Multimodal Transformers

Tripp Deep Learning F23

TODAY'S GOAL

By the end of the class, you should be familiar with some adaptations of transformers to tasks that involve non-text data.

Summary

1. Transformers can perform a wide range of vision tasks
2. Transformers can be multimodal

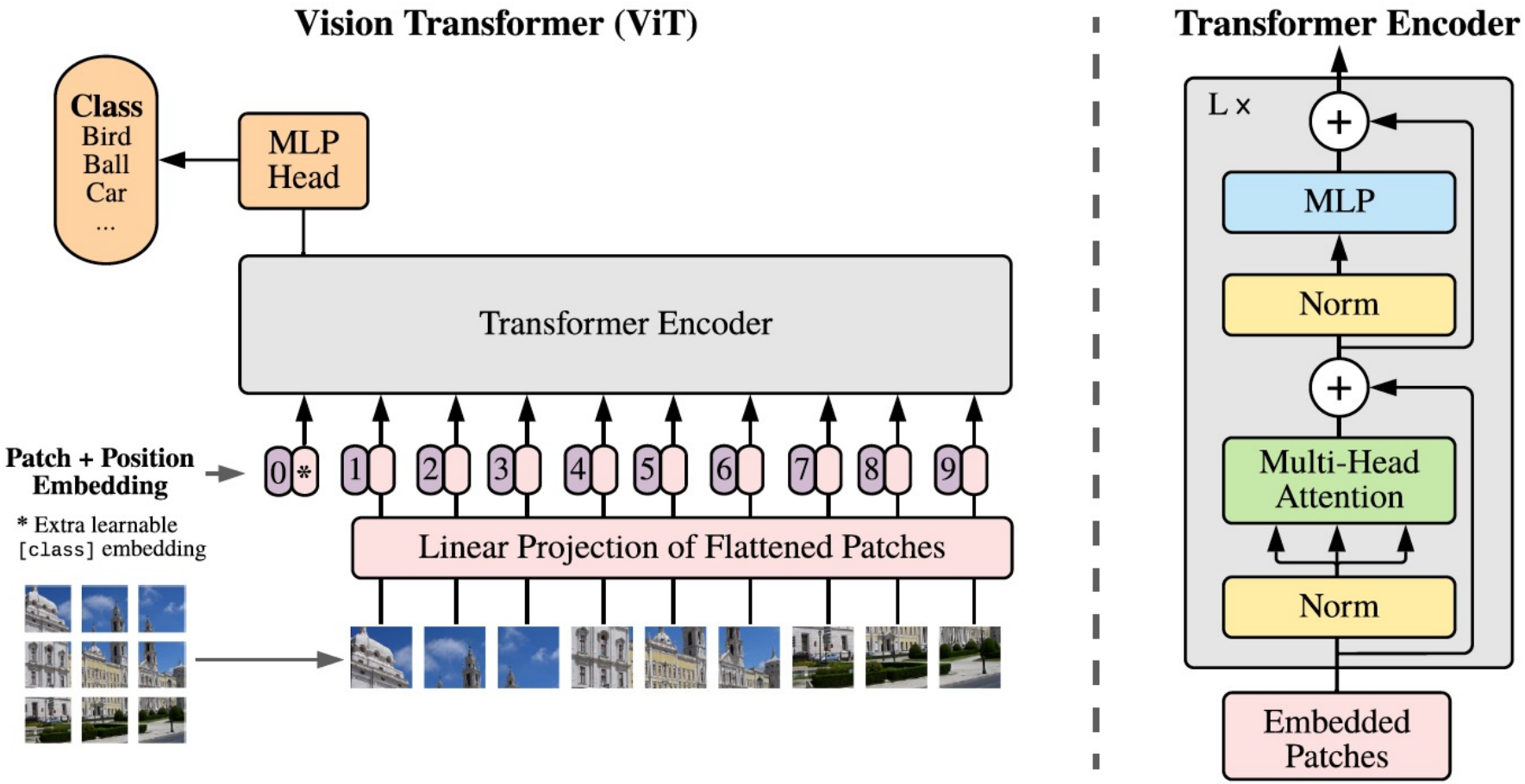
TRANSFORMERS CAN PERFORM A WIDE RANGE OF VISION TASKS

Transformers in vision

- Transformers have been applied to many different vision tasks, including object recognition, object detection, action recognition, segmentation, and pose estimation
- Transformers have also been applied to many tasks that combine vision and language including image captioning, video captioning, visual question answering, and image generation from text
- We will only discuss a few examples, but if you are interested in more, Khan et al. (2022) provides a comprehensive review

Vision Transformer

- Dosovitskiy et al. (2020) introduced the simple and effective approach of applying a transformer encoder to image patches of pixels

