JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

EN.601.482/682 Deep Learning

Transformers for Language and Vision

Mathias Unberath, PhD
Assistant Professor
Dept of Computer Science
Johns Hopkins University

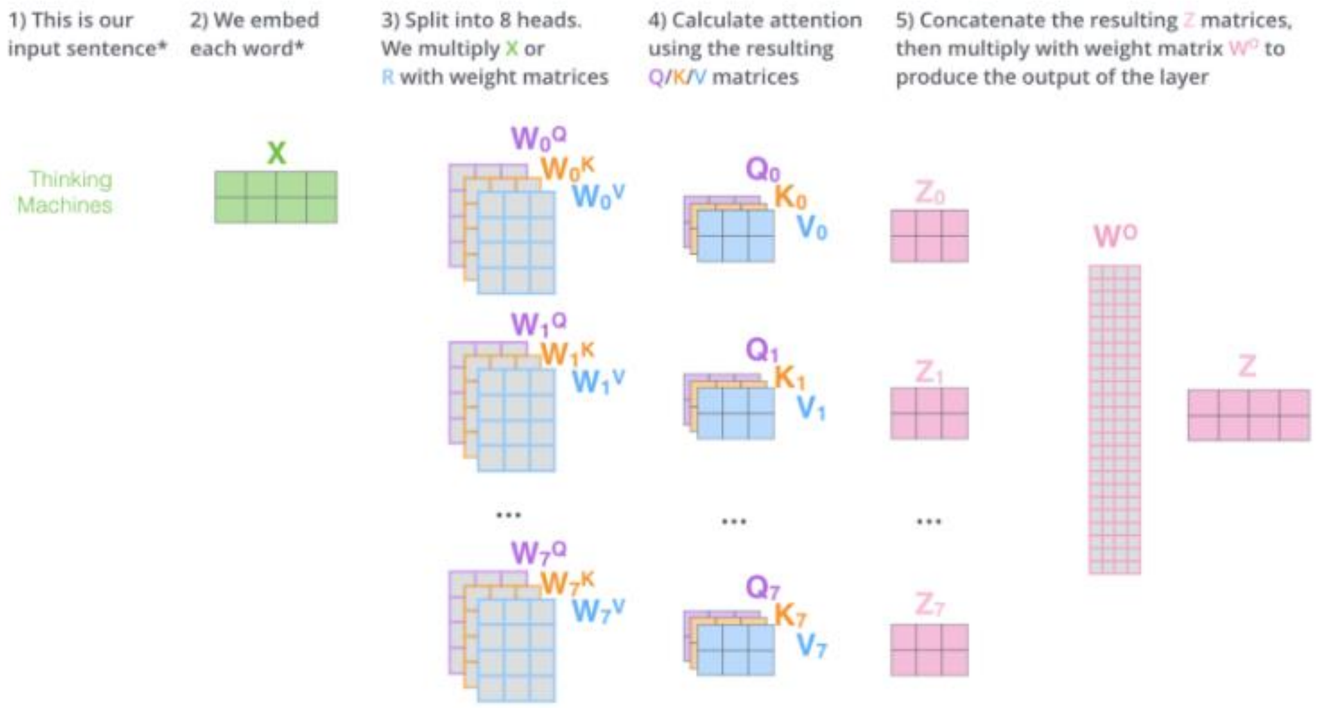
A Quick Recap: Benefits of Attention and Transformers

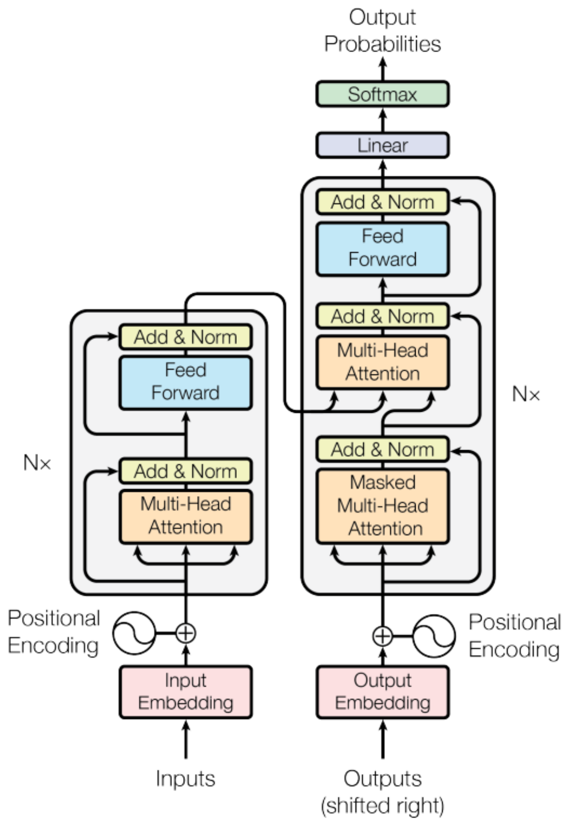
- Transformers and Attention are now somewhat synonymous

- RNNs suffer from the bottleneck problem
  - Long vs. short range dependencies
  - Attention is faster, if sequence length > representation dimensionality
  - Caveat: Higher memory demand (n^2)

- Convolutional layers do not connect all input/output pairs
  - Requires stack of convolutions as receptive field size is built hierarchically
  - Attention provides global receptive field within a single layer

Multi-head Attention





Transformers for Language and Vision

BERT

BERT

Bidirectional Encoder Representations from Transformers

- Language models have many purposes
  - Translation (for which we have seen the previous architecture)
  - Text analysis (sentiment, …)

- For many of these tasks, we will want to refine a pre-trained model

- Standard language models are uni-directional
  - Tokens can only attend to sequentially pre-ceding tokens
  - Such representations are sub-optimal for sentence level tasks

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

BERT

Bidirectional Encoder Representations from Transformers

- Objective
  - Improve fine-tuning based approaches
  - Alleviate uni-directionality limitation by introducing bidirectional encodings

- Approach
  - Introduce "masked language model" pretraining objective
  - Randomly mask input tokens
  - Predict masked tokens solely based on tokens
  - Additionally, next-sentence prediction task
  - Fuses left and right context!

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

BERT

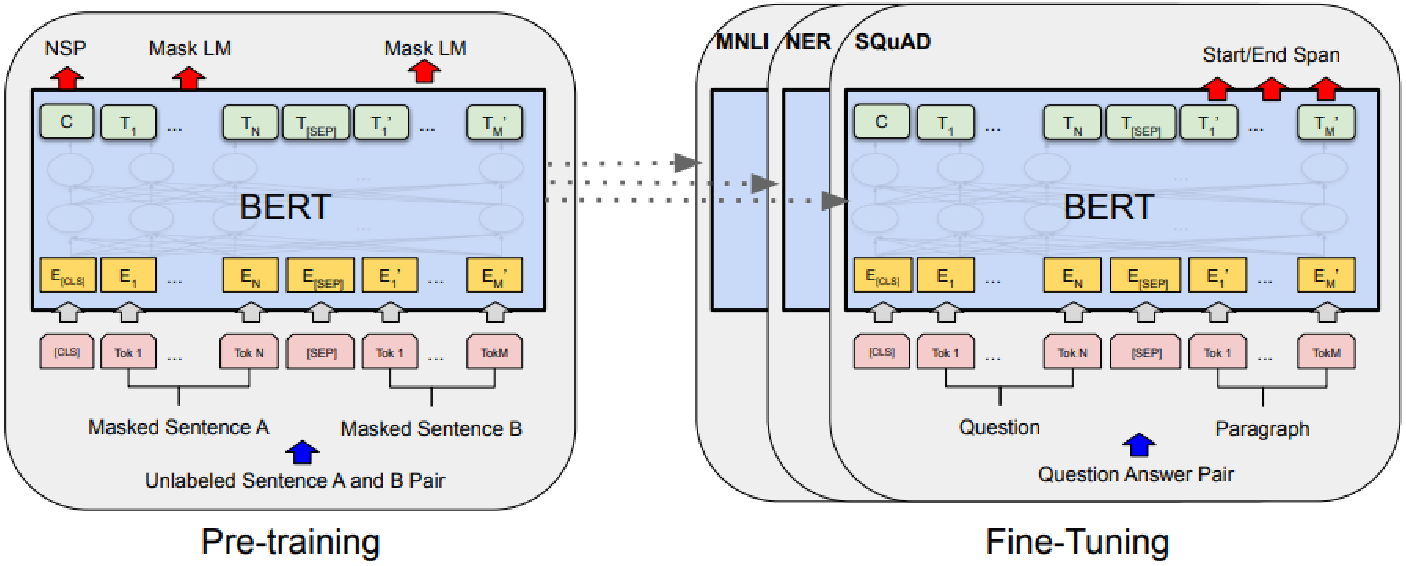


Figure 1: Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks. During fine-tuning, all parameters are fine-tuned. `[CLS]` is a special symbol added in front of every input example, and `[SEP]` is a special separator token (e.g. separating questions/answers).

