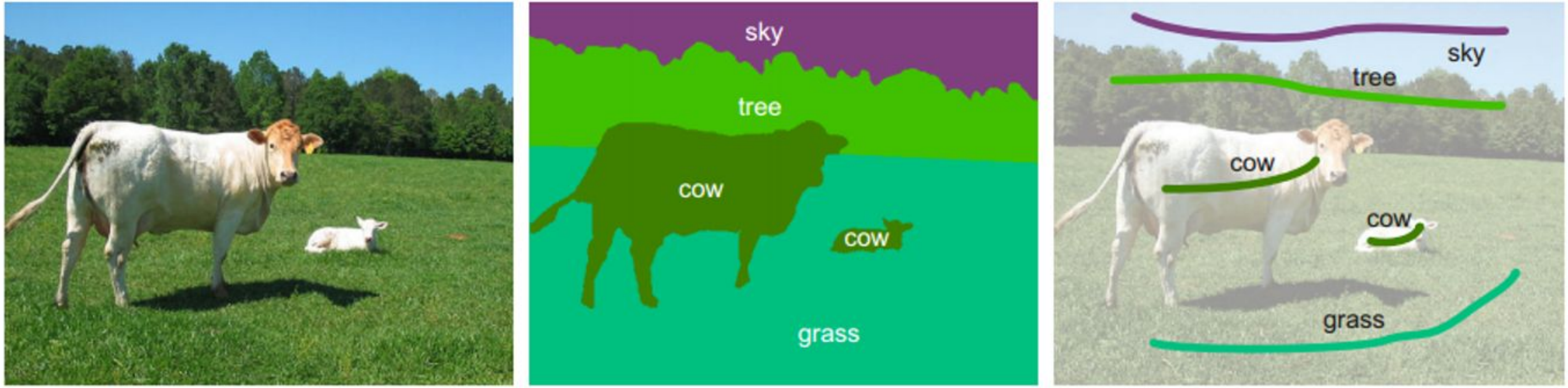


Vision Language Models

- CLIP and Wudao (multimodal)
  - Built on the common Transformer Architecture for NLP and CV
  - Weaker supervision, but still supervised learning
- Applications
  - Zero-shot Image Classification
  - Text to Image

Weakly supervised learning

- Labeling can be very expensive, weaker labels can help reduce cost



(a) image (b) mask annotation (c) scribble annotation

- Abundance of weak labels on Internet
  - Instagram Hashtags
    - **#beautiful** #fashion #art **#photographer** #bhfyp #likeforlikes **#travel** #instadaily #photoshoot **#smile** #model **#naturephotography** ...

CLIP: Learning Transferable Visual Models From Natural Language Supervision (2103.00020)

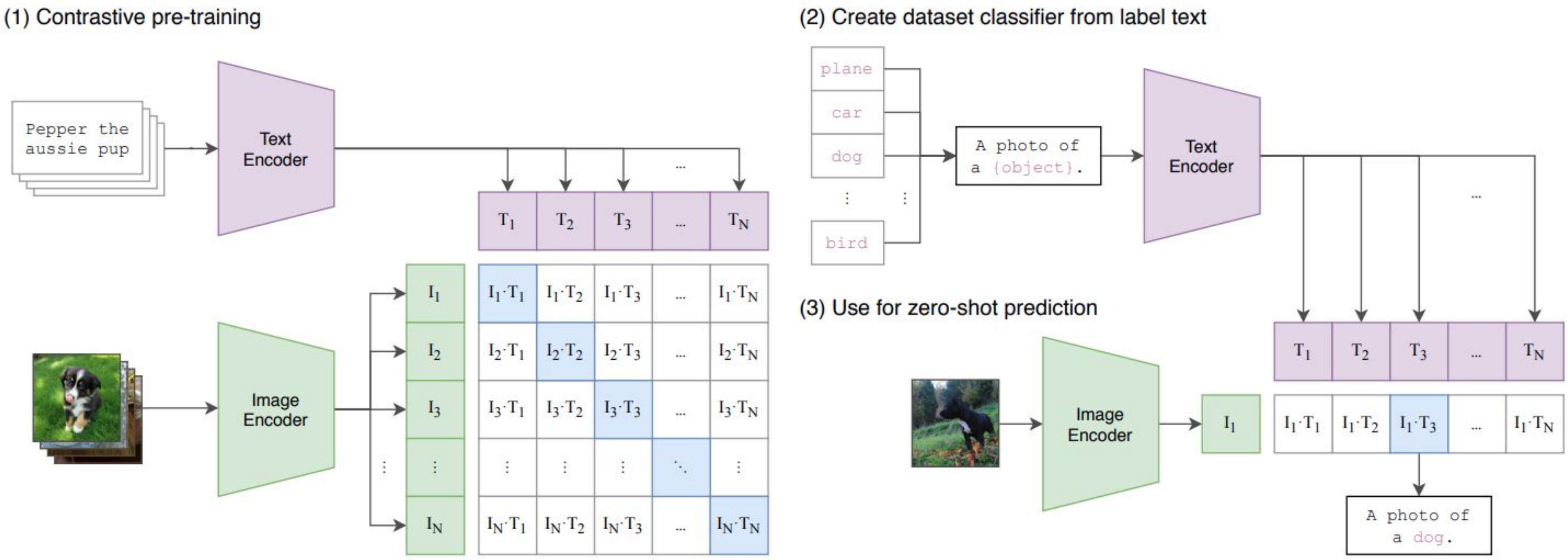
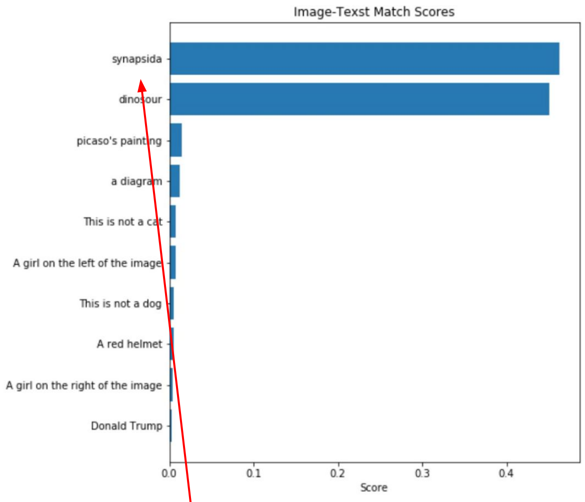
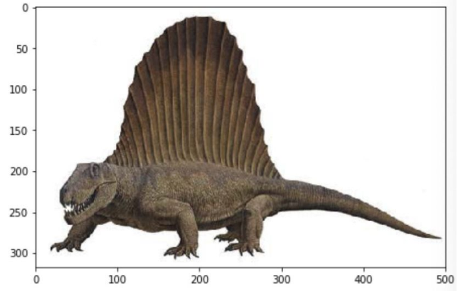


Figure 1. Summary of our approach. While standard image models jointly train an image feature extractor and a linear classifier to predict some label, CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples. At test time the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

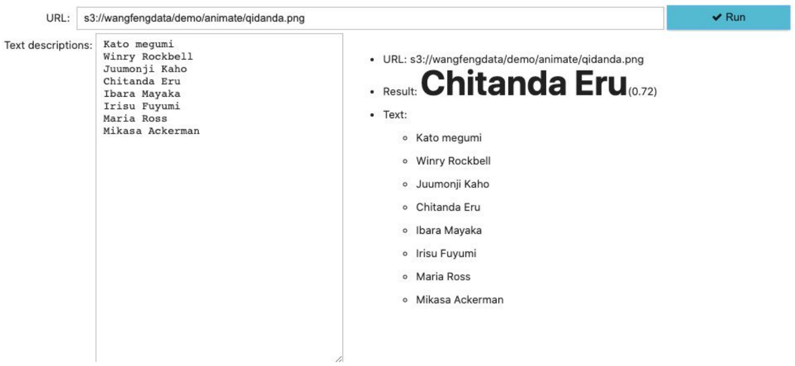
CLIP: Learning Transferable Visual Models From Natural Language Supervision (2103.00020)

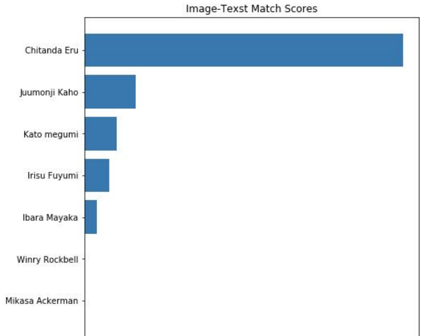
- It rocks
    - can handle some misspellings with BPE from NLP
    - knows trivias like Cartoon Character names

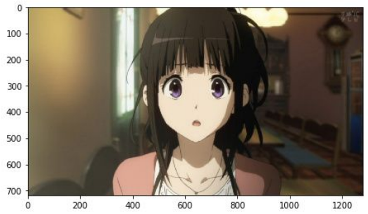




misspelled *synapsids* for *synapsida*



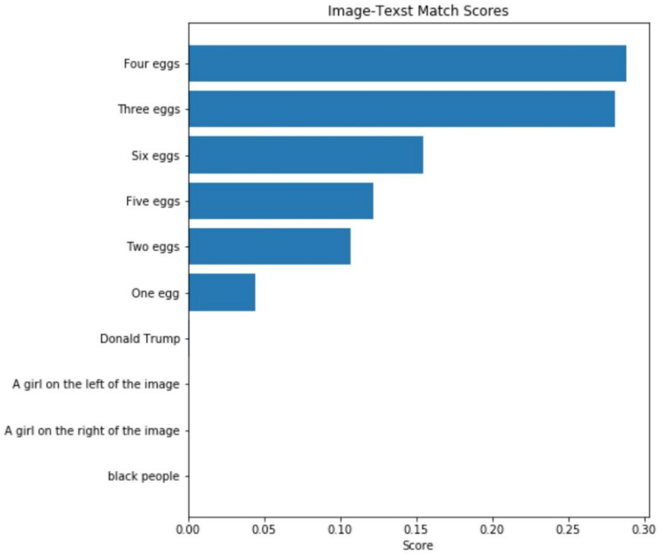


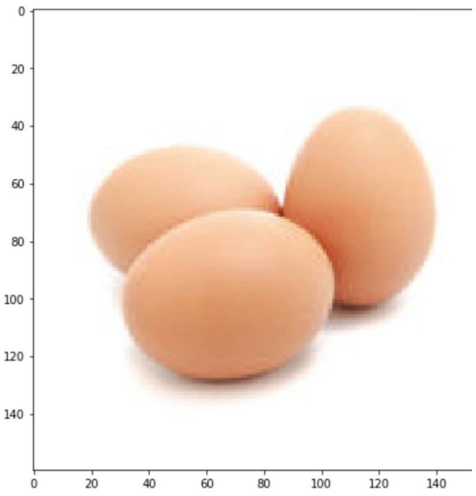


CLIP: Learning Transferable Visual Models From Natural Language Supervision (2103.00020)

- But still
  - can't count
  - don't quite understand "not"

Result: Four eggs (0.29)