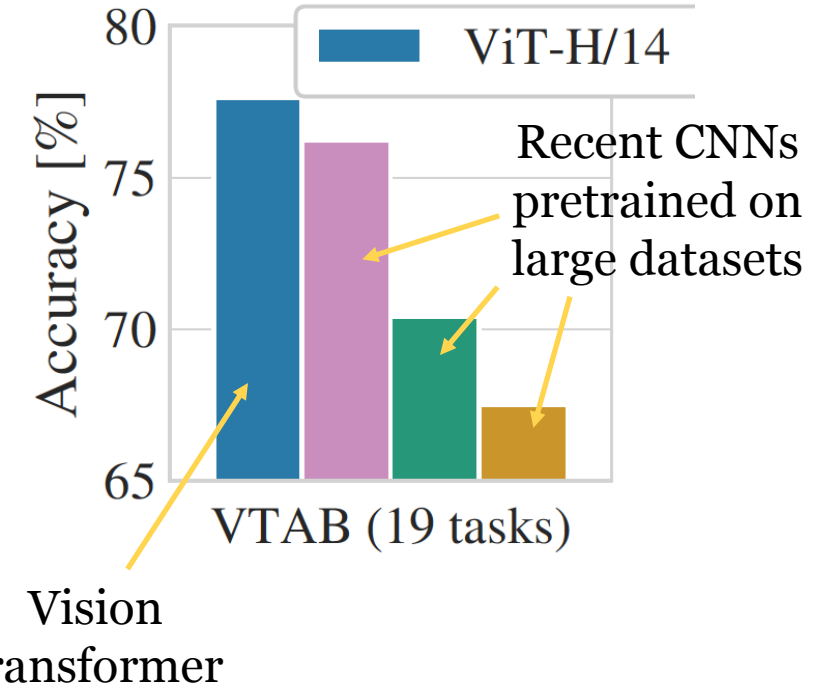


UNIVERSITY OF WATERLOO

Vision Transformer

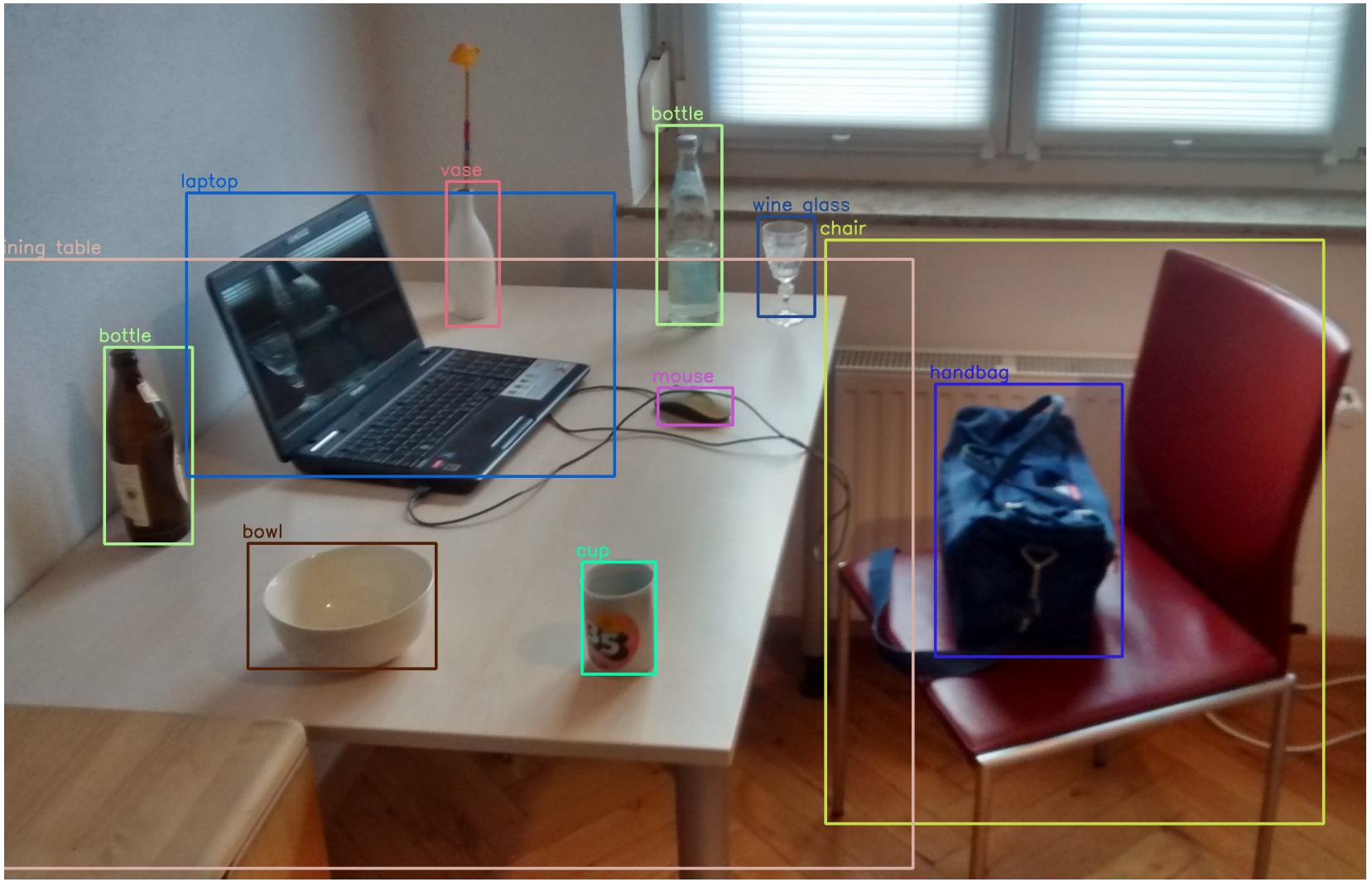
- Compared to convolutional networks, this approach:
  - Did not work as well when trained on ImageNet alone
  - Worked better when both were pretrained on much larger datasets including JFT300M; "… large scale training trumps inductive bias."
- The vision transformer performed well in transfer to other tasks with small datasets

Transfer performance on VTAB benchmark (vision tasks with small datasets)



Object detection

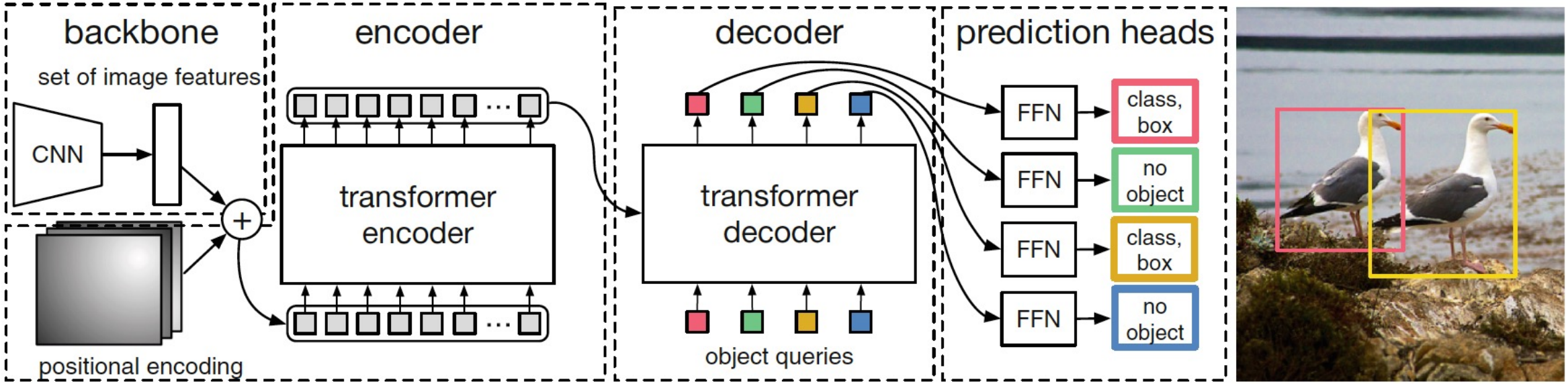
- Object detection consists of estimating bounding boxes and corresponding categories
- Previous successful models include Faster-RCNN and YOLO, both based on convolutional networks



https://commons.wikimedia.org/wiki/File:Detected-with-YOLO--Schreibtisch-mit-Objekten.jpg

Detection Transformer (DETR)

- DETR (Carion et al., 2020) applies a transformer to object detection
- Uses a ResNet to produce image patch embeddings that are fed into an encoder-decoder transformer
- Input to the decoder consists of learned "object query" vectors that are fixed at inference time



Detection Transformer (DETR)

- Each element of transformer output is sent to:
  - A linear layer and softmax to predict object category
  - A 3-layer MLP to predict bounding-box centre, height, and width
- There are more outputs than there normally are objects in a picture, so some objects are matched to a “no object” class
- The objects in an image are a set; they don’t have an intrinsic order and the loss shouldn’t depend on which element predicts which object

Detection Transformer (DETR)

- Let y_i be a vector that represents the class and bounding box of the i^{th} ground-truth object, and $\hat{y}_j$ be the predictions
- The list of y_i vectors has arbitrary ordering and is padded with no-object representations up to the size of the network output
- For each image, the ordering $j = \sigma(i)$ is found which minimizes the sum of losses that compare ground-truth y_i and $\hat{y}_{\sigma(i)}$
- This ordering is found with the Hungarian algorithm
  - This is a standard way to find a minimum-cost pairing between elements of two sets
  - It is out of our current scope, but if you are interested there is a good explanation here: https://brilliant.org/wiki/hungarian-matching/

Detection Transformer (DETR)

- Given an ordering $\sigma(i)$ that produces the best match between predictions and labels, the loss is the sum of two terms for each output:
  - Cross-entropy loss of the class prediction
  - Bounding-box loss, which is a linear combination of the absolute error of each element of the bounding-box vector and a term that depends on the intersection-over-union of the predicted and ground-truth bounding boxes

Semantic Segmentation

- The goal of semantic segmentation is to label each pixel of an image according to its object category
- Transformers applied to semantic segmentation include SegFormer (Xie et al, 2021) and Swin Transformer (Liu et al., 2021)