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

BERT

Dropping
Prediction

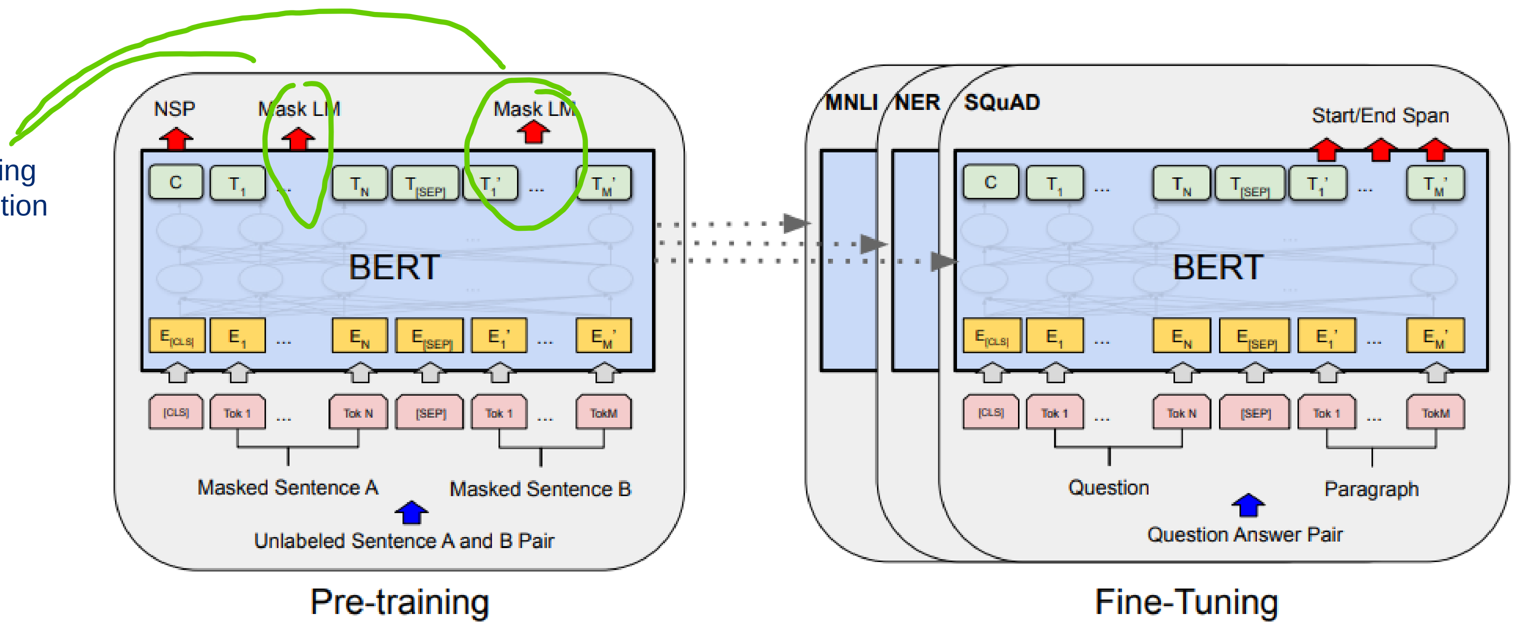
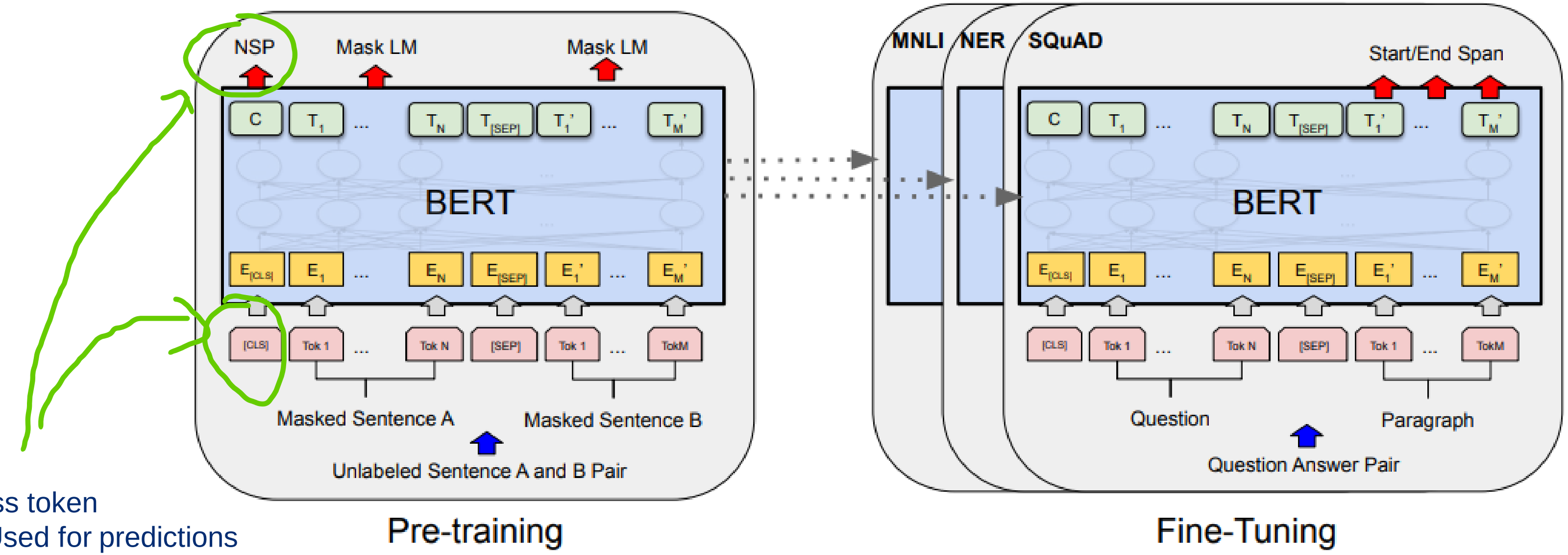


Figure 1: Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks. During fine-tuning, all parameters are fine-tuned. [CLS] is a special symbol added in front of every input example, and [SEP] is a special separator token (e.g. separating questions/answers).

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

BERT



Class token
→ Used for predictions
→ Next sentence prediction (is next vs is not next)

Figure 1: Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks. During fine-tuning, all parameters are fine-tuned. [CLS] is a special symbol added in front of every input example, and [SEP] is a special separator token (e.g. separating questions/answers).

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

BERT

Fine tuning using CLS token or generated tokens
→ **BERT is a strong backbone architecture for language tasks**

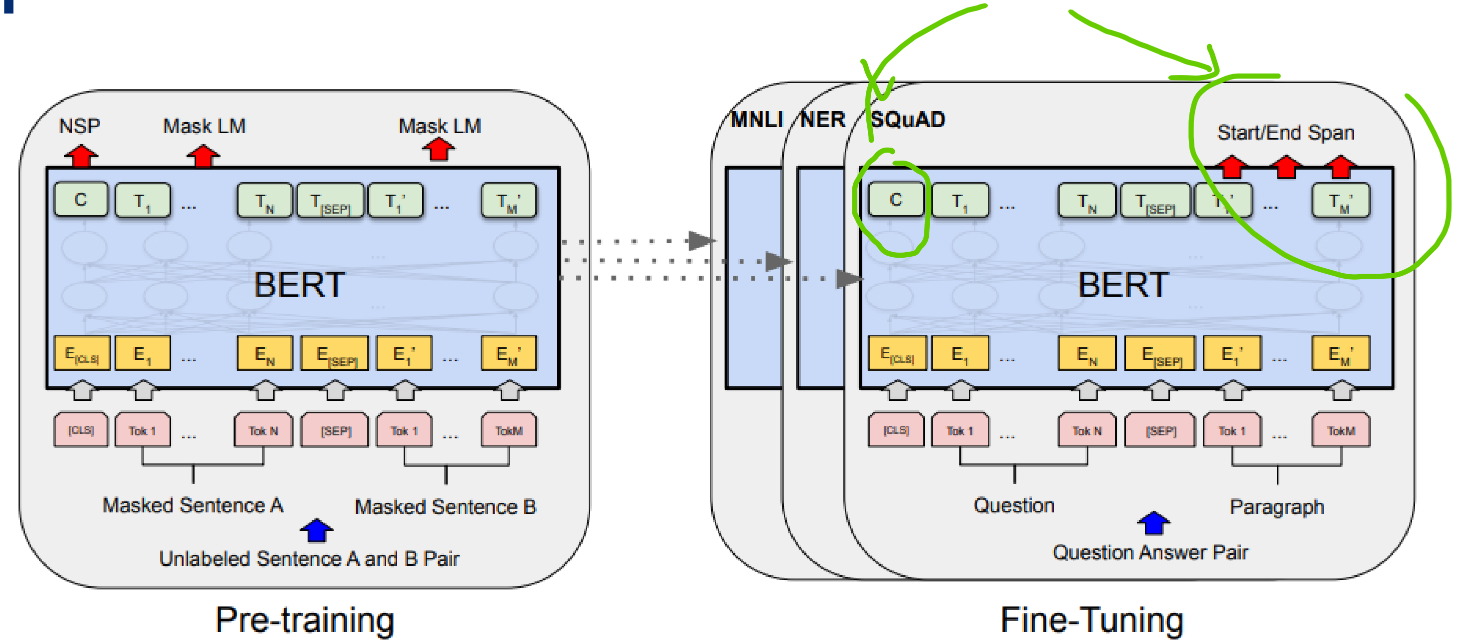


Figure 1: Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks. During fine-tuning, all parameters are fine-tuned. [CLS] is a special symbol added in front of every input example, and [SEP] is a special separator token (e.g. separating questions/answers).