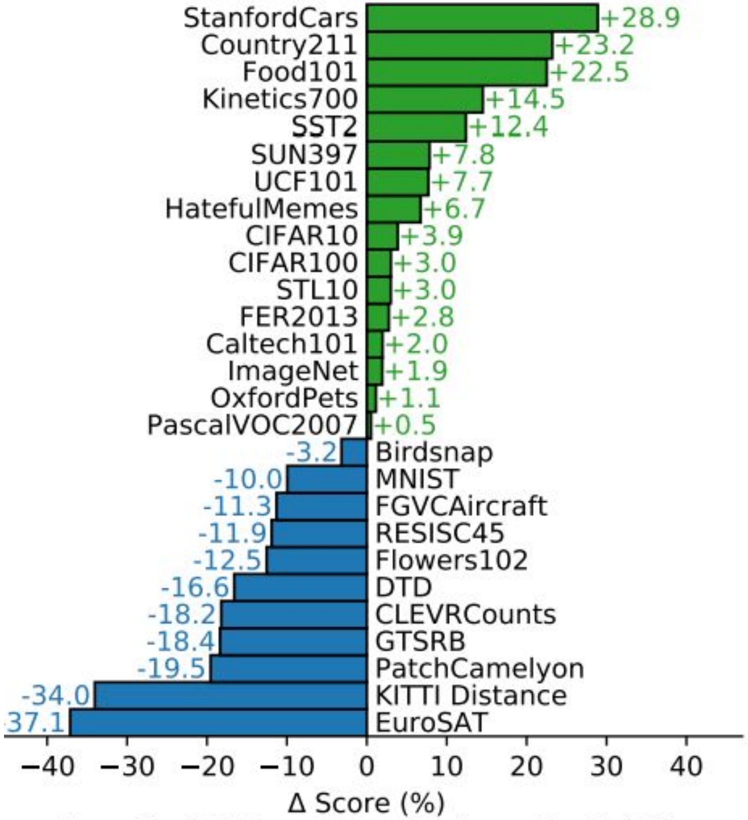




MEGVII 旷视

CLIP: Learning Transferable Visual Models From Natural Language Supervision (2103.00020)

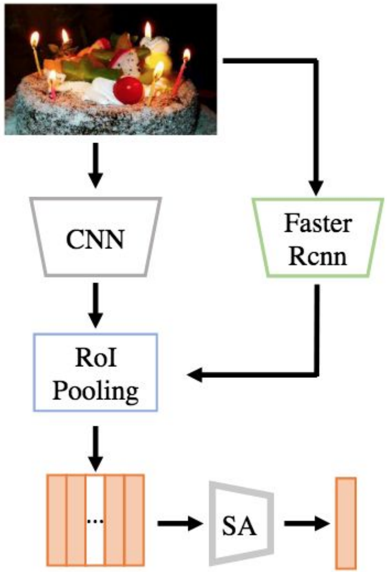
DATASET		IMAGENET RESNET101	CLIP VIT-L
ImageNet		76.2%	76.2%
ImageNet V2		64.3%	70.1%
ImageNet Rendition		37.7%	88.9%
ObjectNet		32.6%	72.3%
ImageNet Sketch		25.2%	60.2%
ImageNet Adversarial		2.7%	77.1%



Zero-Shot CLIP vs. Linear Probe on ResNet50

WenLan: Bridging Vision and Language by Large-Scale Multi-Modal Pre-Training (2103.06561)

- MoCo-style contrastive learning
- CNN-Transformer Encoder



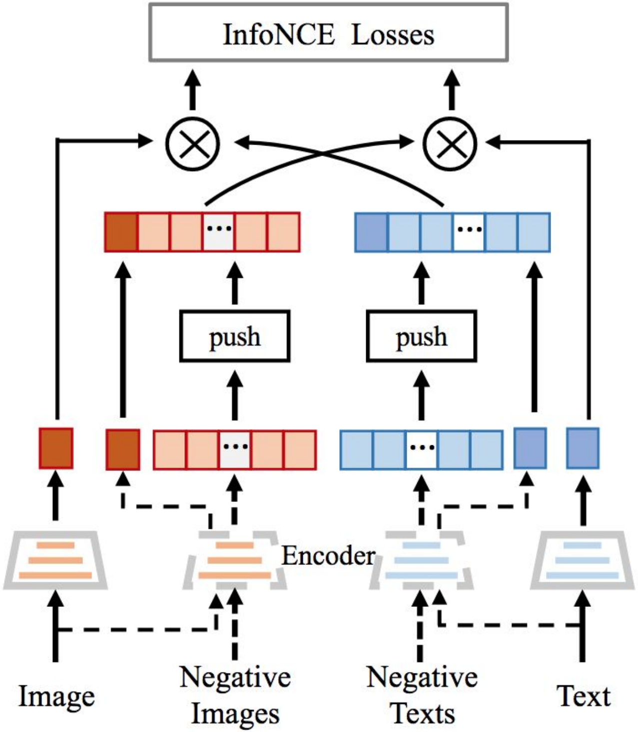
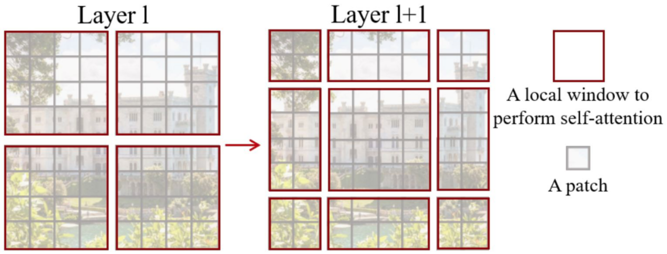


Figure 2. A schematic illustration of our BriVL model within the cross-modal contrastive learning framework.

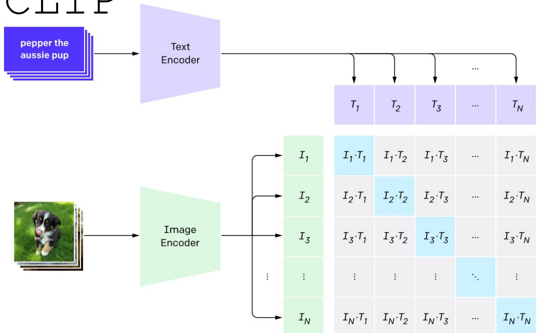
Overview

- Computer Vision meets Natural Language Processing
  - Vision Transformers: Detection, Classification and Segmentation
  - Semi- and Self-Supervised Learning: Vision-Language models
- Computer Vision meets Computer Graphics
  - **Differential Rendering and Analysis by Synthesis**
  - Neural Radiance Field, with applications to SLAM, AR/VR

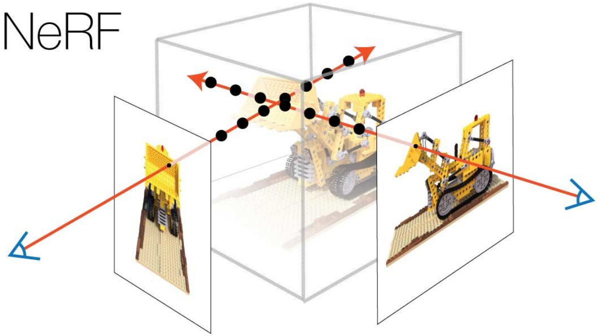
Swin Transformer



CLIP



NeRF



Analyzing an Image: Image to Attributes

David Marr

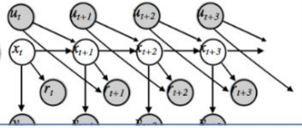
Three levels of description *(David Marr, 1982)*

Computational
Why do things work the way they do?
What is the goal of the computation?
What are the unifying principles?

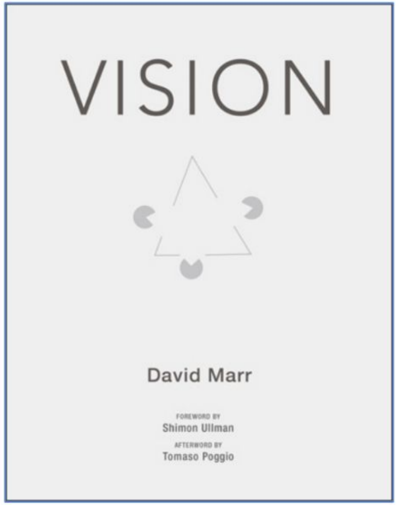
maximize:

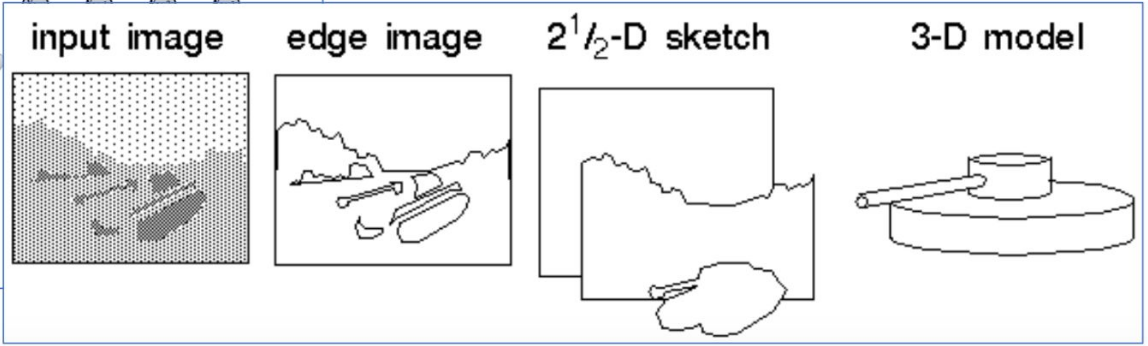
$$R_t = r_{t+1} + r_{t+2} + \cdots + r_T$$

Algorthmic
What representations can implement such computations?
How does the choice of representations determine the algorithm?



Implementational
How can such a system be built in hardware?
How can neurons carry out the computations?





How to be sure we have correct Image Analysis?