Fig. 1 Motorcycle racing image

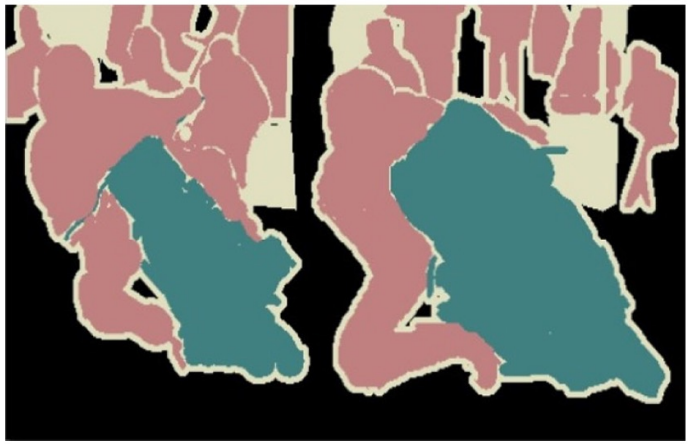
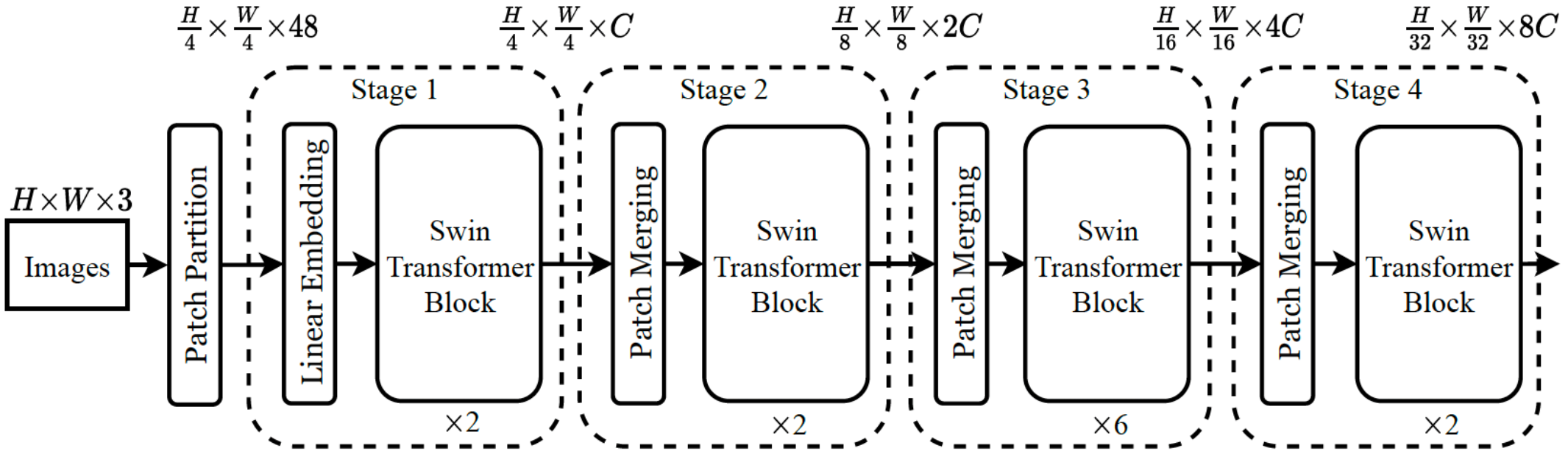


Fig. 2 Segmentation for motorcycle racing image

Guo et al. (2018)

Swin Transformer

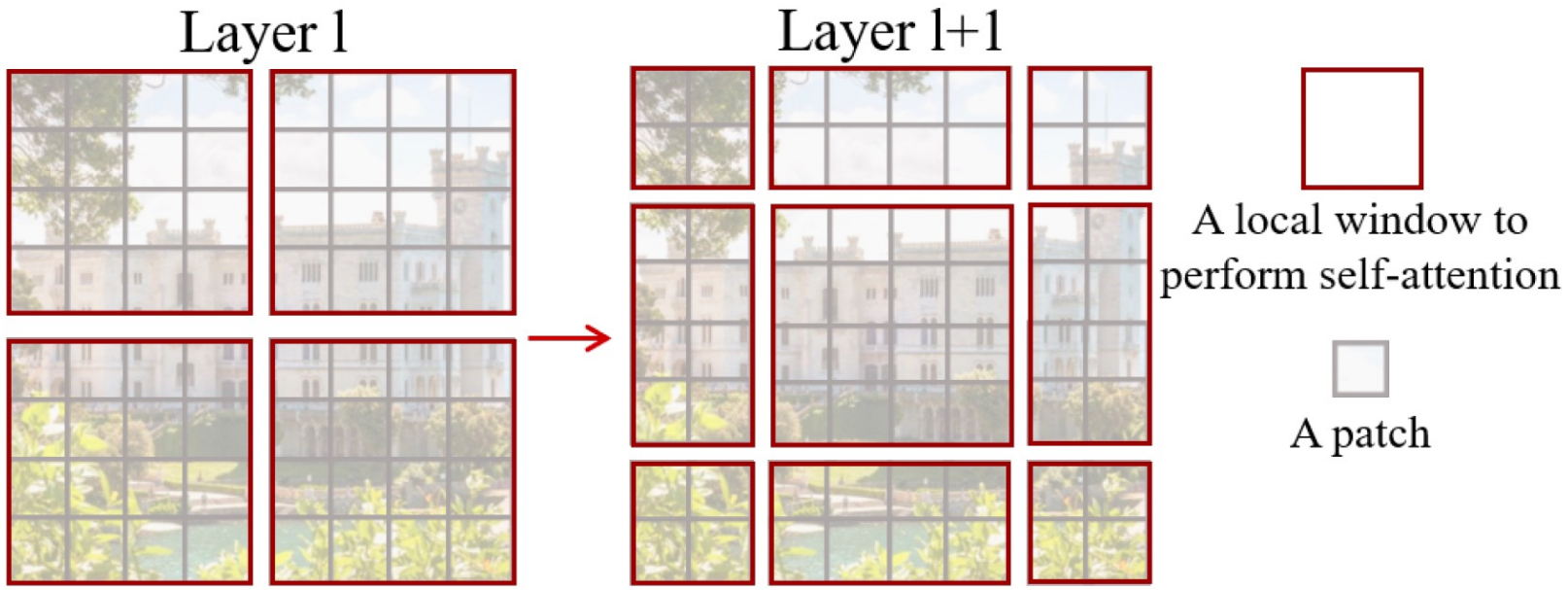
- Creates a convnet-like hierarchy of features at different resolutions
  - Merges 2x2 groups of patches by concatenation and linear map



Liu et al. (2021)

Swin Transformer

- In contrast with ViT, attention is only applied to patches within local windows (typically 7x7 patches)
  - This way attention computation doesn't scale with the square of #patches in the image
- The windows are shifted in alternating transformer blocks