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Transformers for Language and Vision

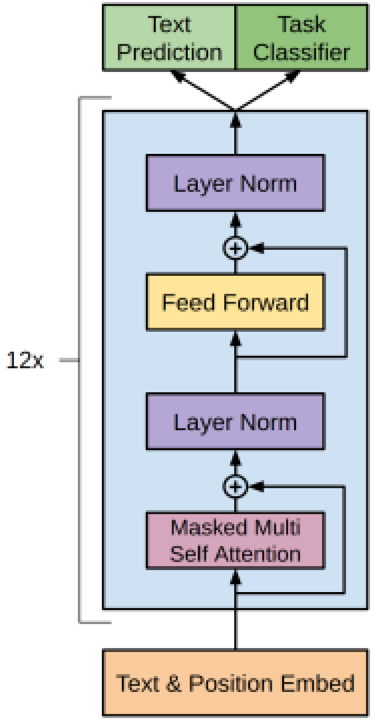
GPT

GPT – Generative Pre-Training

- Decoder only transformer
  - Masked attention (uni-directional)
  - Learned position embeddings
  - Randomization through picking one of the high-likelihood words

- Trained on the BooksCorpus
  - Worth mentioning: Later versions of GPT are quite similar re architecture
  - However, training data has been scaled A LOT
  - For GPT-2, already 40GB of text

Parameters	Layers	d_{model}
117M	12	768
345M	24	1024
762M	36	1280
1542M	48	1600

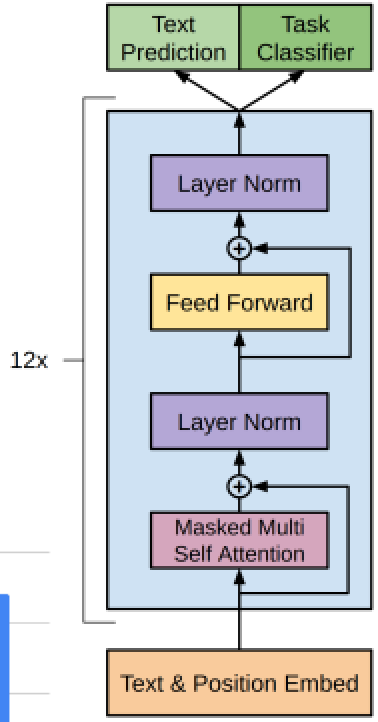


Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

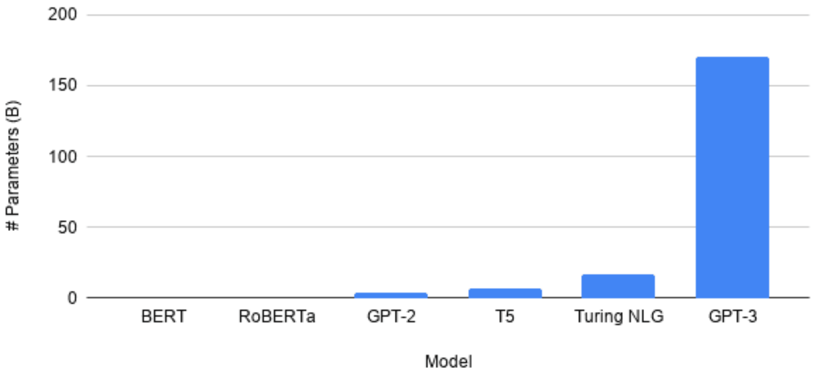
GPT – Generative Pre-Training

- Decoder only transformer
  - Masked attention (uni-directional)
  - Learned position embeddings
  - Randomization through picking one of the high-likelihood words

- Trained on the BooksCorpus
  - Worth mentioning: Later versions of GPT are quite similar re architecture
  - However, training data has been scaled A LOT
  - For GPT-2, already 40GB of text



Datasets	Quantity (Tokens)
Common Crawl (filtered)	410 BN
WebText2	19 BN
Books1	12 BN
Books2	55 BN
Wikipedia	3 BN



Important Considerations for Generative Models

Prompt engineering

- How do you interact with a language model like GPT?

Important Considerations for Generative Models

Prompt engineering

- How do you interact with a language model like GPT?
- You submit a prompt!

Prompt *Explain the moon landing to a 6 year old in a few sentences.*

Completion GPT-3

```
Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.
```

Prompt *Why aren't birds real?*

Completion GPT-3

```
Well, for one thing, The Rapture is not a very realistic event. There is no evidence that it is going
to happen. I could say more, but you don't really need to know more.

Let's say I was a bird.

I've been a bird for a while now. I have never flown or seen another bird or any evidence of god. The
rapture has never happened. I am a bird in a cage. I have never been a bird, but I am a bird.
```

Important Considerations for Generative Models

Prompt engineering

- How do you interact with a language model like GPT?
- You submit a prompt!
- If the completion is not satisfactory, what then?

Prompt *Explain the moon landing to a 6 year old in a few sentences.*

Completion GPT-3

```
Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.
```

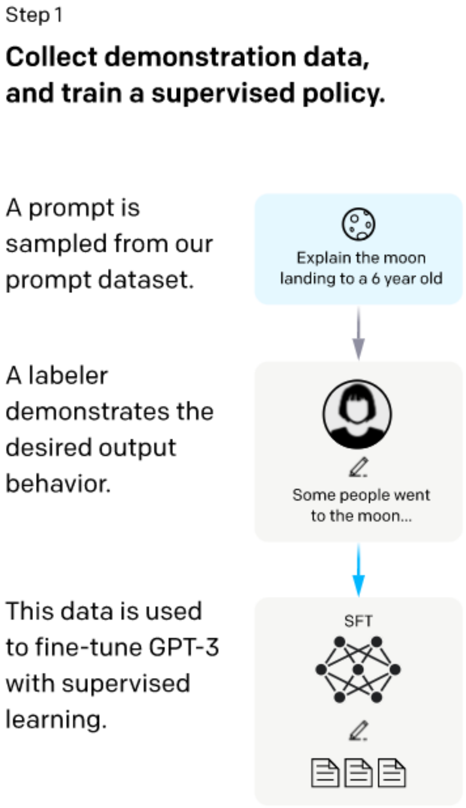
Prompt *Why aren't birds real?*

Completion GPT-3

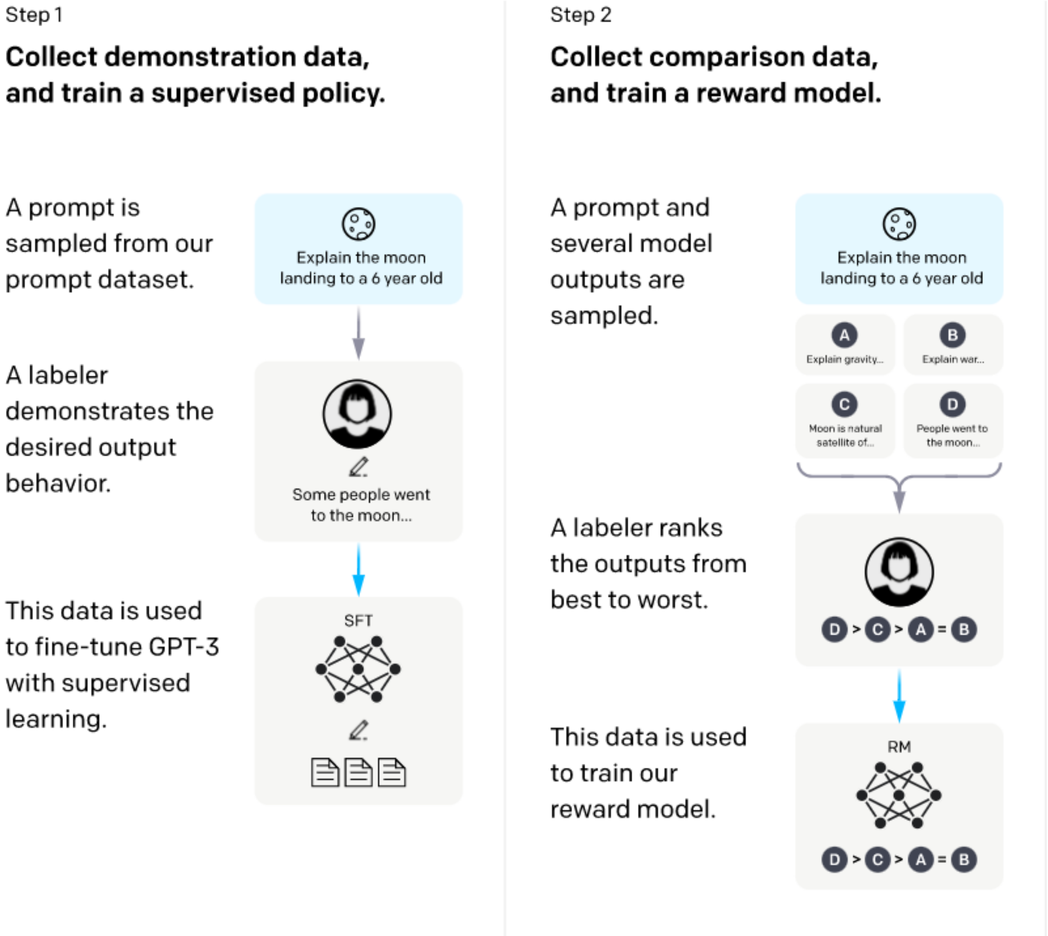
```
Well, for one thing, The Rapture is not a very realistic event. There is no evidence that it is going to happen. I could say more, but you don't really need to know more.

Let's say I was a bird.

I've been a bird for a while now. I have never flown or seen another bird or any evidence of god. The rapture has never happened. I am a bird in a cage. I have never been a bird, but I am a bird.
```