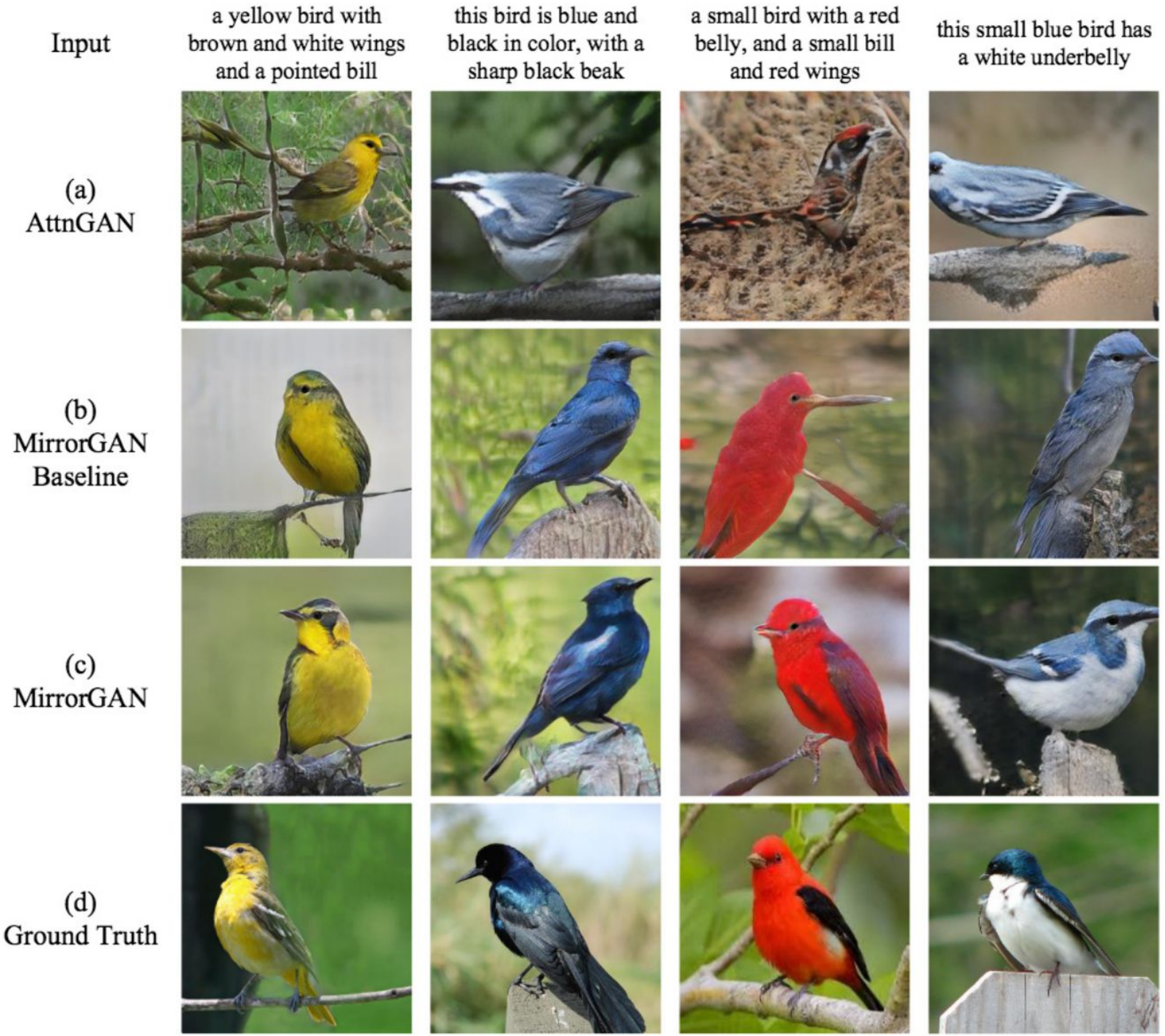
How to be sure we have correct Image Analysis?

What I cannot create, I do not understand. -- Richard Feynman

Synthesizing an Image: Text to Image

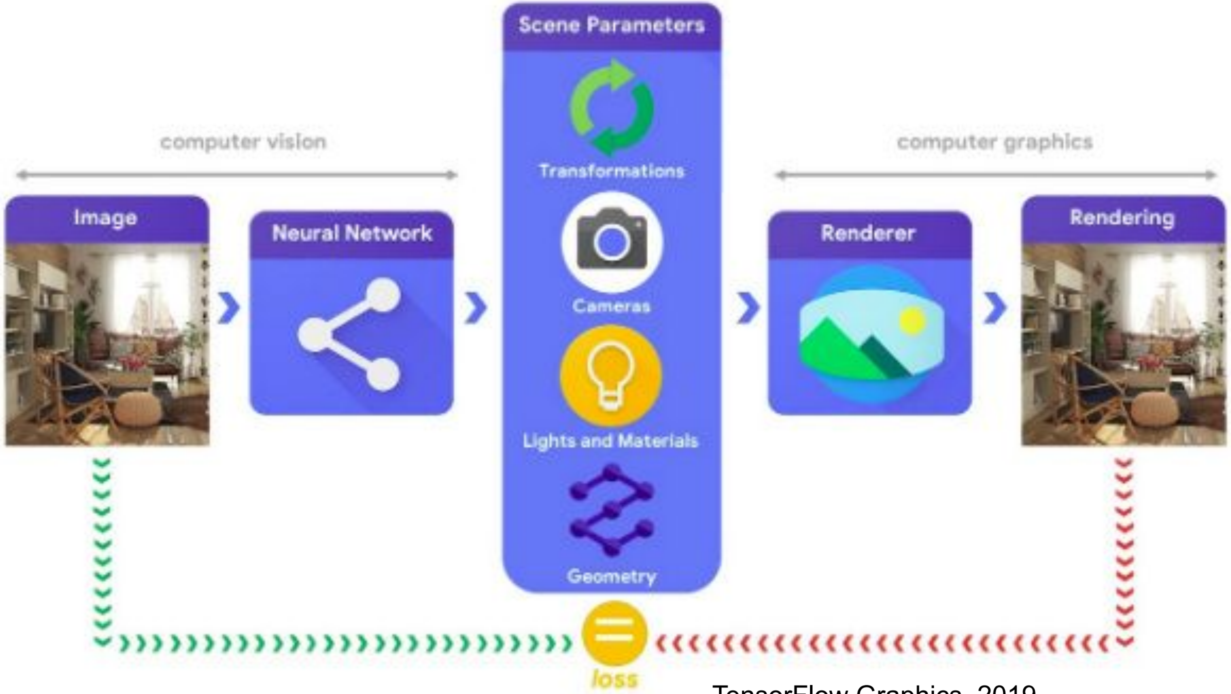
MirrorGAN '19

But... A picture is worth a thousand words.



Closing the loop: Computer Vision meets Computer Graphics

- Analysis by Synthesis (*a long standing idea*)
- The three R's of computer vision: Recognition, reconstruction and reorganization (2016)



TensorFlow Graphics, 2019

Analysis by Synthesis: 3D Object Recognition by Object Reconstruction (CVPR '14)

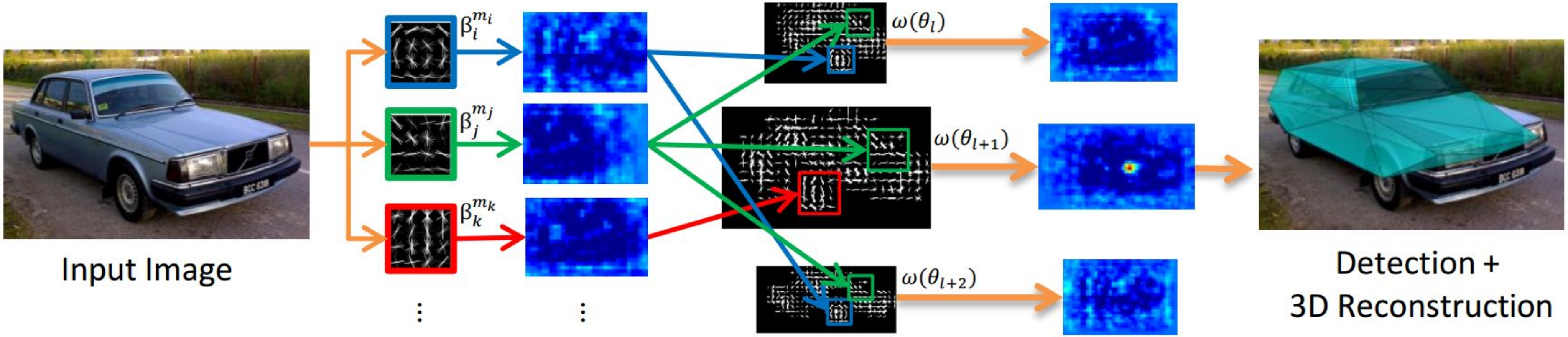
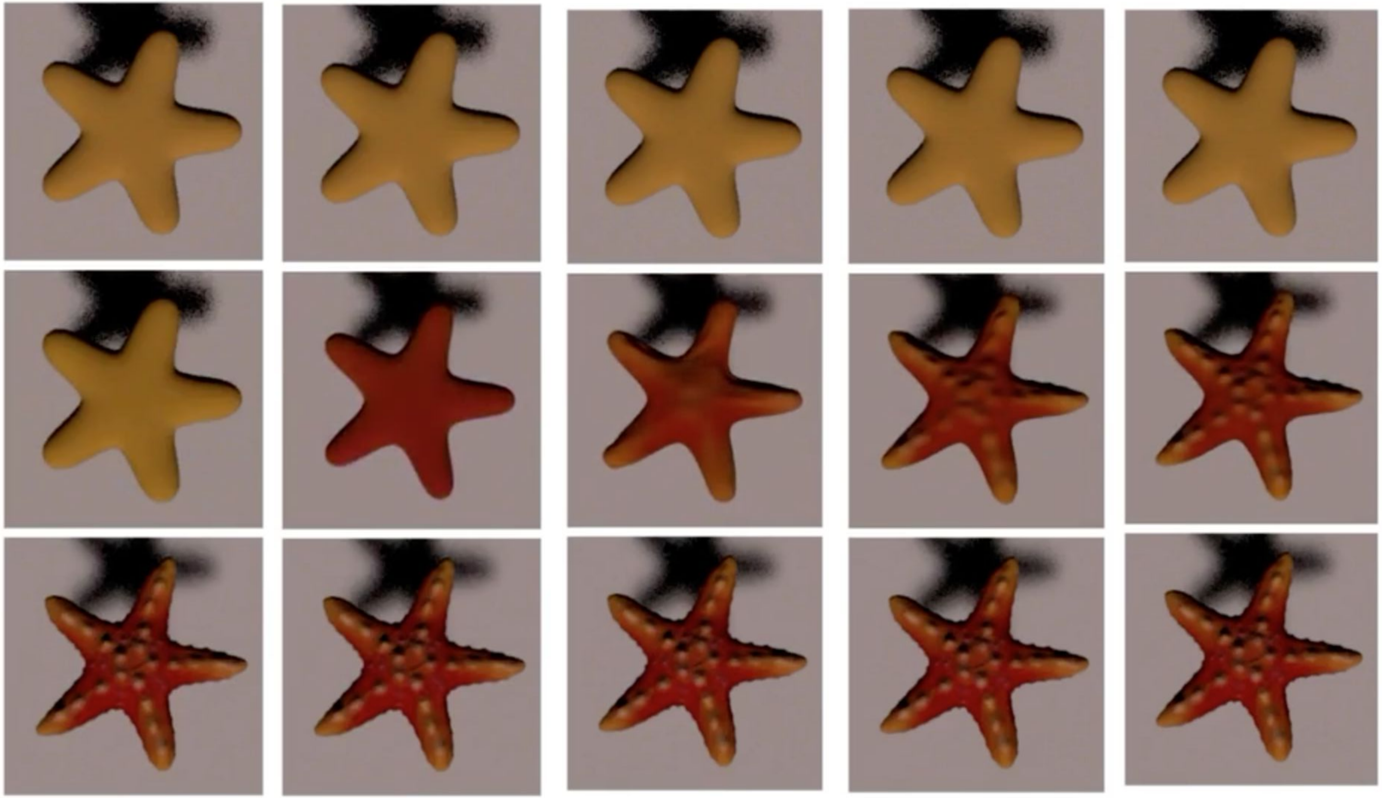


Figure 4. We search through a large collection of templates (with shared parts) by first caching part responses, and then looking up response values to score each template.