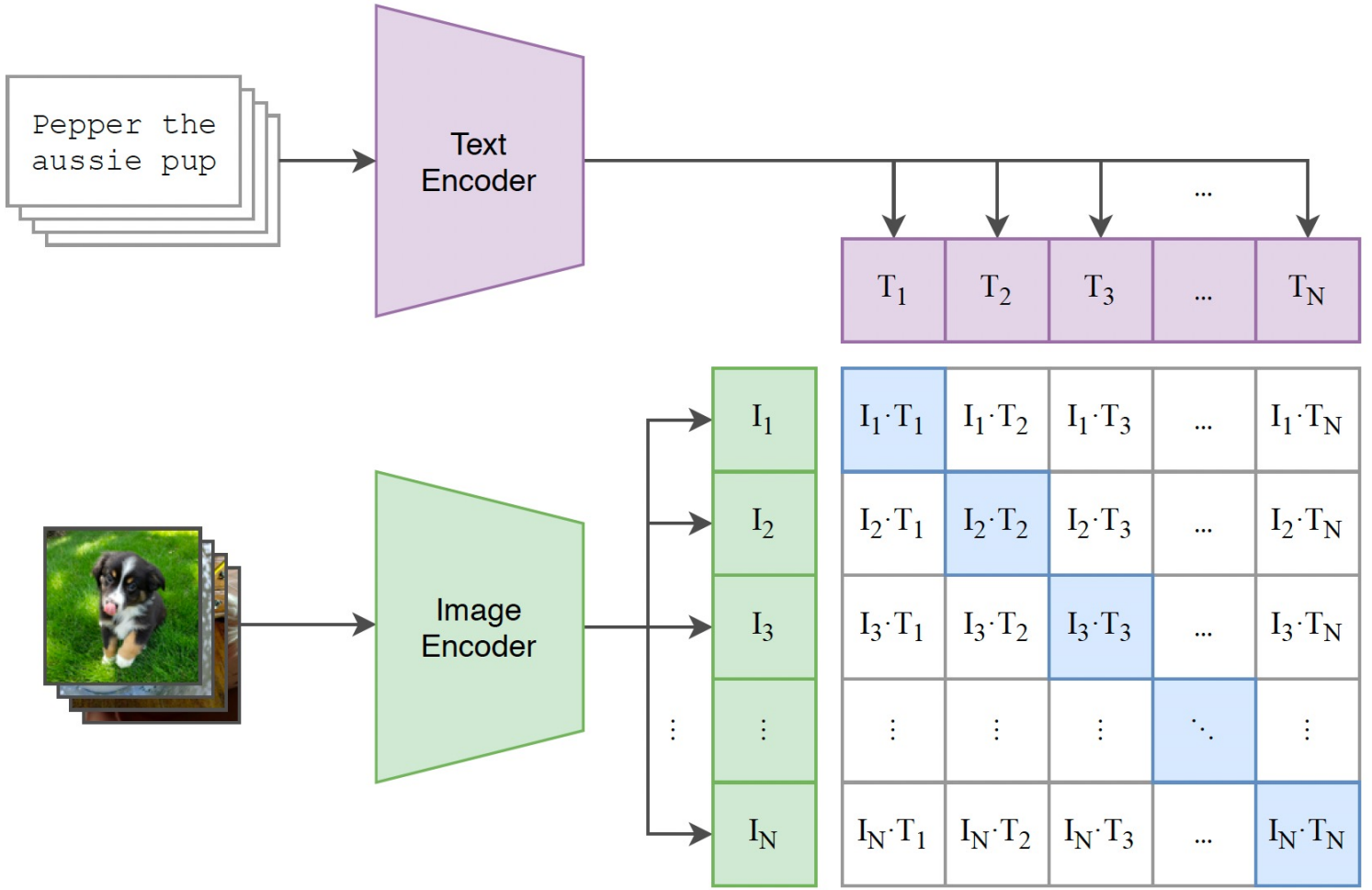
Swin Transformer

- Swin transformer is effective for image recognition and object detection (used as a backbone to suggest bounding boxes and object categories)
- Also applied to semantic segmentation as backbone for UPerNet (a network that predicts multiple image-level and pixel-level properties from different layers)

TRANSFORMERS CAN BE MULTIMODAL

CLIP (Contrastive Language-Image Pretraining)

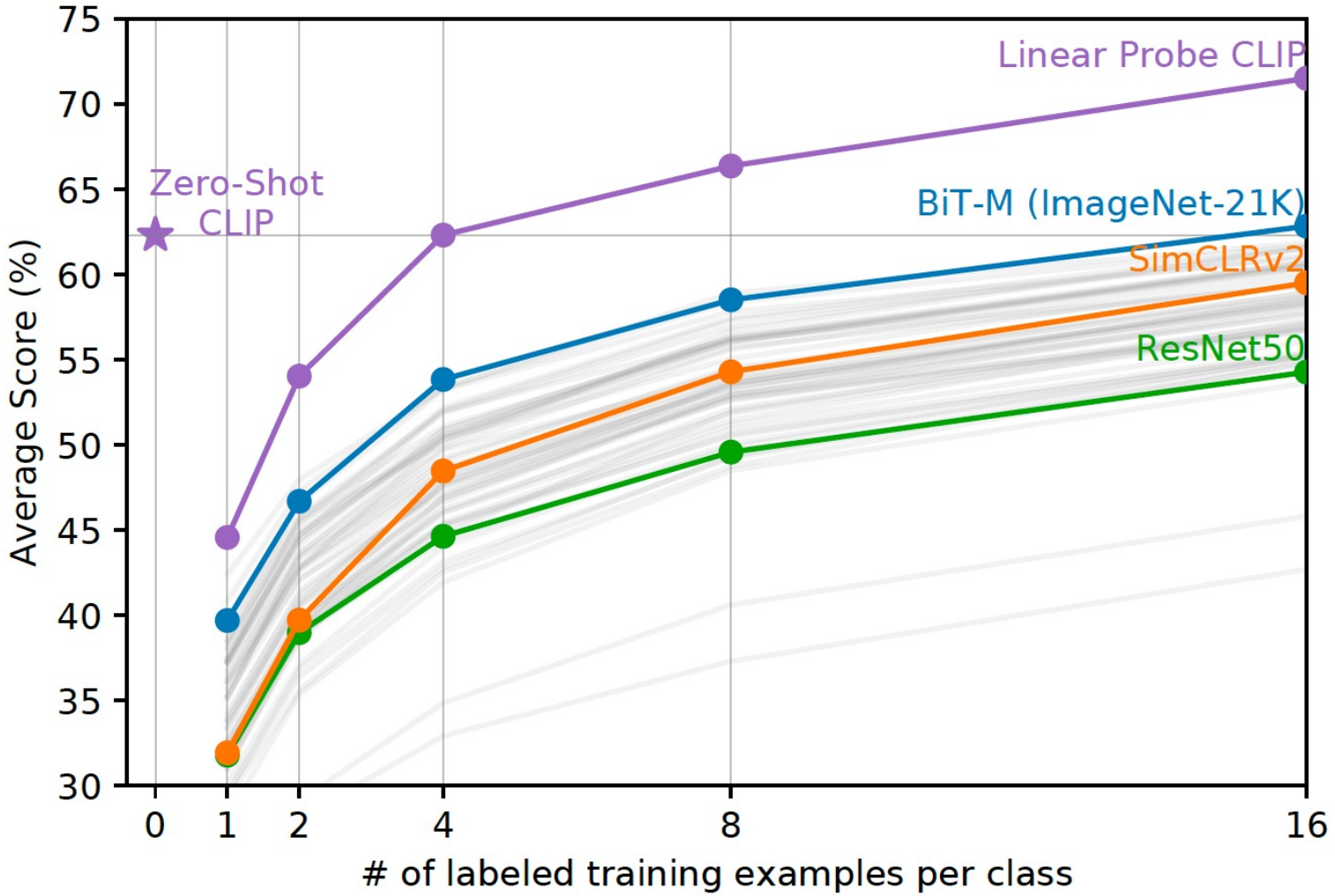
- Embeddings of images and captions jointly trained from 400M image-text pairs
- Within a batch, the model is symmetrically trained to predict which caption matches each image and vice versa



Radford et al. (2021)

CLIP

- Strong transfer learning performance in various downstream tasks (shown here is average performance across tasks)