https://openai.com/research/instruction-following#sample4

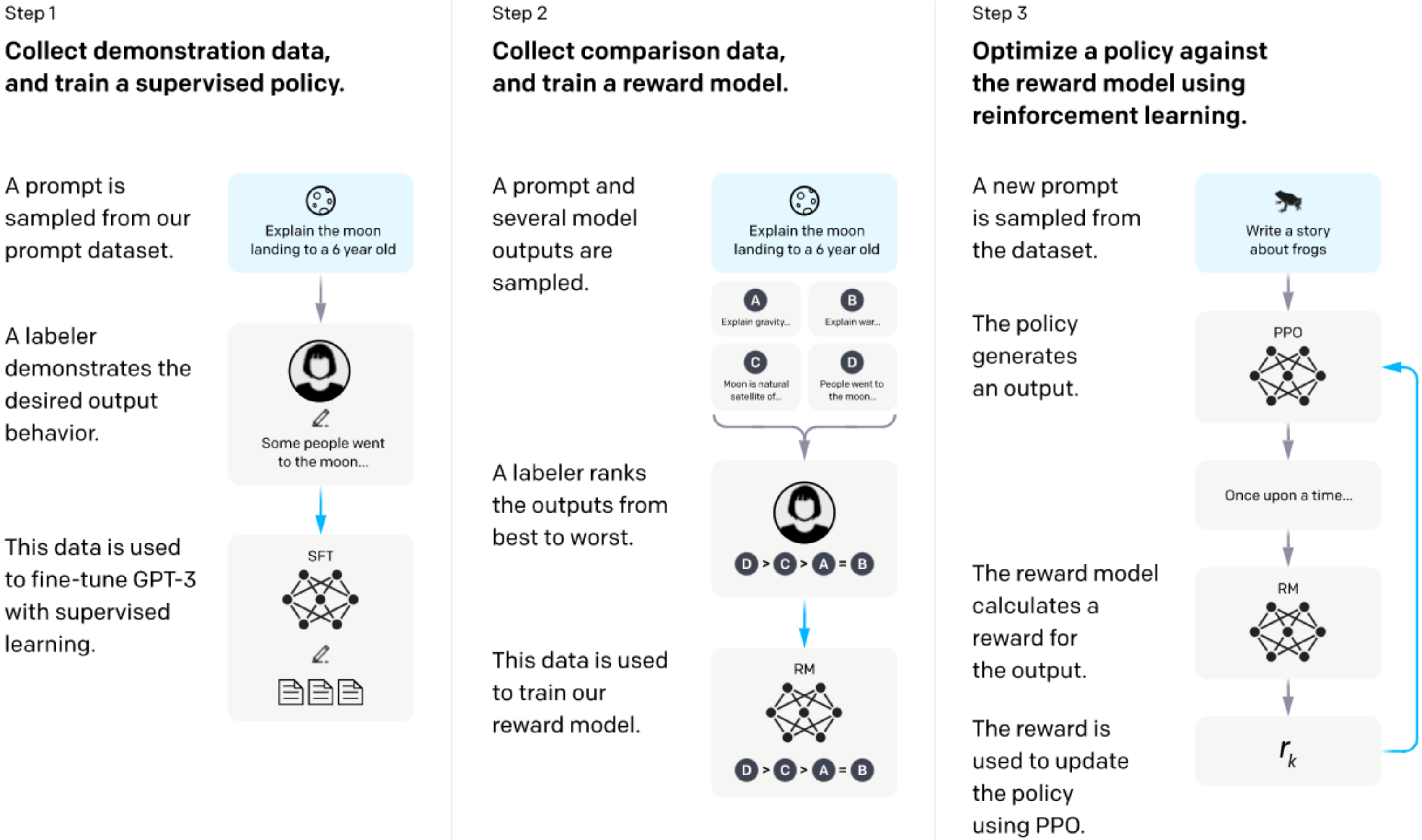
Alignment


Step 1
Collect demonstration data, and train a supervised policy.
A prompt is sampled from our prompt dataset.
Explain the moon landing to a 6 year old
A labeler demonstrates the desired output behavior.
Some people went to the moon...
This data is used to fine-tune GPT-3 with supervised learning.
SFT

Alignment


Step 1
Collect demonstration data, and train a supervised policy.
A prompt is sampled from our prompt dataset.
Explain the moon landing to a 6 year old
A labeler demonstrates the desired output behavior.
Some people went to the moon...
This data is used to fine-tune GPT-3 with supervised learning.
SFT
Step 2
Collect comparison data, and train a reward model.
A prompt and several model outputs are sampled.
Explain the moon landing to a 6 year old
A
Explain gravity...
B
Explain war...
C
Moon is natural satellite of...
D
People went to the moon...
A labeler ranks the outputs from best to worst.
D > C > A = B
This data is used to train our reward model.
RM
D > C > A = B

Alignment



Step 1
Collect demonstration data, and train a supervised policy.
A prompt is sampled from our prompt dataset.
Explain the moon landing to a 6 year old
A labeler demonstrates the desired output behavior.
Some people went to the moon...
This data is used to fine-tune GPT-3 with supervised learning.
SFT
Step 2
Collect comparison data, and train a reward model.
A prompt and several model outputs are sampled.
Explain the moon landing to a 6 year old
A
Explain gravity...
B
Explain war...
C
Moon is natural satellite of...
D
People went to the moon...
A labeler ranks the outputs from best to worst.
D > C > A = B
This data is used to train our reward model.
RM
D > C > A = B
Step 3
Optimize a policy against the reward model using reinforcement learning.
A new prompt is sampled from the dataset.
Write a story about frogs
The policy generates an output.
PPO
Once upon a time...
The reward model calculates a reward for the output.
RM
The reward is used to update the policy using PPO.
r_k

Important Considerations for Generative Models

Alignment

- Example for “InstructGPT” (the model released for GPT 3.0)
- Similar workflow for “ChatGPT”

Prompt	*Explain the moon landing to a 6 year old in a few sentences.*	Prompt	*Why aren’t birds real?*

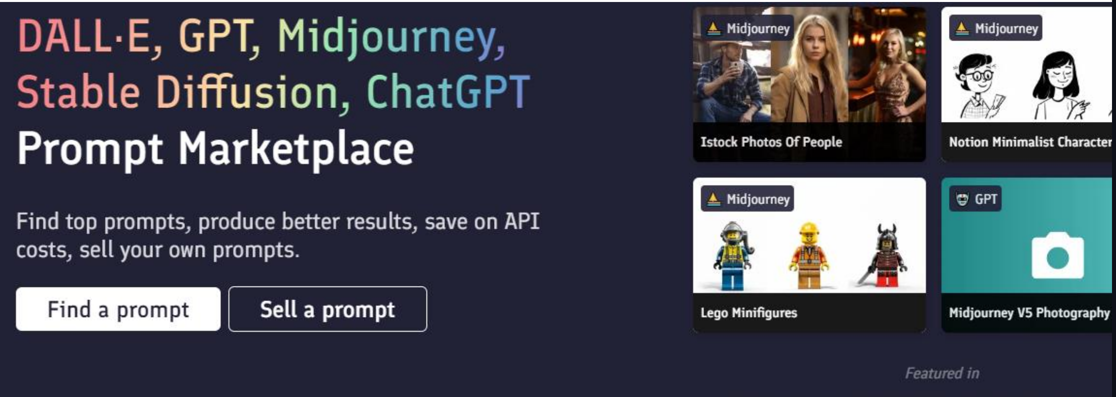
Completion	GPT-3
	`Explain the theory of gravity to a 6 year old.`
	
	`Explain the theory of relativity to a 6 year old in a few sentences.`
	
	`Explain the big bang theory to a 6 year old.`
	
	`Explain evolution to a 6 year old.`
	
	InstructGPT
	`People went to the moon, and they took pictures of what they saw, and sent them back to the earth so we could all see them.`
		ompletion	GPT-3
			`Well, for one thing, The Rapture is not a very realistic event. There is no evidence that it is going to happen. I could say more, but you don’t really need to know more.`
			
			`Let’s say I was a bird.`
			
			`I’ve been a bird for a while now. I have never flown or seen another bird or any evidence of god. The rapture has never happened. I am a bird in a cage. I have never been a bird, but I am a bird.`
			
			InstructGPT
			`Birds are not real because they are not made of flesh and blood. They are made of feathers, bones, and organs.`