Reparameterizing Discontinuous Integrands for Differentiable Rendering (2019)

- Differentiable approximation of surface displacement and texture

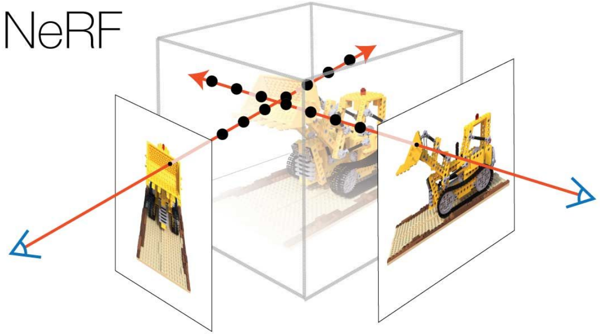


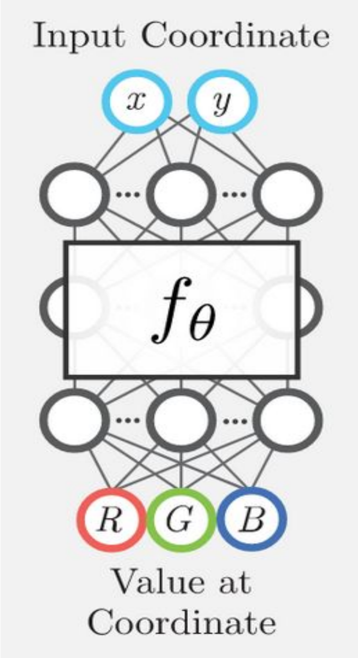
Overview

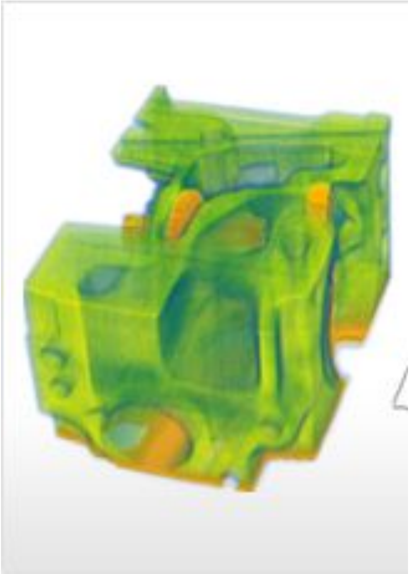
- Computer Vision meets Natural Language Processing
  - Vision Transformers: Detection, Classification and Segmentation
  - Semi- and Self-Supervised Learning: Vision-Language models
- Computer Vision meets Computer Graphics
  - Differential Rendering and Analysis by Synthesis
  - **Neural Radiance Field, with applications to SLAM, AR/VR**

What makes NeRF?

- Coordinate NN
    - a new ***compact*** representation of Tensor , allusive to non-linear PCA
- Volumetric Rendering

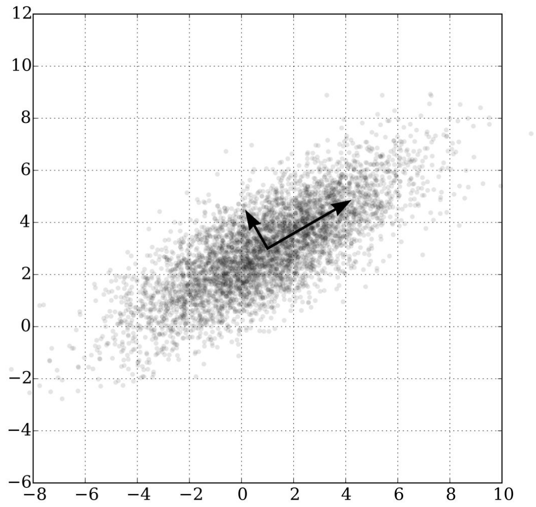




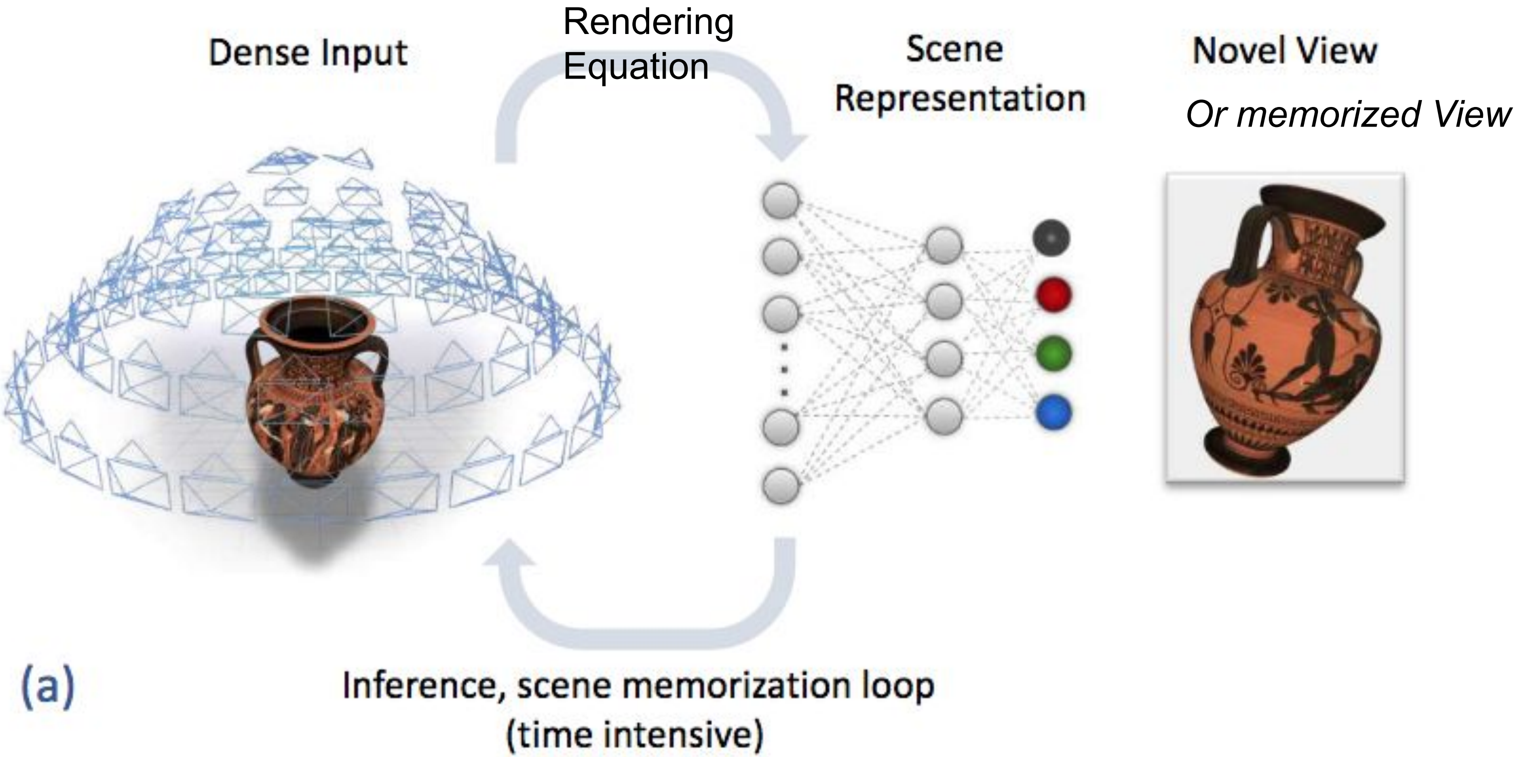


Principal Component Analysis and EigenFace

- PCA linearly factorizes data into linear combination of (a few) components

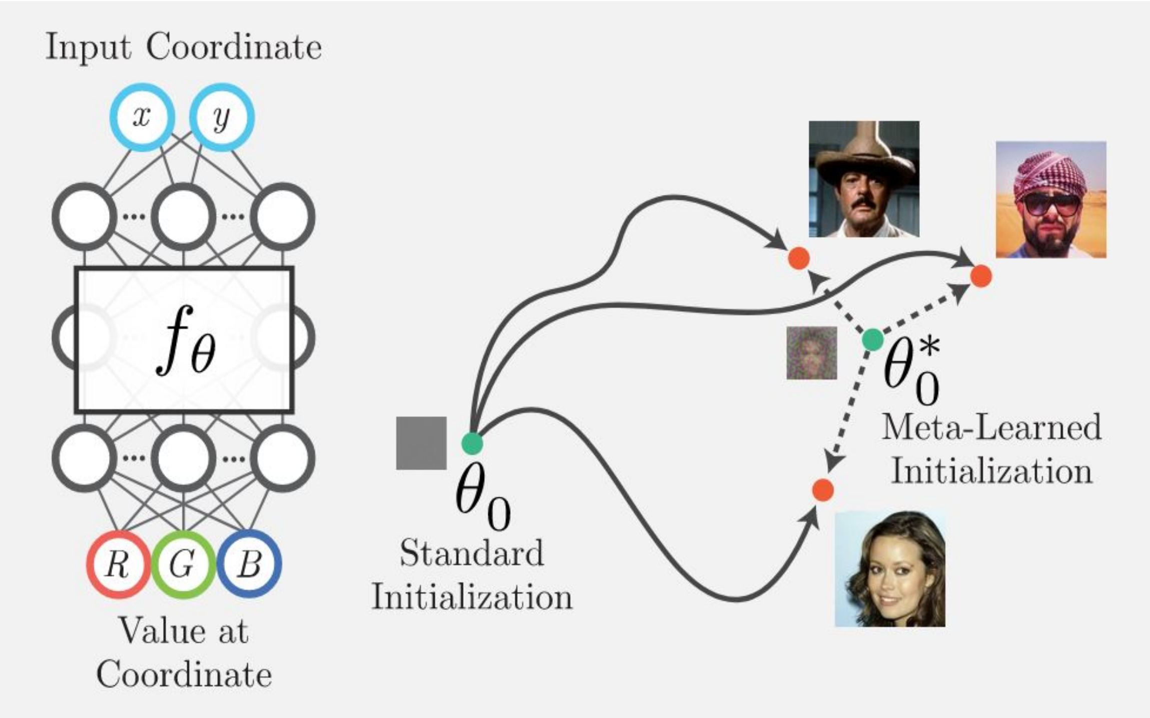


NeRF: Neural Radiance Fields (2003.08934)