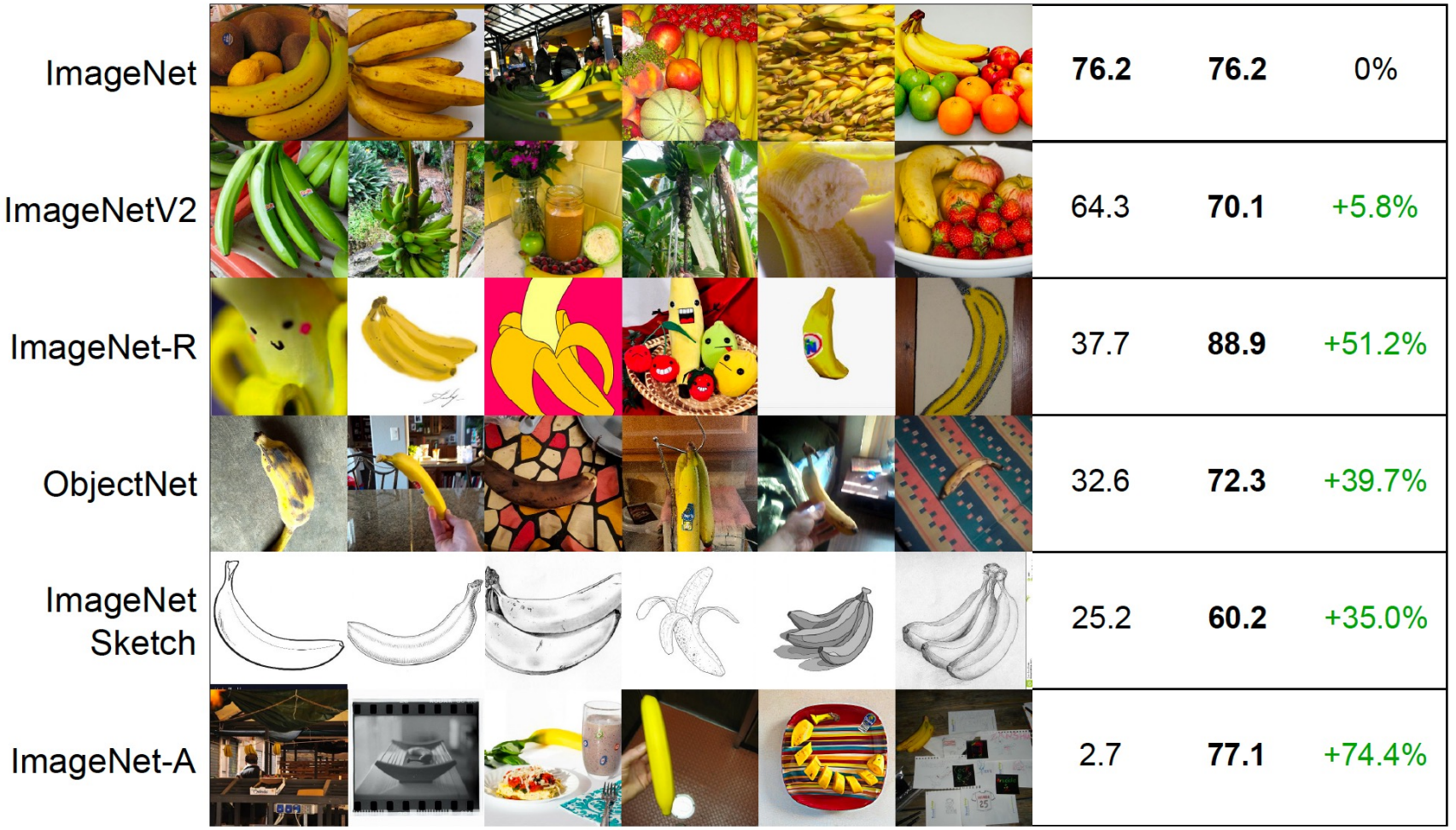


Radford et al. (2021)

UNIVERSITY OF WATERLOO

CLIP

- Relatively robust to distribution shift in the images (note it was trained on a wide variety of images)



Radford et al. (2021)

UNIVERSITY OF WATERLOO

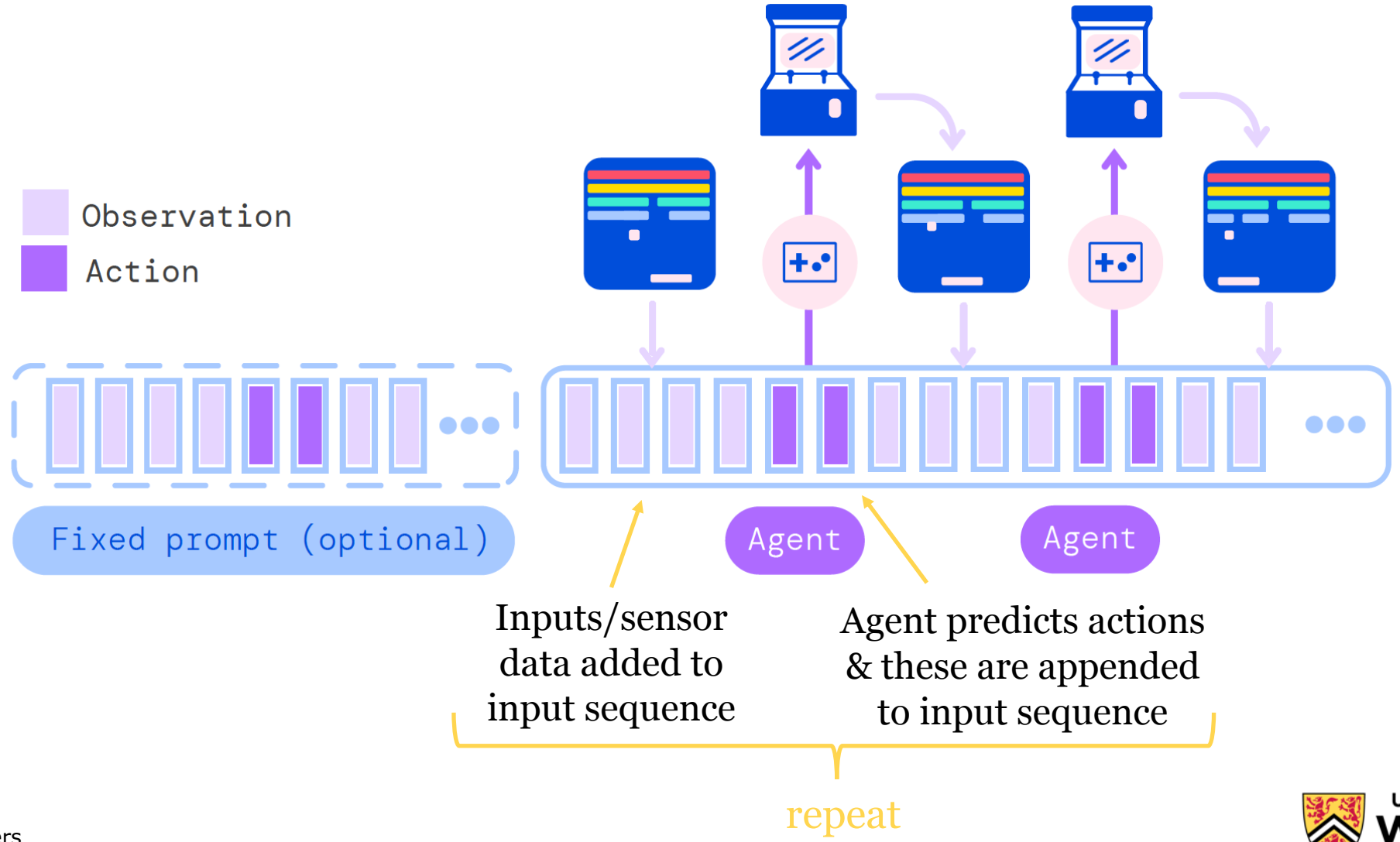
Gato

- Reed et al. (2022) trained a 1.2B-parameter decoder-only transformer on many tasks simultaneously
  - E.g., Atari games, image captioning, text prediction, visual question answering, and block stacking with a physical robotic arm
  - Training data from separate state-of-art RL agents for each task
- Tokenization
  - Text was tokenized in a standard way
  - Image patches were embedded with a ResNet block
  - Discrete actions such as Atari button presses were tokenized as integer values
  - Continuous actions and sensor data were normalized and quantized into a different integer range