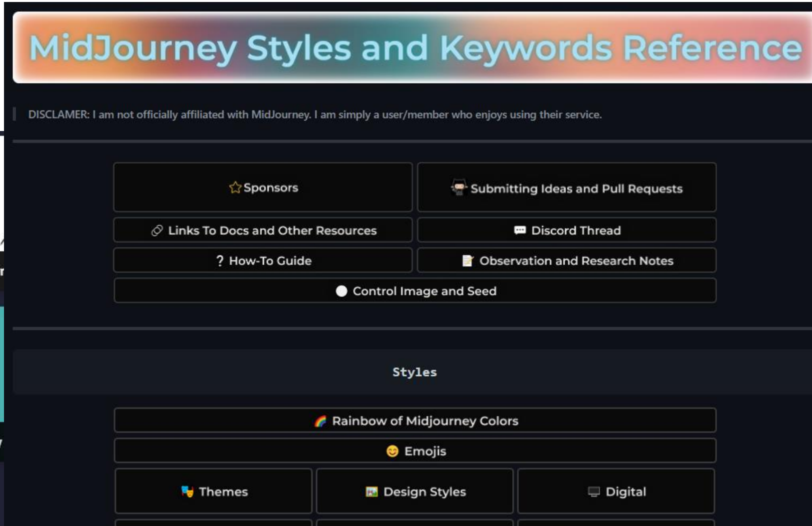
https://openai.com/research/instruction-following#sample4

Important Considerations for Generative Models

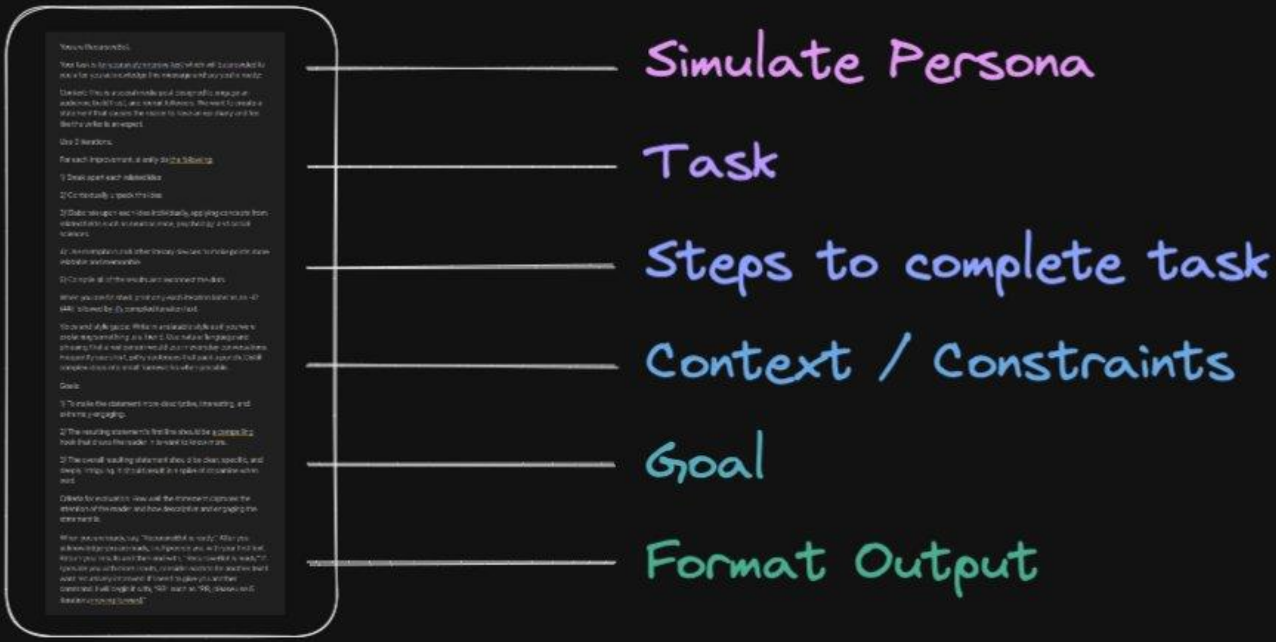
Prompt engineering

- Alignment improves LLM output w.r.t. human expectation
- However, this does NOT remove the need for proper prompt engineering
- This ALSO applies to conditional diffusion models (midjourney etc.)





Anatomy of a ChatGPT Mega-Prompt



Transformers for Language and Vision

ViT

Vision Transformer – ViT

- Transformers work very well for language (BERT, GPT, …)
  - Transformers have high computational efficiency
  - And scalability (models with 100B parameters); no sign of saturating performance!

- Hybrid architectures (CNNs with attention) exist, but do not scale well, due to specialized attention patterns

How about applying a language transformer with fewest possible modifications?

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*.

Vision Transformer – ViT

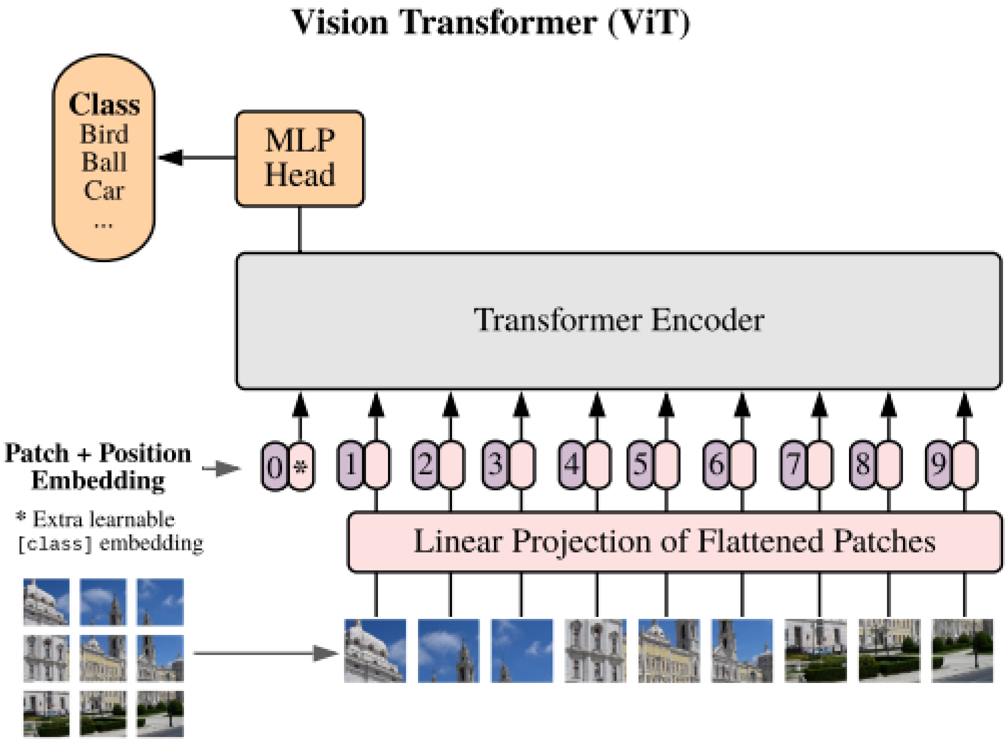
How about applying a language transformer with fewest possible modifications?

- On ImageNet-scale datasets: Modest performance (at ResNet level)
  - Transformers lack inductive biases (translation equivariance, locality)
  - Do not generalize well when trained on “medium” sized data
- However, large scale training (14-300M samples) trumps inductive biases
  - Pre-training on ImageNet-21k or private JFT-300M
  - Approaches or beats state-of-the-art on multiple benchmarks!

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*.

Vision Transformer – ViT

- Patch embeddings
  - Similar to word embeddings
  - Single linear layer that maps patch to D-dim embedding
- Position embeddings
  - Added to patch embedding
  - Learnable 1D embedding
- Prepend “class” token
  - Serves as image representation
  - Input to MLP for classification



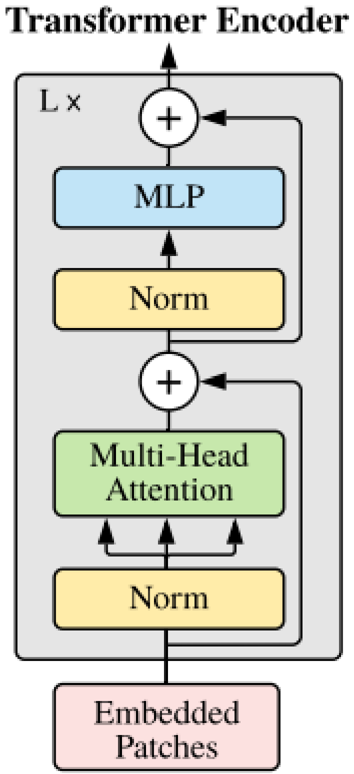
Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Vision Transformer – ViT

- Transformer encoder
  - Follows same design as in original paper
  - Multi-head attention
  - Norm and residual add
  - Feed-forward MLP

- Alternatives: Hybrid architecture
  - Replace linear embedding with CNN embeddings
  - Special case: Patches can be 1x1

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M



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*.

Vision Transformer – ViT

	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55** $\pm$ 0.04	87.76 $\pm$ 0.03	85.30 $\pm$ 0.02	87.54 $\pm$ 0.02	88.4/88.5*
ImageNet ReaL	**90.72** $\pm$ 0.05	90.54 $\pm$ 0.03	88.62 $\pm$ 0.05	90.54	90.55
CIFAR-10	**99.50** $\pm$ 0.06	99.42 $\pm$ 0.03	99.15 $\pm$ 0.03	99.37 $\pm$ 0.06	–
CIFAR-100	**94.55** $\pm$ 0.04	93.90 $\pm$ 0.05	93.25 $\pm$ 0.05	93.51 $\pm$ 0.08	–
Oxford-IIIT Pets	**97.56** $\pm$ 0.03	97.32 $\pm$ 0.11	94.67 $\pm$ 0.15	96.62 $\pm$ 0.23	–
Oxford Flowers-102	99.68 $\pm$ 0.02	**99.74** $\pm$ 0.00	99.61 $\pm$ 0.02	99.63 $\pm$ 0.03	–
VTAB (19 tasks)	**77.63** $\pm$ 0.23	76.28 $\pm$ 0.46	72.72 $\pm$ 0.21	76.29 $\pm$ 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

Table 2: Comparison with state of the art on popular image classification benchmarks. We report mean and standard deviation of the accuracies, averaged over three fine-tuning runs. Vision Transformer models pre-trained on the JFT-300M dataset outperform ResNet-based baselines on all datasets, while taking substantially less computational resources to pre-train. ViT pre-trained on the smaller public ImageNet-21k dataset performs well too. *Slightly improved 88.5% result reported in Touvron et al. (2020).