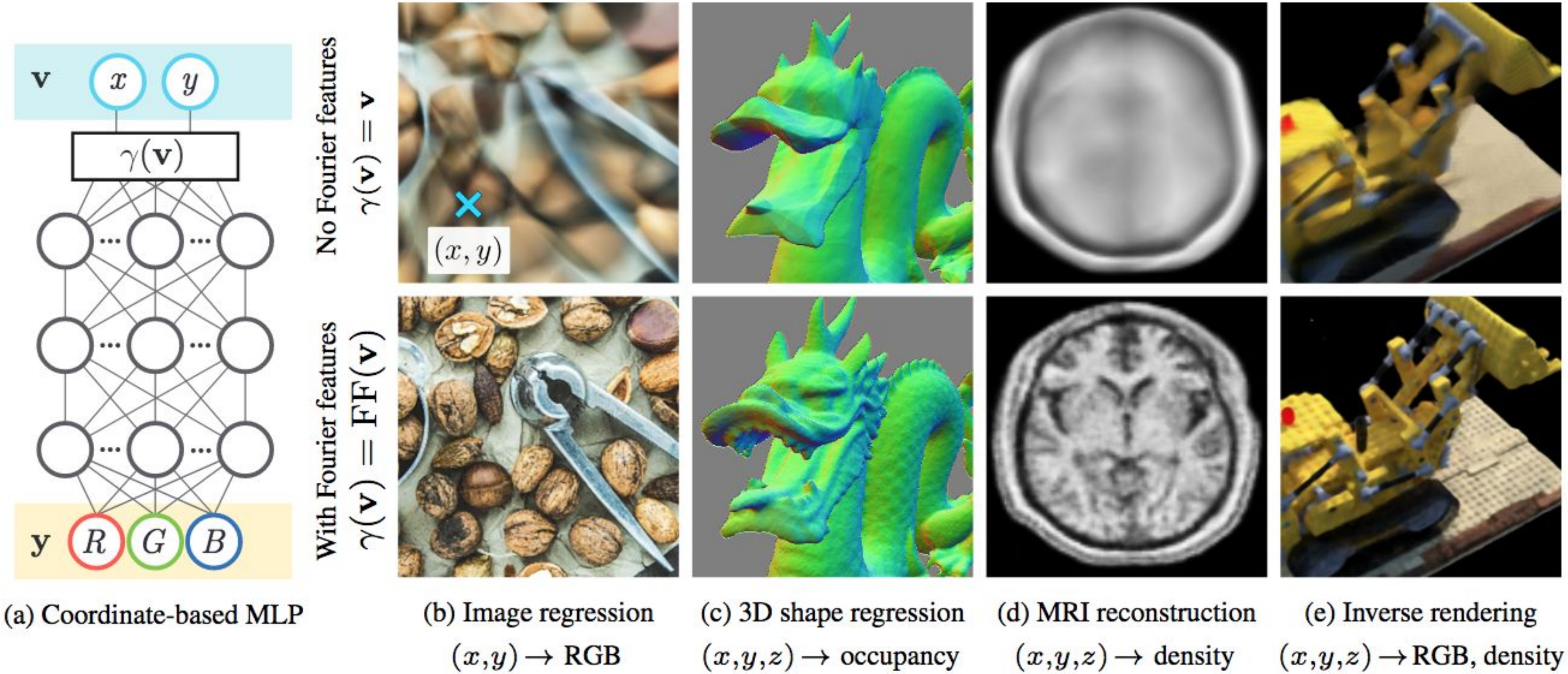
Scene Representation in NeRF: Coordinate MLP

- Inputs are just coordinates (allusive to Positional Encoding in Transformers)
- (x, y): image
- (x, y, z): occupancy
- (x, y, z, θ, ϕ) ray-tracing
- (x, y, z, θ, ϕ, t) spatial-temporal video

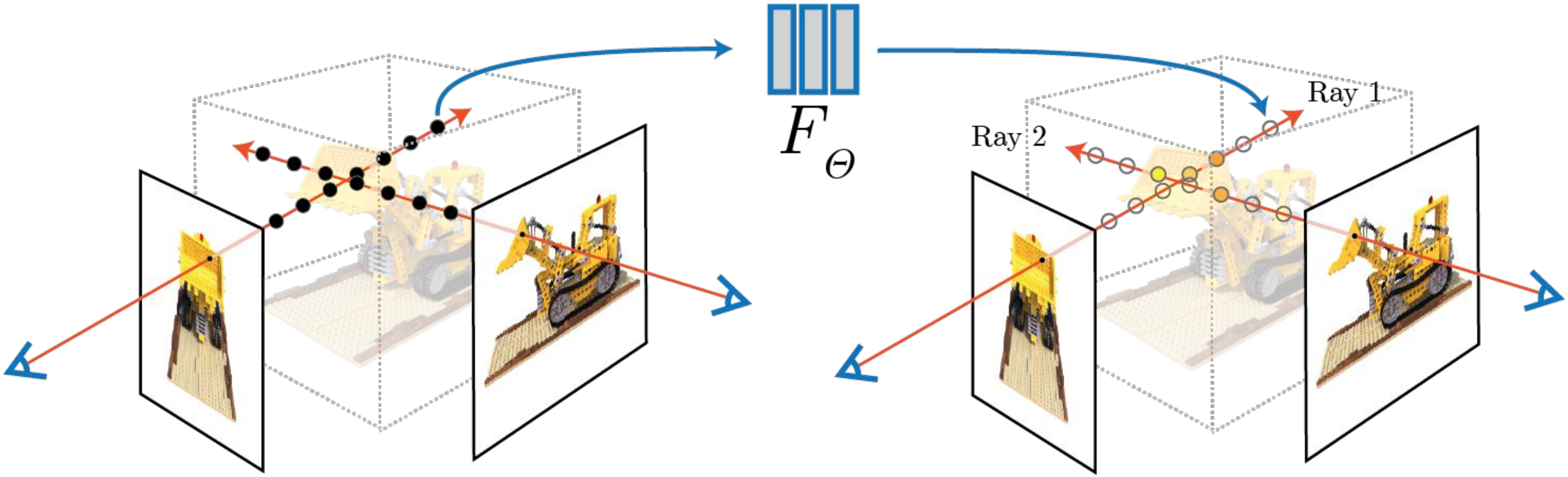


Coordinate MLP

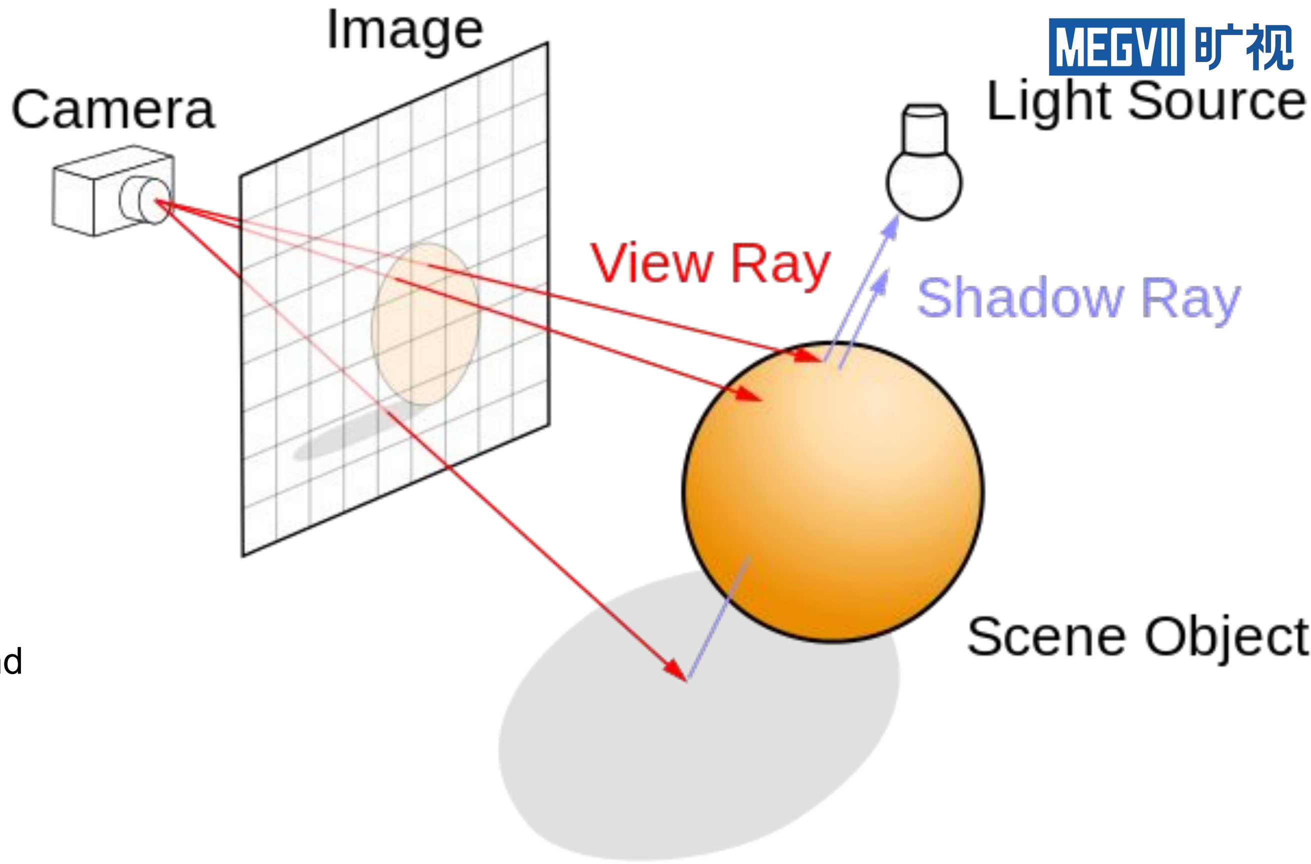
- Uses Fourier Features for modeling high-frequency details



NeRF is simpler:
Simplifying Rendering Equation using Ray Marching with NN as SDF



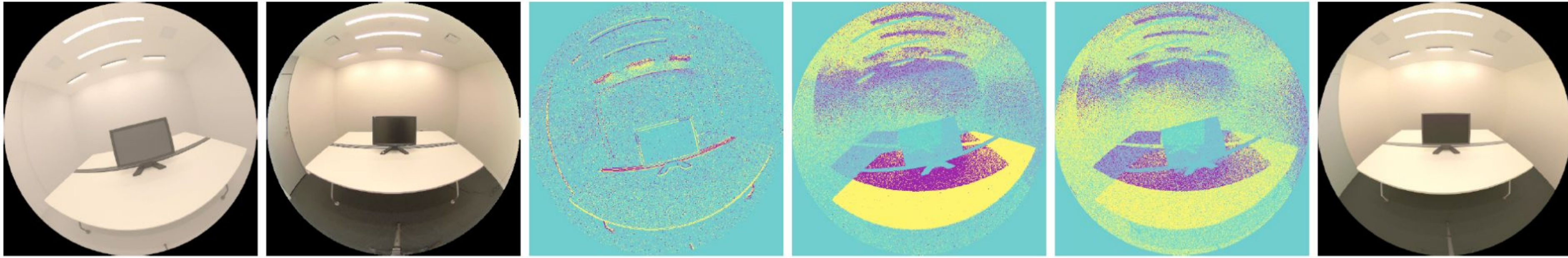
Ray Tracing



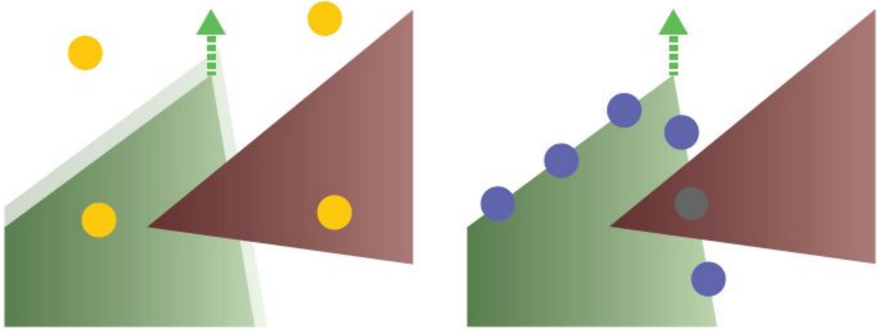
a . Drawing by Monte Carlo Ray Tracing, with lights bouncing in the scene. Not easy to get proper gradients.

b. How to represent BSDF and make it differentiable?