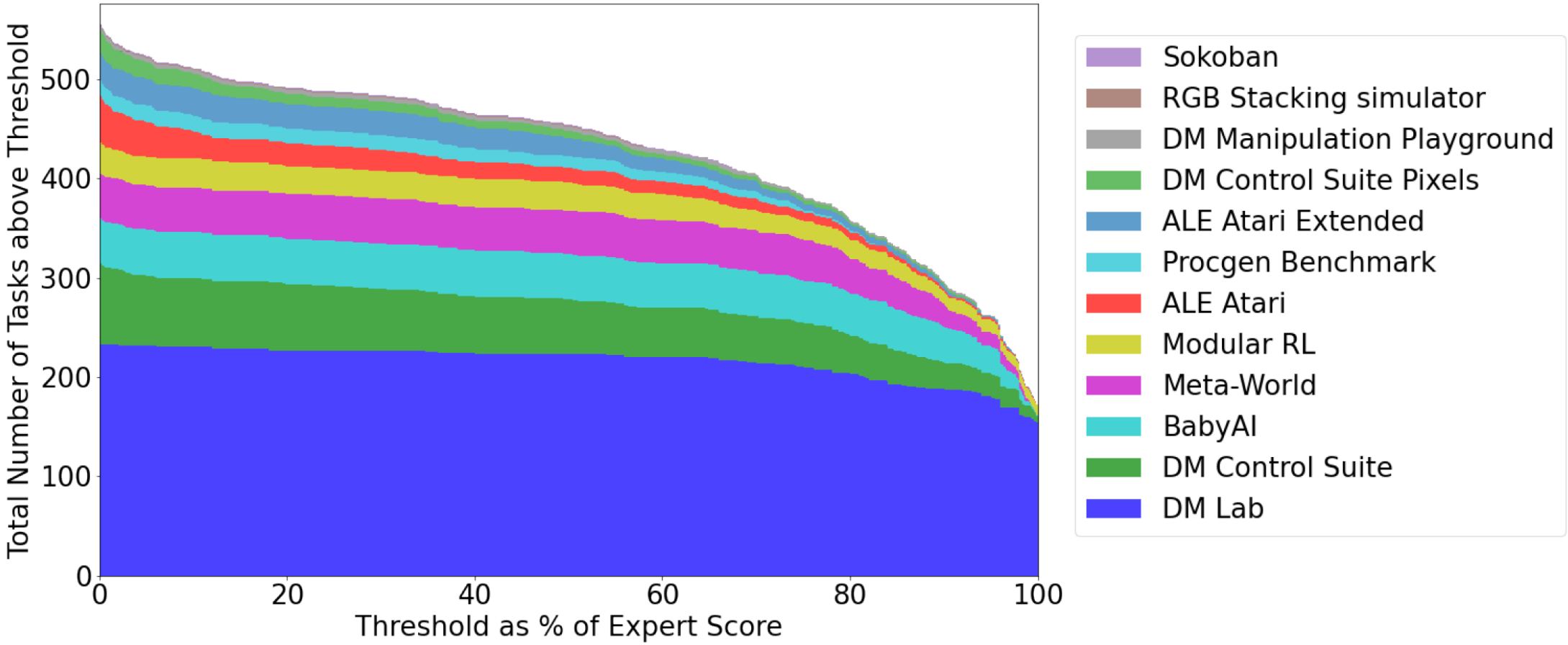
Gato: Prompt conditioning

- Similar inputs may occur in different tasks, so input alone may not make it clear to the agent what it is supposed to do; the task must be explicitly identified
- This was done by prompt conditioning:
  - A prompt consisted of an example demonstration of the desired task
  - A prompt sequence was prepended to 25% of training sequences
  - During evaluation, the agent was prompted with a successful demonstration of the desired task
  - Not the same as one-shot inference because the model was trained on the target tasks; just used to tell the model what to do

Gato: Performing a task

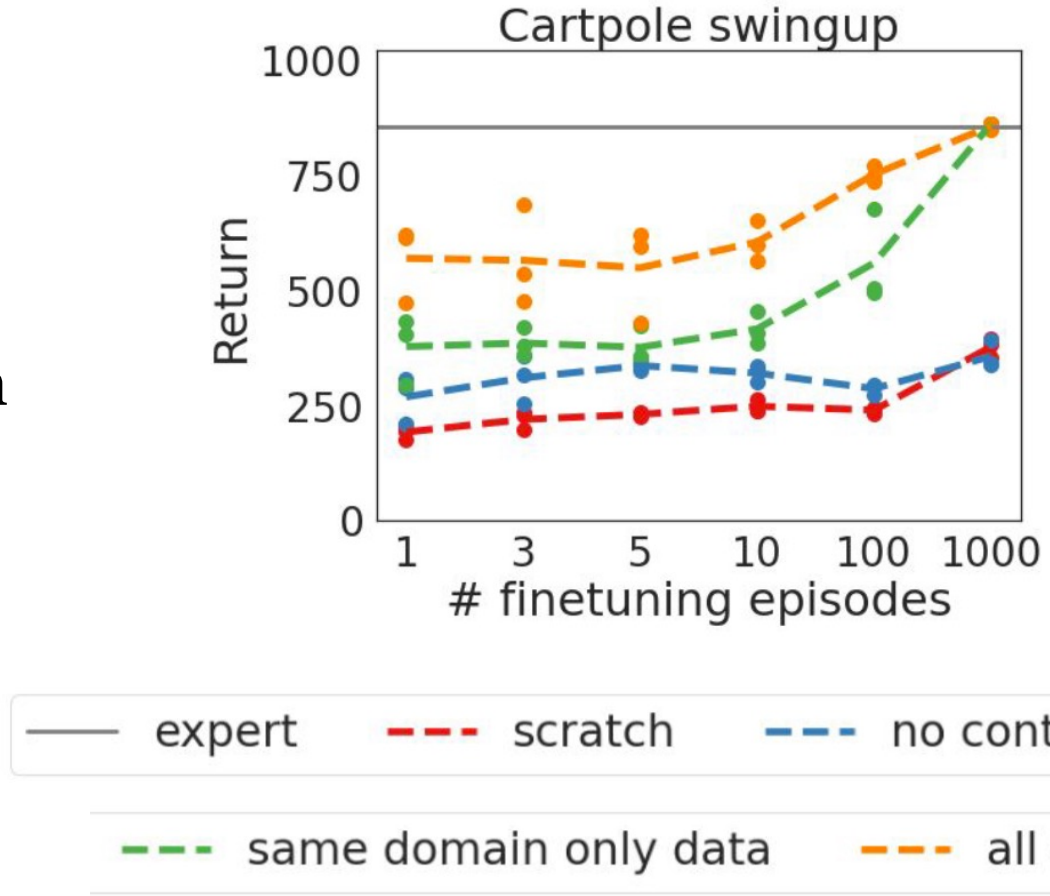


Gato: Performance on diverse tasks without fine tuning



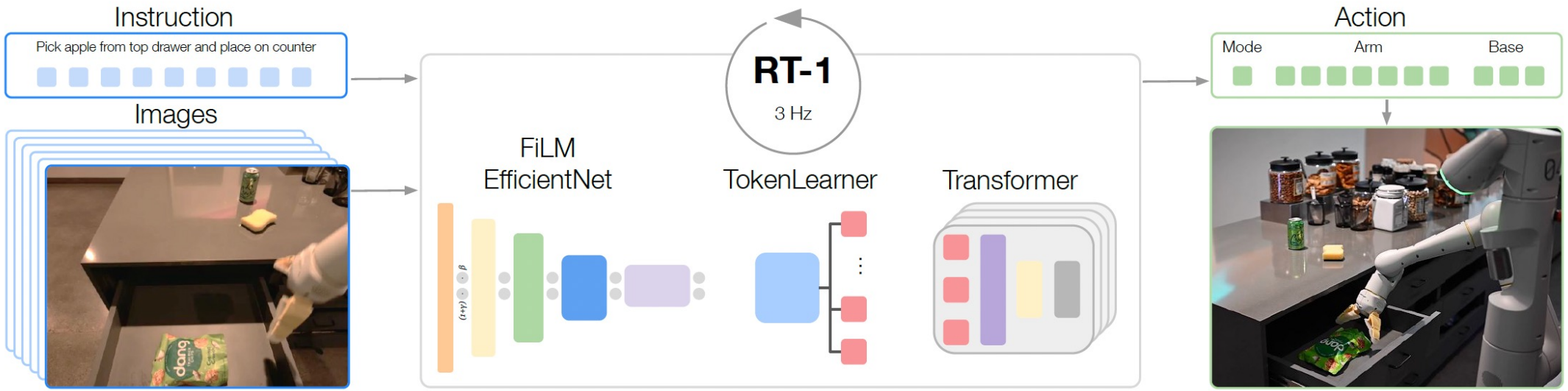
Gato: Rapid learning of held-out tasks

- After training smaller versions of Gato with large numbers of tasks, they tried fine tuning on held-out tasks with small numbers of examples
- For most of held-out tasks, pretraining on all tasks, or only tasks in the same domain, facilitated learning from few examples