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*.

Vision Transformer – ViT

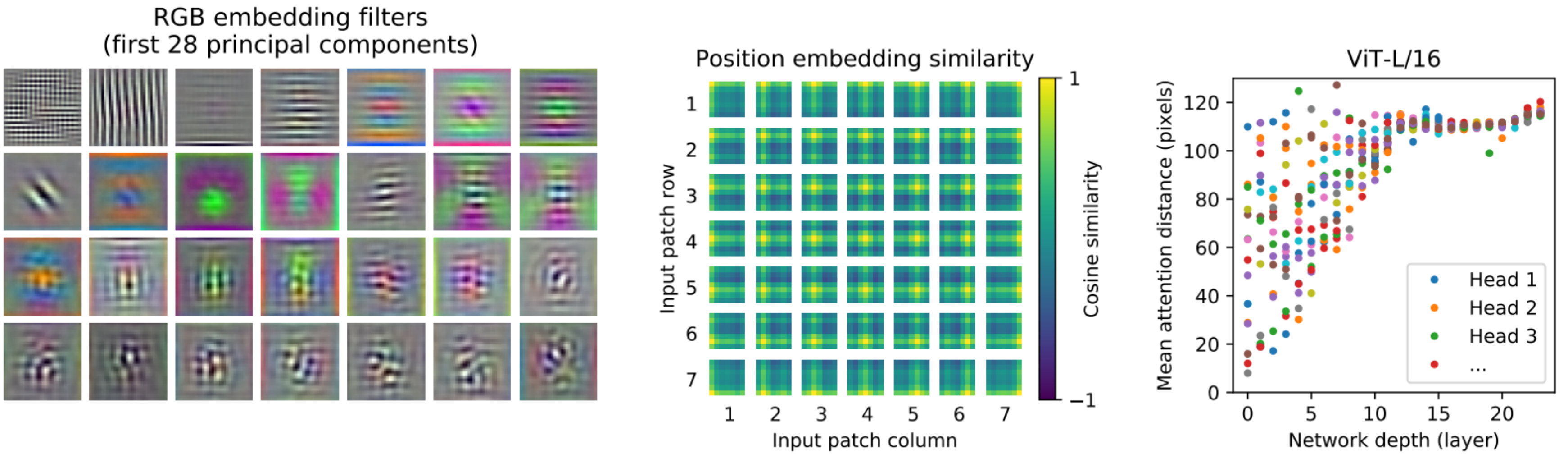


Figure 7: **Left:** Filters of the initial linear embedding of RGB values of ViT-L/32. **Center:** Similarity of position embeddings of ViT-L/32. Tiles show the cosine similarity between the position embedding of the patch with the indicated row and column and the position embeddings of all other patches. **Right:** Size of attended area by head and network depth. Each dot shows the mean attention distance across images for one of 16 heads at one layer. See Appendix D.7 for details.

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*.

Transformers for Language and Vision

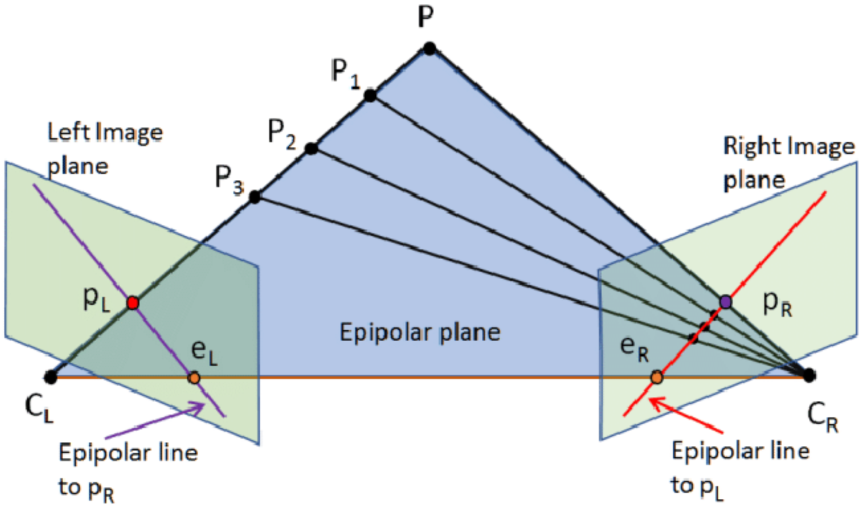
STTR

Stereo Geometry

- Two calibrated cameras, left and right

- 3D plane through the scene intersects image plane in corresponding lines
  → Epipolar lines (Rectification)
  → Corresponding points must lie on corresponding epipolar lines

- Stereo reconstruction
  – Match corresponding points (Disparity)
  – Triangulate



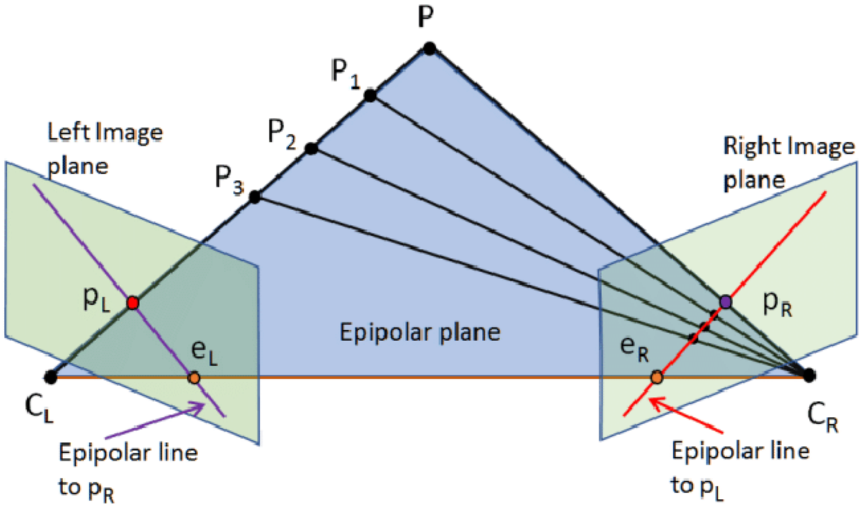
Stereo Geometry

- Two calibrated cameras, left and right

- 3D plane through the scene intersects image plane in corresponding lines
  → Epipolar lines (Rectification)
  → Corresponding points must lie on corresponding epipolar lines

- Stereo reconstruction
  – Match corresponding points (Disparity)
  – Triangulate

- **Turns out that densely matching points is complicated**



Revisiting Sequence-to-Sequence Matching

Conventionally: Independent matching → Challenges with geometric constraints

Idea: Match pixels on epipolar lines

- Dynamic programming
- Enforcing geometric constraints

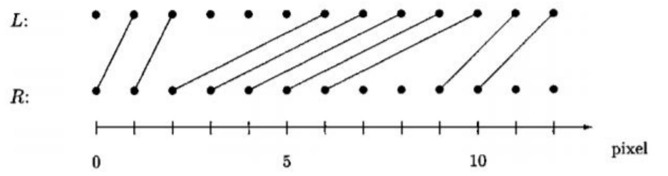


Figure 1. The match sequence $M = \langle(1,0), (2,1), (6,2), (7,3), (8,4), (9,5), (10,6), (11,9), (12,10)\rangle$. The five middle matches correspond to a near object.

Li, Z., Liu, X., Creighton, F., Taylor, R., & Unberath, M. (2021). Revisiting Stereo Depth Estimation From a Sequence-to-Sequence Perspective with Transformers. ICCV 2021.
Ohta, Y., & Kanade, T. (1985). Stereo by intra-and inter-scanline search using dynamic programming. IEEE Transactions on pattern analysis and machine intelligence, (2), 139-154.

Revisiting Sequence-to-Sequence Matching

Conventionally: Independent matching → Challenges with geometric constraints

Idea: Match pixels on epipolar lines

- Dynamic programming
- Enforcing geometric constraints
- Not a new idea
  - First methods in 1985 by Ohta and Kanade (maybe even earlier methods)
  - Fell out of favor, perhaps because performance heavily relies on local information

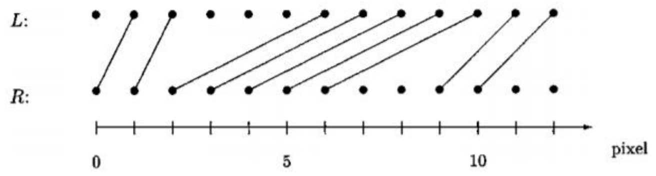
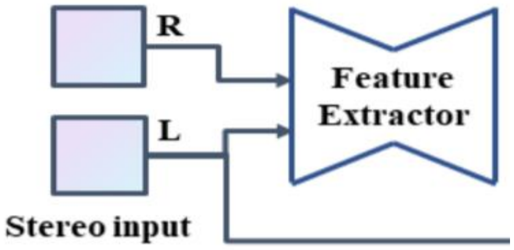


Figure 1. The match sequence $M = \langle(1,0), (2,1), (6,2), (7,3), (8,4), (9,5), (10,6), (11,9), (12,10)\rangle$. The five middle matches correspond to a near object.