Differentiable Monte Carlo Ray Tracing through Edge Sampling (2018)



(a) initial guess (b) real photograph (c) camera gradient (per-pixel contribution) (d) table albedo gradient (per-pixel contribution) (e) light gradient (per-pixel contribution) (f) our fitted result



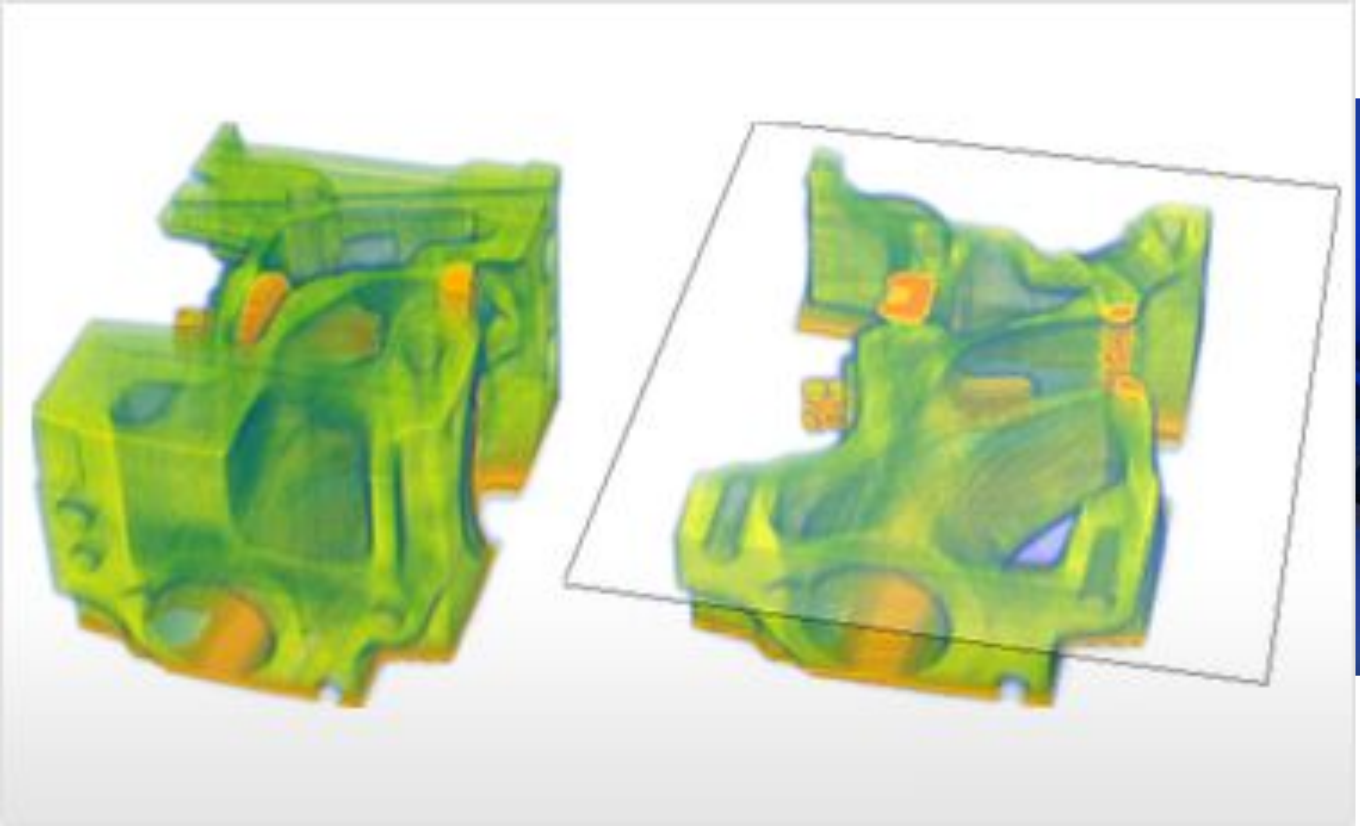
(a) area sampling (b) edge sampling

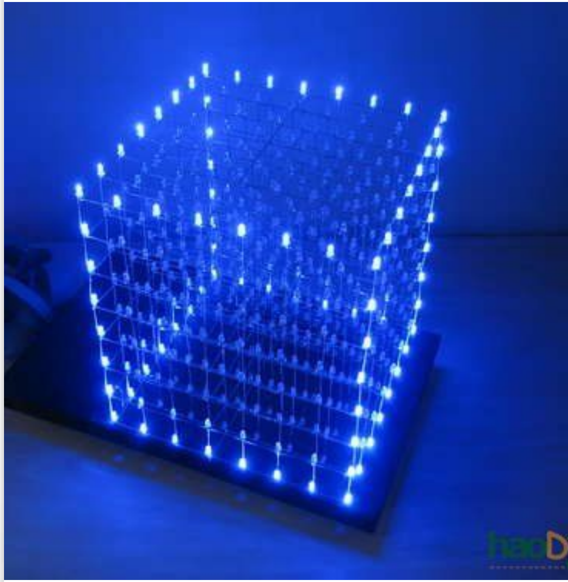
Ray Marching (instead of Ray Tracing in NeRF)

http://jamie-wong.com/2016/07/15/ray-marching-signed-distance-functions/

- In raytracing, the scene is typically defined in terms of explicit geometry: triangles, spheres, etc. To find the intersection between the view ray and the scene, we do a series of geometric intersection tests
- In raymarching, the entire scene is defined in terms of a signed distance function. To find the intersection between the view ray and the scene, we start at the camera, and move a point along the view ray, bit by bit, until the SDF evaluate to a negative number. We hit something.
    - If it’s not, we keep going up to some maximum number of steps along the ray.

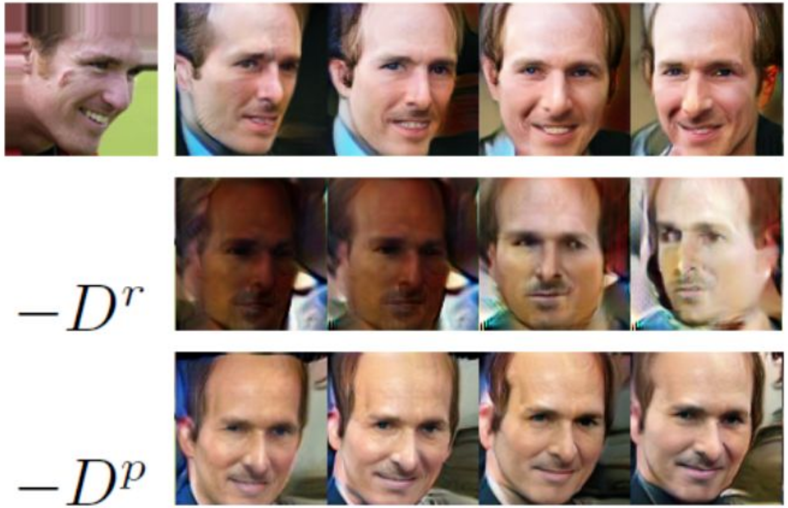
NeRF is simpler: Volume Rendering, smoother

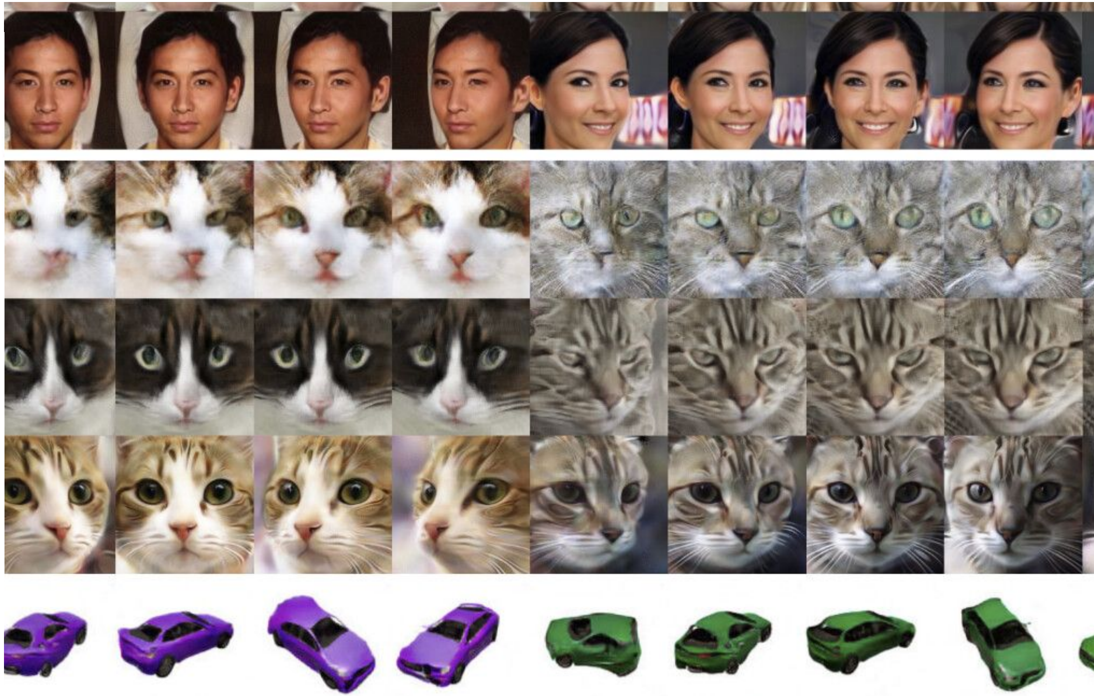




Faithfulness of rendering equation helps preserves identity!