RT-1 (Robotics Transformer 1)

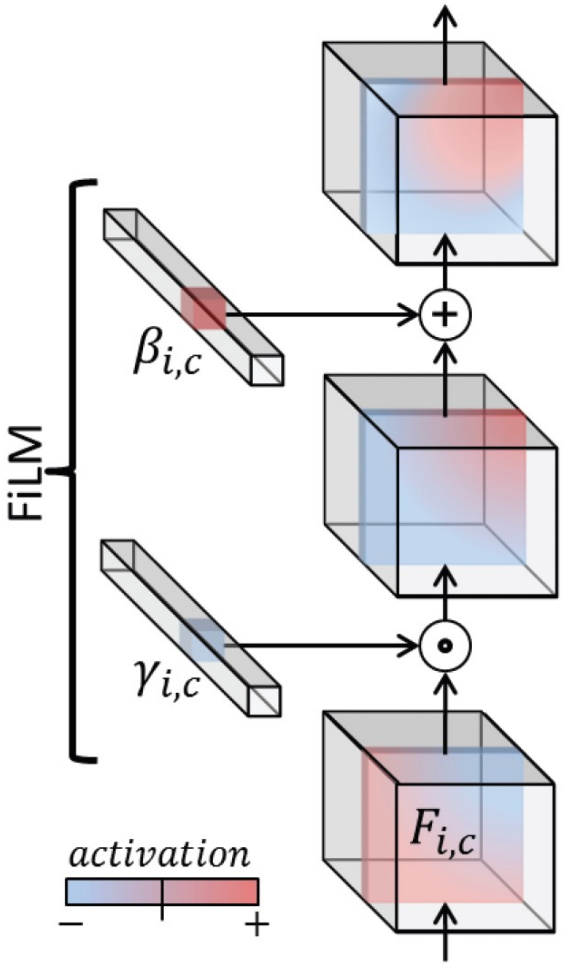
- This model produces a sequence of low-level robot controls in real time from a text instruction and several video frames
- Control tokens include quantized targets for base and gripper position and orientation



Brohan et al. (2022)

RT-1

- Video frames are processed with an EfficientNet to create an 9x9 grid of 512-D tokens per frame
- Feature-wise layer modulation (Perez et al., 2018) is used to condition EfficientNet layers on an embedding of the instructions
  - Instruction embedding mapped to scale and bias terms that modulate each channel
  - This allows the model to learn to emphasize channels that are relevant to the text
  - They start with a pretrained EfficientNet and initialize FiLM with scale 1 and bias 0

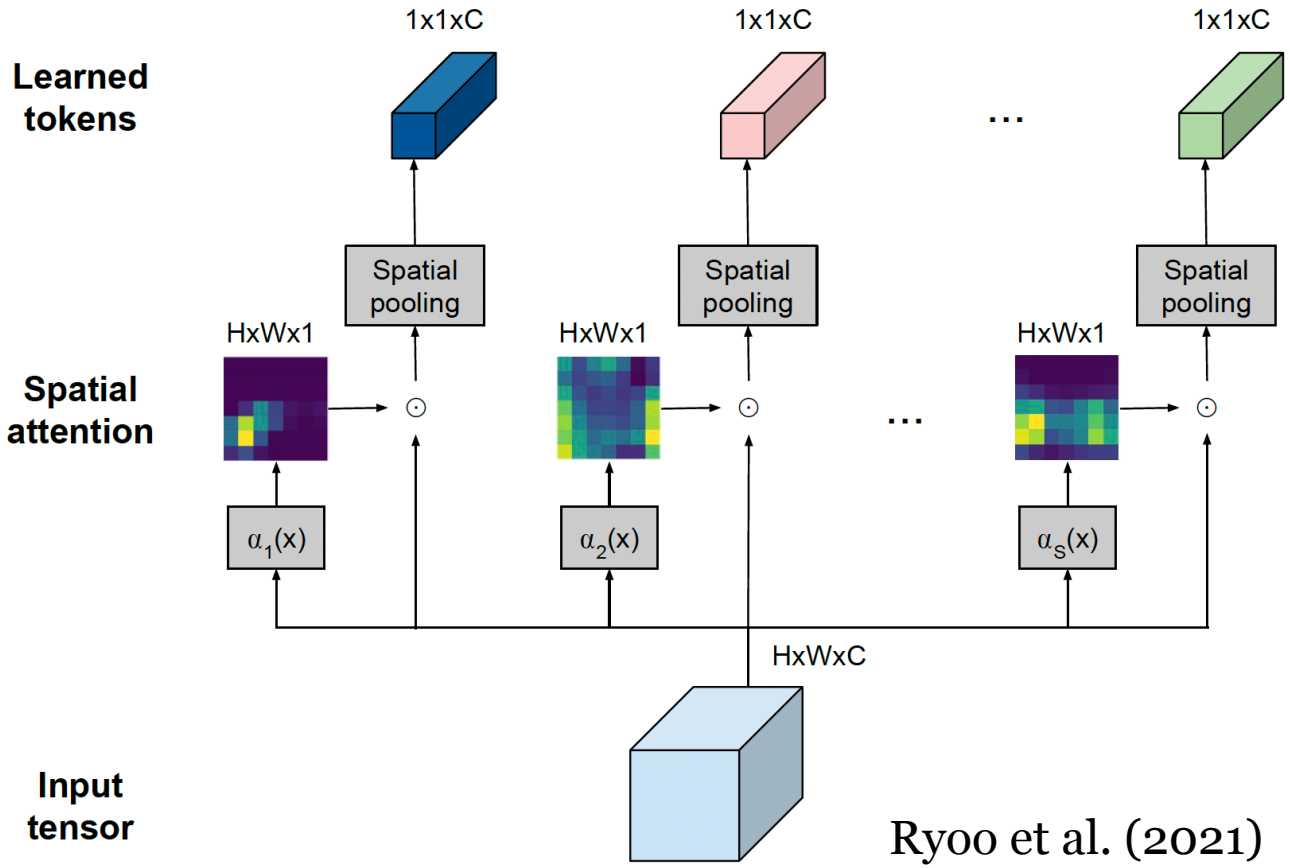


Perez et al. (2018)

UNIVERSITY OF WATERLOO

RT-1

- A TokenLearner (Ryoo et al., 2021) is used to reduce 9x9 tokens per image to 8 tokens per image
  - Maps input to eight learned spatial weighting kernels
  - Tokens are multiplied by these kernels and average-pooled, resulting in new tokens that selected information
- This reduces the sequence length, which is important for video processing



Ryoo et al. (2021)