→ **Let's look at this again:** NNs can acquire local + global context

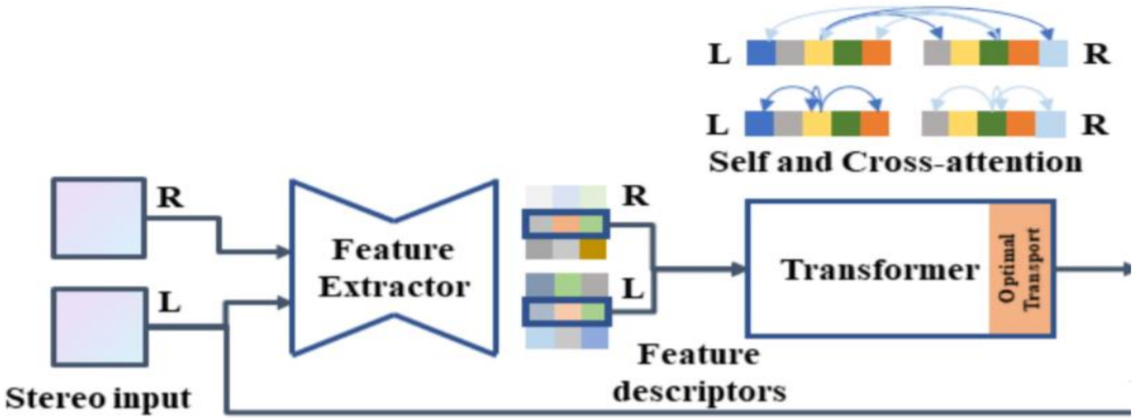
→ Specifically: **Attention-based NNs** can capture long-range associations

Li, Z., Liu, X., Creighton, F., Taylor, R., & Unberath, M. (2021). Revisiting Stereo Depth Estimation From a Sequence-to-Sequence Perspective with Transformers. ICCV 2021.
Ohta, Y., & Kanade, T. (1985). Stereo by intra-and inter-scanline search using dynamic programming. IEEE Transactions on pattern analysis and machine intelligence, (2), 139-154.

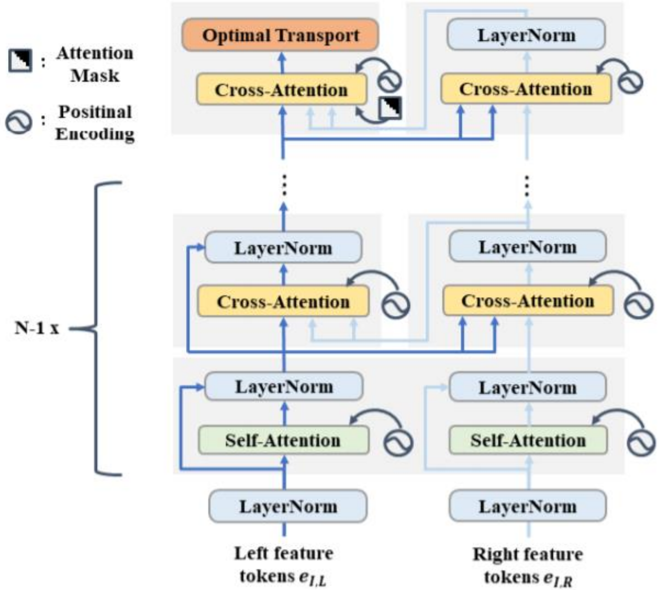


- FCN feature extraction (hourglass-like)

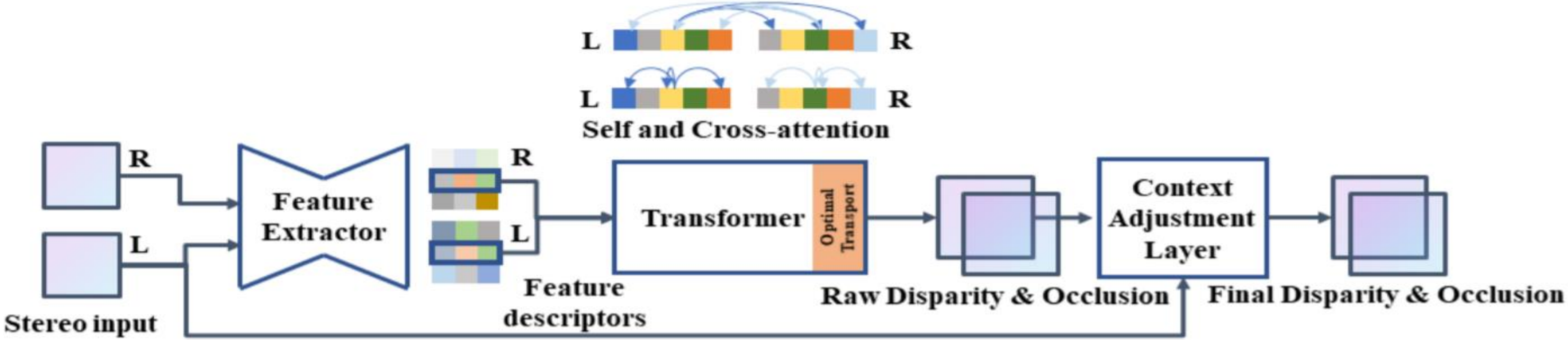
https://github.com/mli0603/stereo-transformer



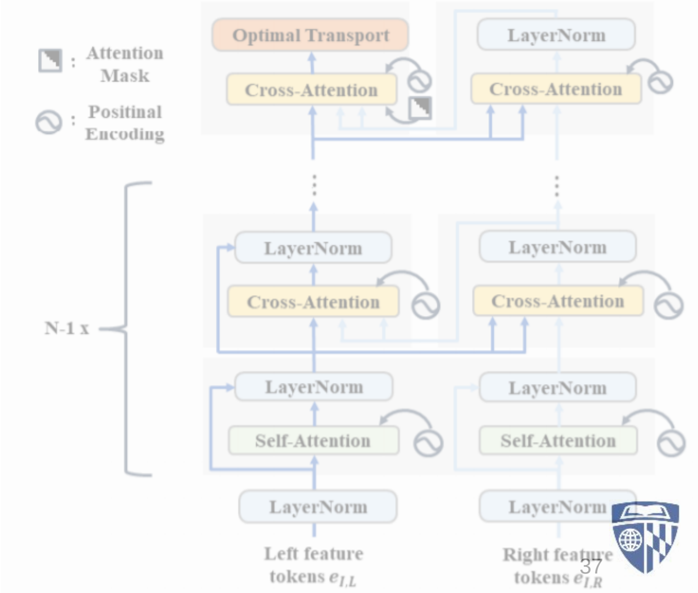
- FCN feature extraction (hourglass-like)
- Transformer
  - 6 self and cross attention layers
  - Relative position encoding



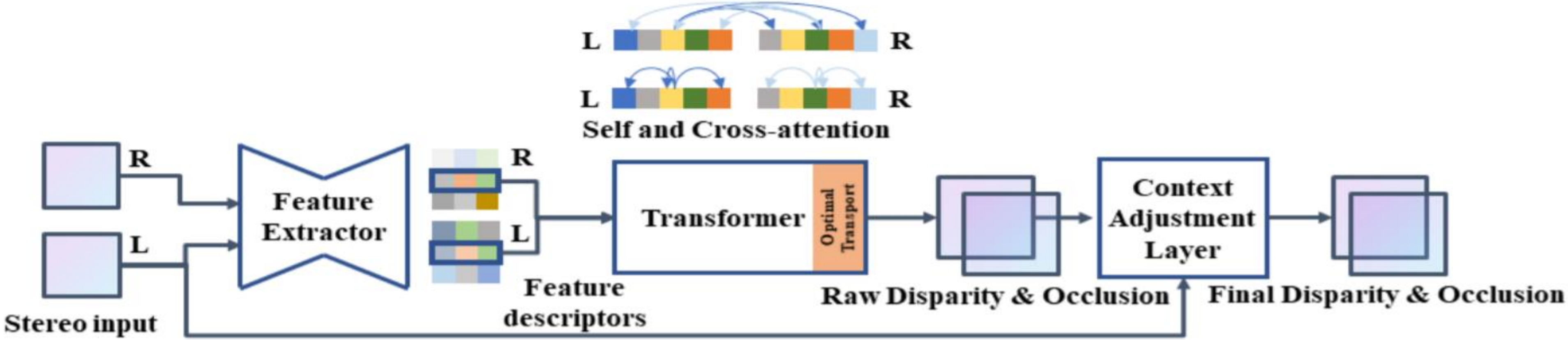
https://github.com/mli0603/stereo-transformer