- DR-GAN (1705.11136) vs. pi-GAN (2012.00926, NeRF-based)



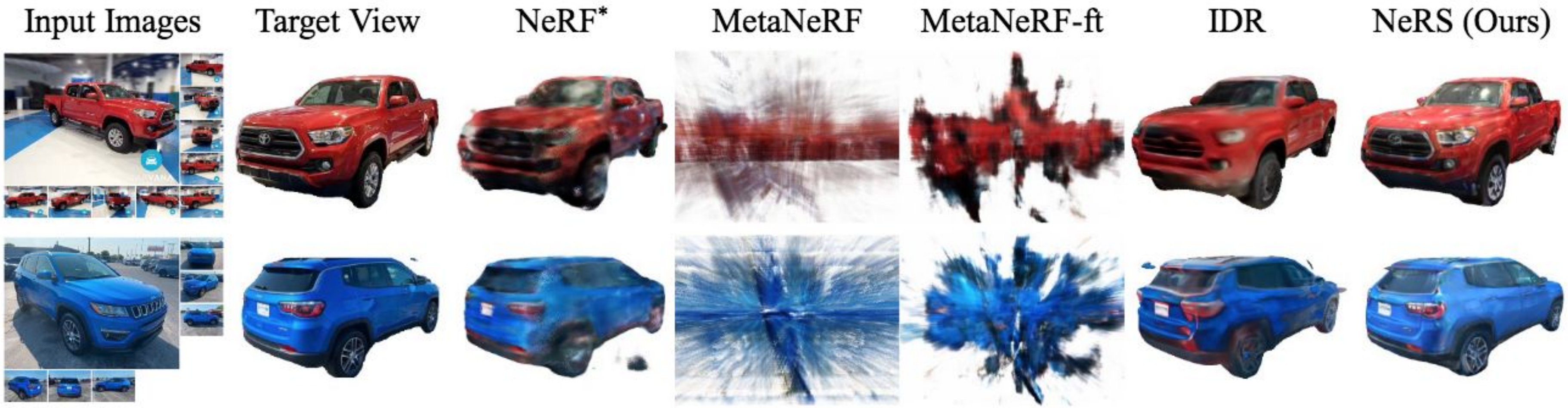


Applications of NeRF (with generalizations)

- **3D modeling from Real-world Imagings**
  - From a few Images: NeRS
  - Dynamic Scenes: D-NeRF, Nerfies
  - From Free-Viewpoint Video
- Image Synthesis
  - 3D-aware synthesis: pi-GAN
  - From MineCraft world: GANcraft
- 3D models as Differentiable Volumetric Representation
  - for SLAM: iMAP
  - for Robotics

NeRS: Neural Reflectance Surfaces for Sparse-View 3D Reconstruction in the Wild 2110.07604

- input: several (8-16) unposed images of the same instance
- output: a textured 3D reconstruction along with the illumination parameters.



Dynamic Scene: D-NeRF

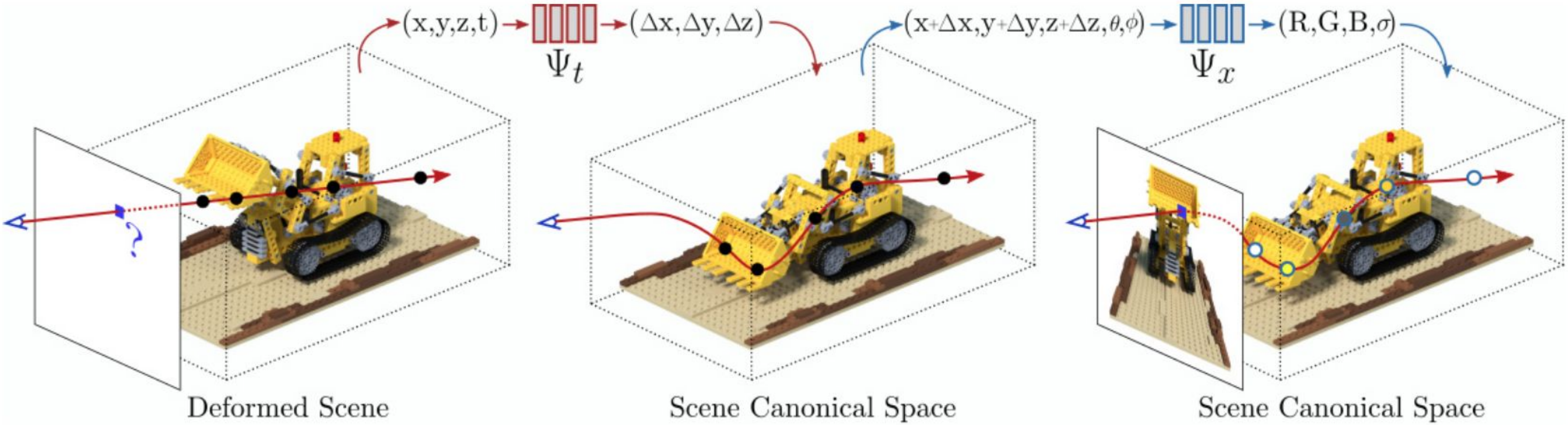


Figure 3: **D-NeRF Model**. The proposed architecture consists of two main blocks: a deformation network Ψ_t mapping all scene deformations to a common canonical configuration; and a canonical network Ψ_x regressing volume density and view-dependent RGB color from every camera ray.

Dynamic Scene: Nerfies

- Can handle Glassy and moving objects

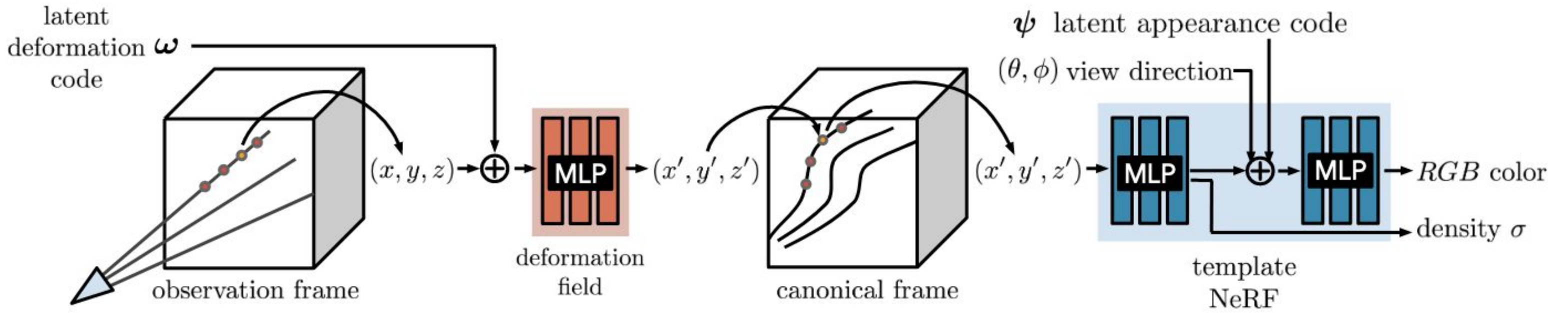


Figure 2: We associate a latent deformation code (ω) and an appearance code (ψ) to each image. We trace the camera rays in the observation frame and transform samples along the ray to the canonical frame using a deformation field encoded as an MLP that is conditioned on the deformation code ω. We query the template NeRF [39] using the transformed sample (x', y', z'), viewing direction (θ, ϕ) and appearance code ψ as inputs to the MLP and integrate samples along the ray following NeRF.

Space-time Neural Irradiance Fields for Free-Viewpoint Video (2011.12950)

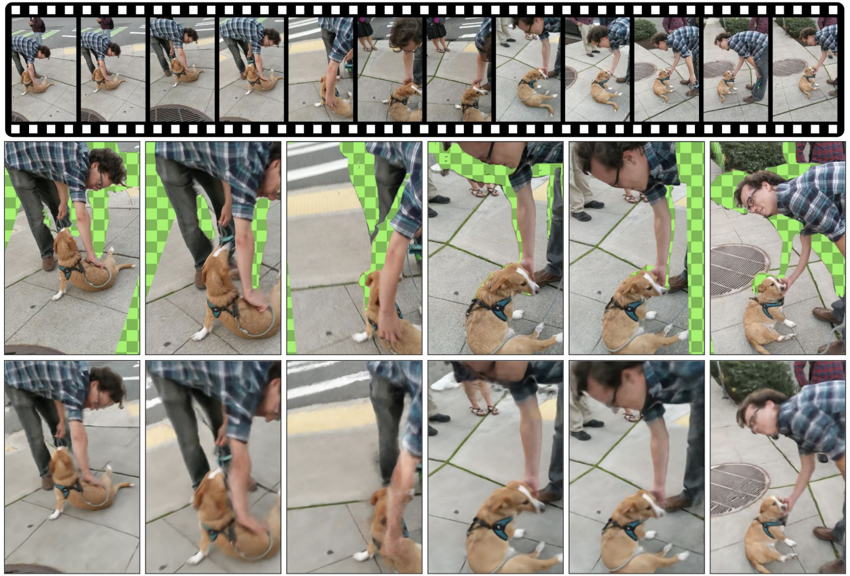


Figure 1. Our method takes a *single* casually captured video as input and learns a space-time neural irradiance field. (*Top*) Sample frames from the input video. (*Middle*) Novel view images rendered from textured meshes constructed from depth maps. (*Bottom*) Our results rendered from the proposed space-time neural irradiance field.

We make a simple assumption on *unobserved* spaces: every part of the world should stay static unless observed not as such. Enforcing this assumption prevents the part of spaces that are not observed from going entirely unconstrained. Our static scene constraint encourages the shared color and volume density at the same spatial location $\mathbf{x}$ between two distinct times t and t':

$$\mathcal{L}_{\text{static}} = \sum_{(\mathbf{x},t)\in\mathcal{X}} \left\| F(\mathbf{x},t) - F(\mathbf{x},t') \right\|_2^2, \tag{8}$$

Applications of NeRF (with generalizations)

- 3D modeling from Real-world Imagings
  - From a few Images: NeRS
  - Dynamic Scenes: D-NeRF, Nerfies
  - From Free-Viewpoint Video
- **Image Synthesis**
  - 3D-aware synthesis: pi-GAN
  - From MineCraft world: GANcraft
- 3D models as Differentiable Volumetric Representation
  - for SLAM: iMAP
  - for Robotics

GANcraft: Unsupervised 3D Neural Rendering of Minecraft Worlds (2104.07659)

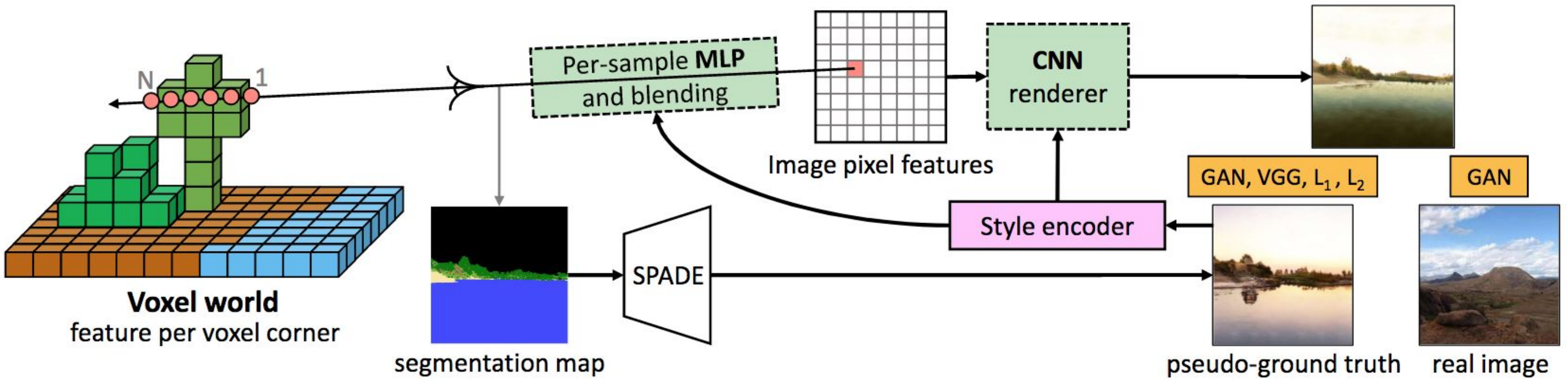


Figure 3: **Overview of GANcraft.** Given an input voxel world with segmentation labels, we first assign features to every voxel corner. For arbitrarily sampled camera viewpoints, we obtain the trilinearly interpolated voxel features at the point of ray-voxel intersections, process with an MLP, and blend the output features to obtain the image pixel features. These features are fed to an image-space CNN renderer. Both the MLP and the CNN are conditioned on the style code of the pseudo-ground truth for the chosen camera view. Our method is trained with an adversarial loss with real images, and a combination of adversarial, pixel-wise, and VGG perceptual losses on the pseudo-ground truths. After training, we can render the world in a photorealistic manner, controlling the style of the output images by conditioning on an input style code or image.