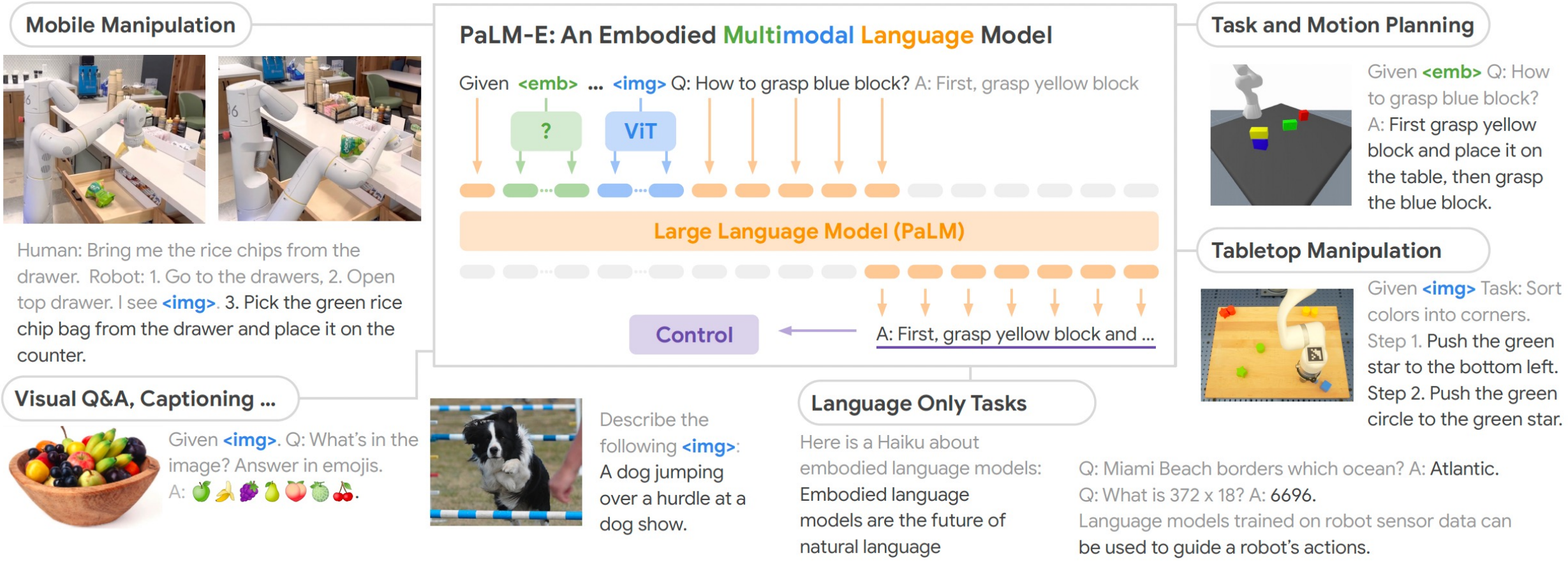
RT-1

- Trained on ~130K action demonstrations which they grouped into “skills” that correspond to different verbs; the model generalized to noun-verb combinations not seen in training

Skill	Count	Description	Example Instruction
Pick `Object`	130	Lift the object off the surface	pick iced tea can
Move `Object` Near `Object`	337	Move the first object near the second	move pepsi can near rxbar blueberry
Place `Object` Upright	8	Place an elongated object upright	place water bottle upright
Knock `Object` Over	8	Knock an elongated object over	knock redbull can over
Open `Drawer`	3	Open any of the cabinet drawers	open the top drawer
Close `Drawer`	3	Close any of the cabinet drawers	close the middle drawer
Place `Object` into `Receptacle`	84	Place an object into a receptacle	place brown chip bag into white bowl
Pick `Object` from `Receptacle` and Place on the Counter	162	Pick an object up from a location and then place it on the counter	pick green jalapeno chip bag from paper bowl and place on counter
Section 6.3 and 6.4 tasks	9	Skills trained for realistic, long instructions	open the large glass jar of pistachios pull napkin out of dispenser grab scooper
Total	744		

Brohan et al. (2022)

PaLM-E (Pathways Language Model Embodied)



Driess et al. (2023)

PaLM-E

- Combines pretrained PaLM LLM, ViT, and environment-specific control modules such as RT-1
  - ViT outputs are mapped to embeddings of the same size as language token embeddings and inserted into the same sequence, e.g., *Human: <instruction> Robot: <step history>. I see <img>.*
- Given a high-level instruction (e.g., "I spilled my drink, can you bring me something to clean it up?"), the language model produces a sequence of actions (in text form) needed to carry it out
  - The actions must be selected from the control module's capabilities
  - The model must use vision to determine which actions are currently possible and to detect failures
- The control module maps action text and video to of low-level commands

PaLM-E

start → goal

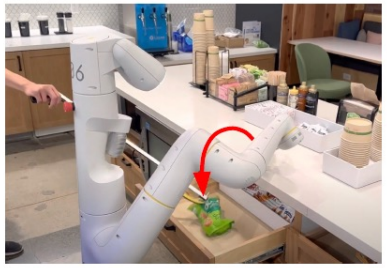
PaLM-E guiding a real robot through a **long horizon** mobile manipulation task
Instruction: *"bring me the rice chips from the drawer"*

Go to the drawers

Open the top drawer

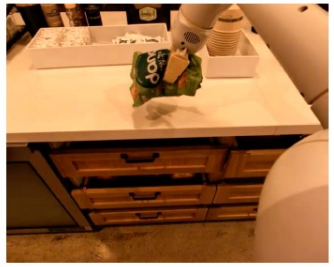
Take the rice chips out of the drawer

Adversarial Disturbance:
human knocks the rice chips back into the drawer



Take the rice chips out of the drawer

Bring it to the user



Put it down

success

Summary

1. Transformers can perform a wide range of vision tasks
2. Transformers can be multimodal

References

- Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., ... & Zitkovich, B. (2022). RT-1: Robotics transformer for real-world control at scale. arXiv preprint arXiv:2212.06817.
- Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020, August). End-to-end object detection with transformers. In *European conference on computer vision* (pp. 213-229). Springer, Cham.
- Cordonnier, J. B., Loukas, A., & Jaggi, M. (2019). On the relationship between self-attention and convolutional layers. *arXiv preprint arXiv:1911.03584*.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*. Fedus, W., Zoph, B., & Shazeer, N. (2021). Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity.
- Driess, D., Xia, F., Sajjadi, M. S., Lynch, C., Chowdhery, A., Ichter, B., ... & Florence, P. (2023). Palm-e: An embodied multimodal language model. arXiv preprint arXiv:2303.03378.
- Guo, Y., Liu, Y., Georgiou, T., & Lew, M. S. (2018). A review of semantic segmentation using deep neural networks. International journal of multimedia information retrieval, 7, 87-93.
- Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., & Shah, M. (2022). Transformers in vision: A survey. *ACM computing surveys (CSUR)*, *54*(10s), 1-41.
- Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 10012-10022).

References

- Perez, E., Strub, F., De Vries, H., Dumoulin, V., & Courville, A. (2018, April). Film: Visual reasoning with a general conditioning layer. In Proceedings of the AAAI conference on artificial intelligence (Vol. 32, No. 1).
- Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... & Sutskever, I. (2021, July). Learning transferable visual models from natural language supervision. In International conference on machine learning (pp. 8748-8763). PMLR.
- Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., & Chen, M. (2022). Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*.
- Reed, S., Zolna, K., Parisotto, E., Colmenarejo, S. G., Novikov, A., Barth-Maron, G., ... & de Freitas, N. (2022). A generalist agent. *arXiv preprint arXiv:2205.06175*.
- Ryoo, M., Piergiovanni, A. J., Arnab, A., Dehghani, M., & Angelova, A. (2021). Tokenlearner: Adaptive space-time tokenization for videos. Advances in Neural Information Processing Systems, 34, 12786-12797.
- Srinivas, A., Lin, T. Y., Parmar, N., Shlens, J., Abbeel, P., & Vaswani, A. (2021). Bottleneck transformers for visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 16519-16529).
- Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., & Zhang, L. (2021). Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 22-31).