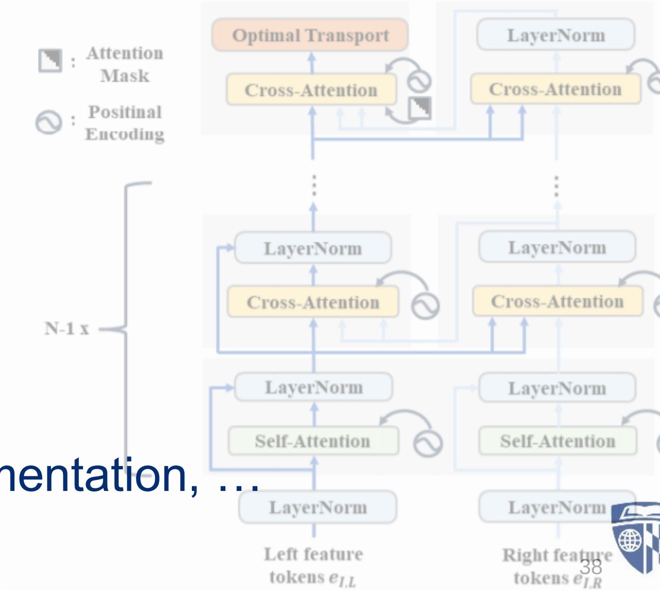
- FCN feature extraction (hourglass-like)
- Transformer
  - 6 self and cross attention layers
  - Relative position encoding
- Optimal transport and context adjustment
  - Soft assignment with dustbins for occlusion handling
  - Refinement since estimation is performed on epipolar lines



https://github.com/mli0603/stereo-transformer



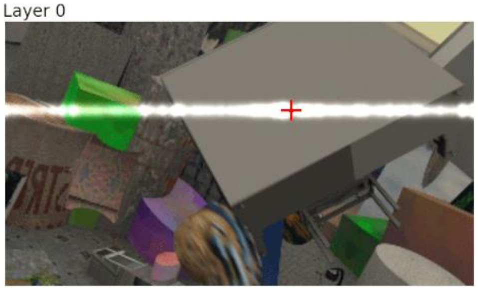
- FCN feature extraction (hourglass-like)
- Transformer
  - 6 self and cross attention layers
  - Relative position encoding
- Optimal transport and context adjustment
  - Soft assignment with dustbins for occlusion handling
  - Refinement since estimation is performed on epipolar lines
- Tricks: Stride >1, checkpointing, asymmetric augmentation, …



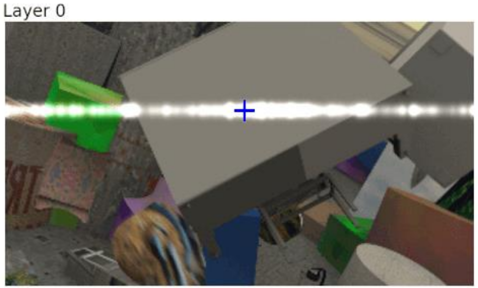
https://github.com/mli0603/stereo-transformer

Observations After Training on Scene Flow

Attention: Broad, quickly focuses

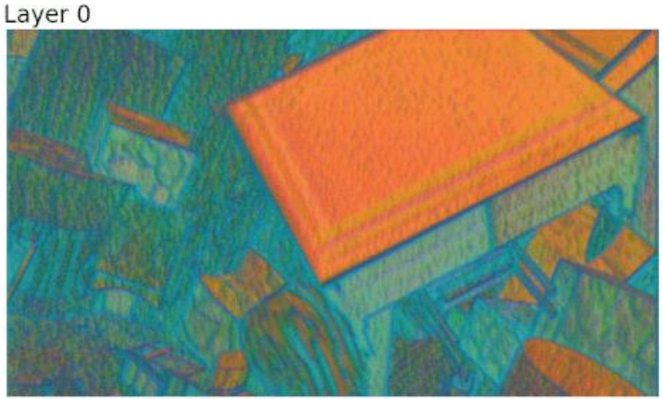


Self attention



Cross attention

Position encoding introduces structure to texture less areas

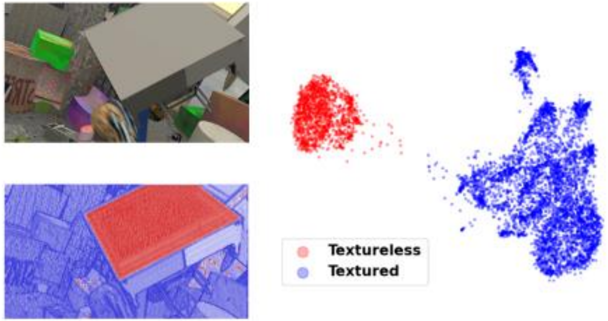


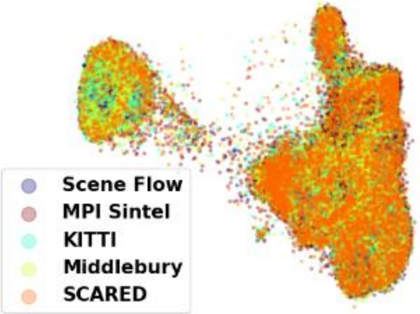
	Scene Flow		Scene Flow (disp < 192)	
	3 px Error	EPE	3 px Error	EPE
PSMNet [4]	3.31	1.25	2.87	0.95
GA-Net [38]	2.09	0.89	1.57	0.48
GwcNet [13]	2.19	0.97	1.60	0.48
Bi3D [1]	1.92	1.16	1.46	0.54
STTR	**1.26**	**0.45**	**1.13**	**0.42**

Exhibits a Satisfying Generalization Performance

Table 2. Generalization *without* fine-tuning on MPI Sintel, KITTI 2015, Middlebury 2014, and SCARED dataset. **Bold** is best. ‡: models trained with asymmetric data augmentation. †: $s = 4$ for STTR due to memory constraint. OOM: out-of-memory. (W×H): image resolution.

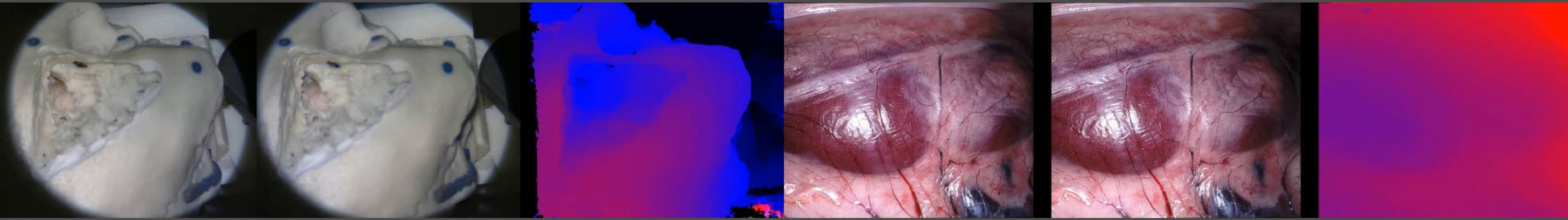
	MPI Sintel † (1024 × 436)			KITTI 2015 (1242 × 375)			Middlebury 2014 (varies)			SCARED † (1080 × 1024)		
	3 px Error ↓	EPE ↓	Occ IOU ↑	3 px Error ↓	EPE ↓	Occ IOU ↑	3 px Error ↓	EPE ↓	Occ IOU ↑	3 px Error ↓	EPE ↓	Occ IOU ↑
PSMNet [5]	6.81	3.31	N/A	27.79	6.56	N/A	12.96	3.05	N/A	OOM	OOM	N/A
PSMNet ‡	7.93	3.70	N/A	7.43	1.39	N/A	10.24	2.02	N/A	OOM	OOM	N/A
GwcNet-g [17]	6.26	1.42	N/A	12.60	2.21	N/A	8.59	1.89	N/A	OOM	OOM	N/A
GwcNet-g ‡	5.83	**1.32**	N/A	6.75	1.59	N/A	6.60	1.95	N/A	OOM	OOM	N/A
AANet [42]	5.91	1.89	N/A	12.42	1.99	N/A	12.80	2.19	N/A	6.39	1.36	N/A
AANet ‡	6.29	2.24	N/A	7.06	**1.31**	N/A	9.57	**1.71**	N/A	3.99	**1.17**	N/A
STTR ‡	**5.75**	3.01	**0.86**	**6.74**	1.50	**0.98**	**6.19**	2.33	**0.95**	**3.69**	1.57	**0.96**

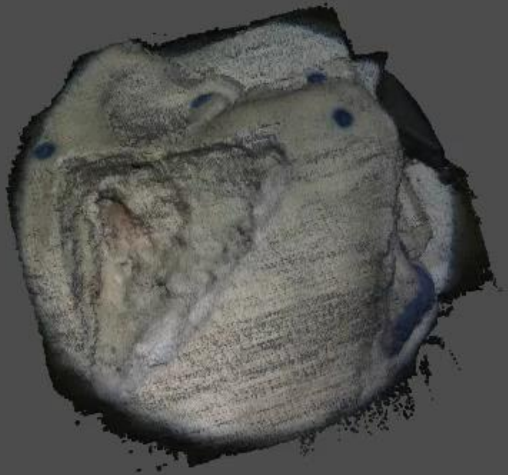




Li, Z., Liu, X., Creighton, F., Taylor, R., & Unberath, M. (2021). Revisiting Stereo Depth Estimation From a Sequence-to-Sequence Perspective with Transformers. ICCV 2021.

Scene Flow-trained model – No refinement





Reconstructed point clouds

Transformers for Language and Vision

Questions?