Applications of NeRF (with generalizations)

- 3D modeling from Real-world Imagings
  - From a few Images: NeRS
  - Dynamic Scenes: D-NeRF, Nerfies
  - From Free-Viewpoint Video
- Image Synthesis
  - 3D-aware synthesis: pi-GAN
  - From MineCraft world: GANcraft
- **3D models as Differentiable Volumetric Representation**
  - for SLAM: iMAP
  - for Robotics

NeRF-GTO: Using a Neural Radiance Field to Grasp Transparent Objects (2021)

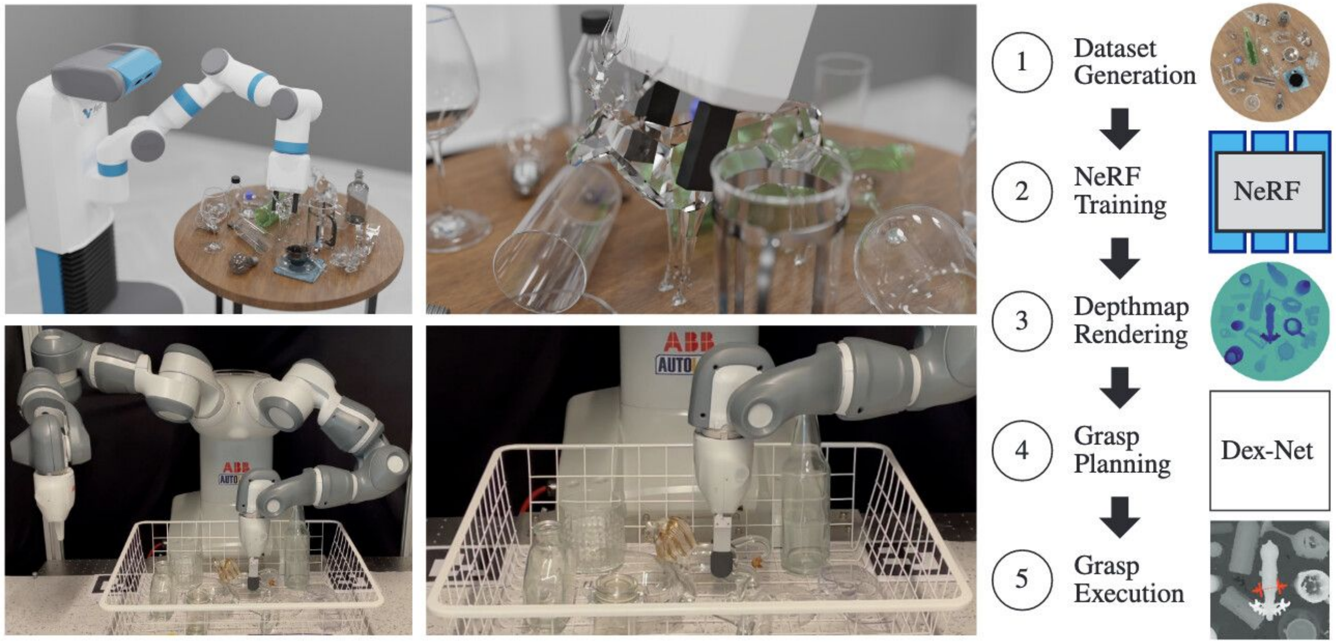


Figure 1: **Using NeRF to grasp transparent objects** Given a scene with transparent objects (left column), we the pipeline on the right to compute grasps (middle column). The top row shows NeRF-GTO working in a simulated scene while the bottom row shows it working in a physical scene.

Vision-Only Robot Navigation in a Neural Radiance World 2110.00168

- collision penalty (based on NeRF) is now soft
- control penalty for less jerky control

$$J(W) = \sum_{\tau=0}^{h} \left[\overbrace{\sum_{b_i \in \mathcal{B}} \rho(R_\tau b_i + \hat{\sigma}_\tau) s(b_i)}^{\text{collision penalty}} + \overbrace{u_\tau^T \Gamma u_\tau}^{\text{control penalty}} \right]$$

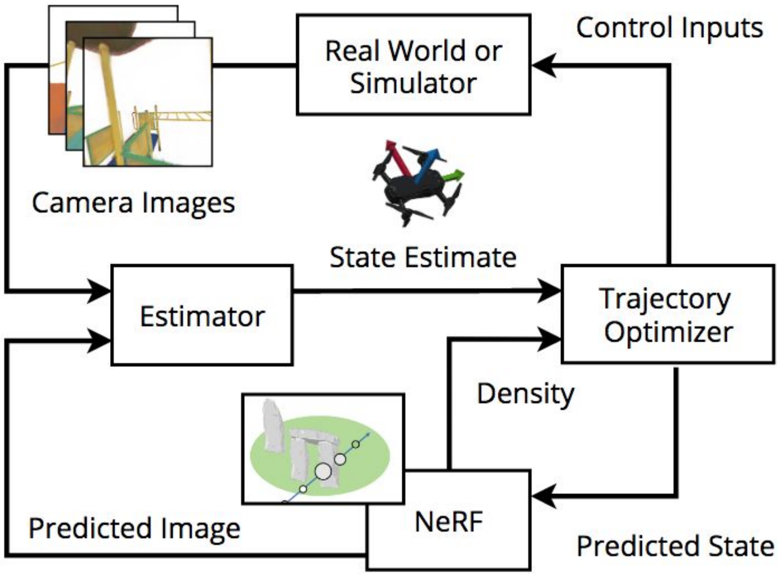


Fig. 3. Block diagram of the proposed pipeline. Our method consists of a trajectory optimizer and state estimator which use a NeRF representation of the environment for planning and localization. At each timestep, the planner optimizes a trajectory from the current mean state estimate which minimizes a NeRF-based collision metric. The robot then applies the first control action of this trajectory, and receives a noisy image from its onboard camera. Finally, the state estimator, using the NeRF as a nonlinear measurement model, uses this image to generate a posterior belief over the new state.

Non-conclusive Conclusions, as of *2021Q3*

- **New Models** like Transformers
  - frees many CV tasks of bells and whistles
  - creates a unified foundation for CV and NLP
- **Large Models:** large pre-trained Vision and Language models benefiting downstream tasks
- **Easier 3D**: NeRF is expected to further simplify 3D Vision Infrastructure
  - Easier 3D model acquisition
  - Easier Image Synthesizing
  - Differential rendering is now accessible to everyone

References (Vision Transformer + Vision-language model)

- Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).
- Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).
- Radford, Alec, et al. "Language models are unsupervised multitask learners." *OpenAI blog* 1.8 (2019): 9.
- Brown, Tom B., et al. "Language models are few-shot learners." *arXiv preprint arXiv:2005.14165* (2020).
- Carion, Nicolas, et al. "End-to-end object detection with transformers." *European Conference on Computer Vision*. Springer, Cham, 2020.
- Chen, Qiang, et al. "You only look one-level feature." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.
- Ge, Zheng, et al. "OTA: Optimal Transport Assignment for Object Detection." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.
- Wang, Jianfeng, et al. "End-to-end object detection with fully convolutional network." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.
- Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." *arXiv preprint arXiv:2010.11929* (2020).
- Touvron, Hugo, et al. "Training data-efficient image transformers & distillation through attention." *International Conference on Machine Learning*. PMLR, 2021.
- Liu, Ze, et al. "Swin transformer: Hierarchical vision transformer using shifted windows." *arXiv preprint arXiv:2103.14030* (2021).
- Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." *International conference on machine learning*. PMLR, 2020.
- Qi, Hang, Matthew Brown, and David G. Lowe. "Low-shot learning with imprinted weights." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.
- Pathak, Deepak, et al. "Context encoders: Feature learning by inpainting." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
- Chen, Xinlei, et al. "Improved baselines with momentum contrastive learning." *arXiv preprint arXiv:2003.04297* (2020).
- Grill, Jean-Bastien, et al. "Bootstrap your own latent: A new approach to self-supervised learning." *arXiv preprint arXiv:2006.07733* (2020).
- Chen, Mark, et al. "Generative pretraining from pixels." *International Conference on Machine Learning*. PMLR, 2020.
- Radford, Alec, et al. "Learning transferable visual models from natural language supervision." *arXiv preprint arXiv:2103.00020* (2021).
- Huo, Yuqi, et al. "WenLan: Bridging vision and language by large-scale multi-modal pre-training." *arXiv preprint arXiv:2103.06561* (2021).

References (NeRF)

- Qiao, Tingting, et al. "Mirrorgan: Learning text-to-image generation by redescription." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.
- Hejrati, Mohsen, and Deva Ramanan. "Analysis by synthesis: 3d object recognition by object reconstruction." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014.
- Loubet, Guillaume, Nicolas Holzschuch, and Wenzel Jakob. "Reparameterizing discontinuous integrands for differentiable rendering." *ACM Transactions on Graphics (TOG)* 38.6 (2019): 1-14.
- Mildenhall, Ben, et al. "Nerf: Representing scenes as neural radiance fields for view synthesis." *European conference on computer vision*. Springer, Cham, 2020.
- Li, Tzu-Mao, et al. "Differentiable monte carlo ray tracing through edge sampling." *ACM Transactions on Graphics (TOG)* 37.6 (2018): 1-11
- Tran, Luan, Xi Yin, and Xiaoming Liu. "Disentangled representation learning gan for pose-invariant face recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
- Chan, Eric R., et al. "pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.
- Zhang, Jason Y., et al. "NeRS: Neural Reflectance Surfaces for Sparse-view 3D Reconstruction in the Wild." *arXiv preprint arXiv:2110.07604* (2021).
- Pumarola, Albert, et al. "D-nerf: Neural radiance fields for dynamic scenes." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.
- Park, Keunhong, et al. "Nerfies: Deformable neural radiance fields." *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021.
- Xian, Wenqi, et al. "Space-time neural irradiance fields for free-viewpoint video." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.
- Hao, Zekun, et al. "GANcraft: Unsupervised 3D Neural Rendering of Minecraft Worlds." *arXiv preprint arXiv:2104.07659* (2021).
- Ichnowski, Jeffrey, et al. "NeRF-GTO: Using a Neural Radiance Field to Grasp Transparent Objects." *5th Annual Conference on Robot Learning*. 2021.
- Adamkiewicz, Michal, et al. "Vision-Only Robot Navigation in a Neural Radiance World." *arXiv preprint arXiv:2110.00168* (2021).

Backup after this slide