MSA Scoring & Ranking

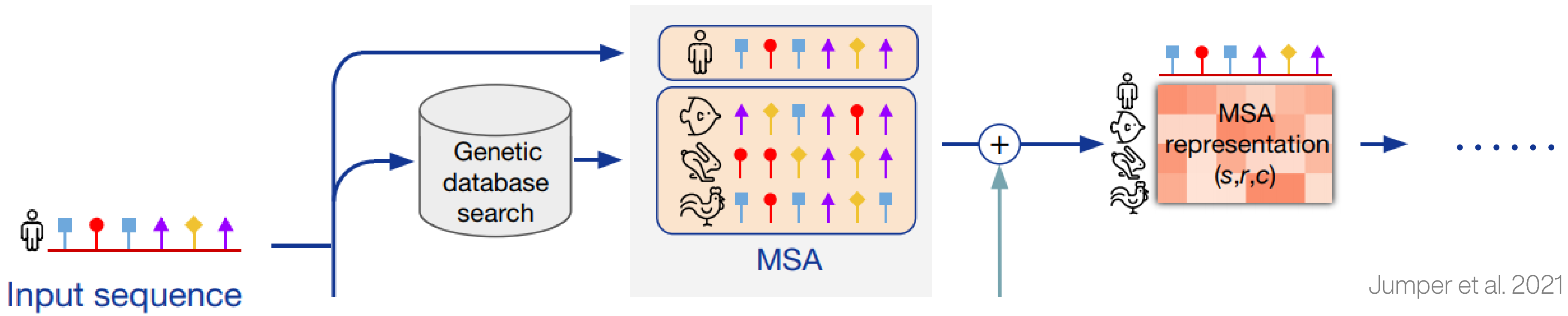
Based on MSA Transformer

Group lianyh, tengyue & yangxch

Background

AlphaFold & MSA Transformer

Background: AlphaFold

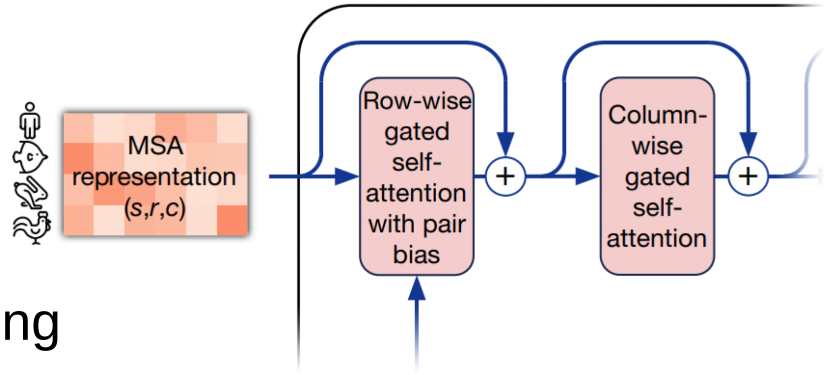


Jumper et al. 2021

- Highly accurate in protein structure prediction
- Requires **Multiple-Sequence Alignment (MSA)** as a key input
- Various MSA acquisition approaches
  - Databases: `BFD, Uniclust30, Uniref90, MGnify, …`
  - Tools: `jackhammer, HHBlits, HHSearch, MMseqs264, …`
- **Motivation**: Given 2 or more MSA inputs, **which MSA input is of higher quality** to help AlphaFold predict highly accurate structure?

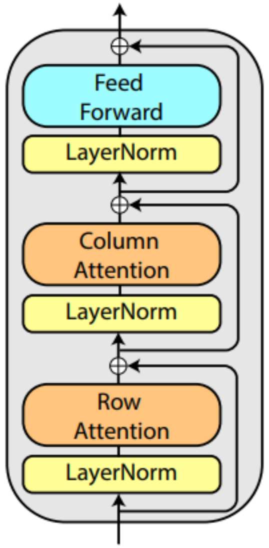
Background: MSA Transformer

- Unsupervised protein language model
- **Masked language modeling objective**
  - Reconstruct masked tokens, similar to BERT pretraining
  - AlphaFold has similar **BERT-like loss** at training
- **Axial self-attention** over rows and columns
  - enables to extract information from dependencies in the input set and generalize patterns across MSAs
  - Also similar to part of AlphaFold **Evoformer** blocks, which exchange information within the MSA to enable direct reasoning about the spatial and evolutionary relationships


MSA representation (s,r,c)
Row-wise gated self-attention with pair bias
Column-wise gated self-attention

AlphaFold Evoformer block, Jumper et al. 2021 ↑

MSA Transformer block, Rao et al. 2021 ↓

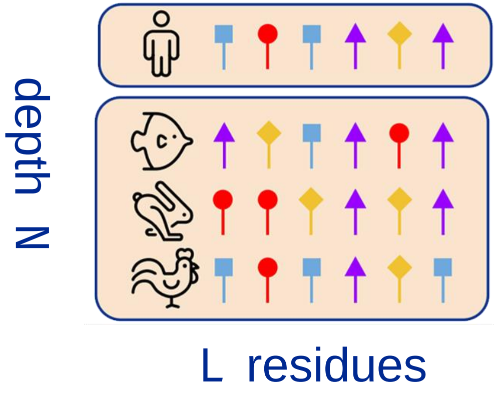


Methodology

Dataset & Networks & Loss Metrics

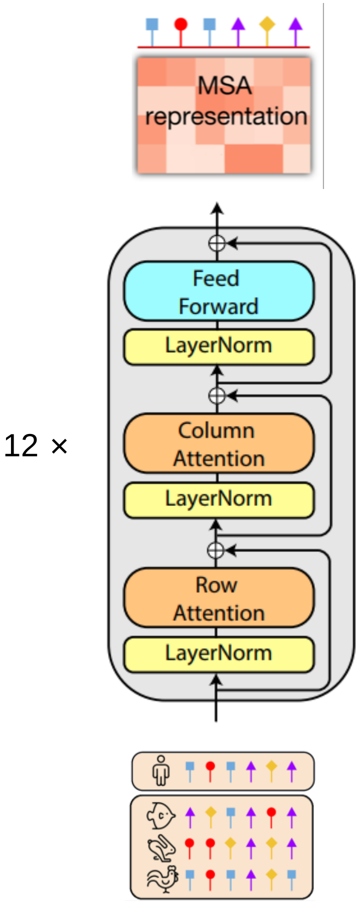
Methodology: Dataset

- 2850 MSAs from CASP14-fm Dataset
  - 2660 train MSAs: 95 query sequences and 28 MSAs for each query
  - 190 test MSAs: 1 pair of MSAs for each query sequence
  - Scores: from AlphaFold output accuracy, scaled to 0~1.0
- MSA subsampling strategy
  - **HH-Filter**: >256 to ≈256 sequences
    - `hhfilter –i input.a3m -o filtered.a3m -diff 256`
  - **Diversity Maximizing**: ≈256 to 256 sequences
    - greedily pick sequence with maximum average hamming distance
  - Result shape: N × L
    - N ≤ 256 and L ≤ 584



Methodology: MSA Embedding

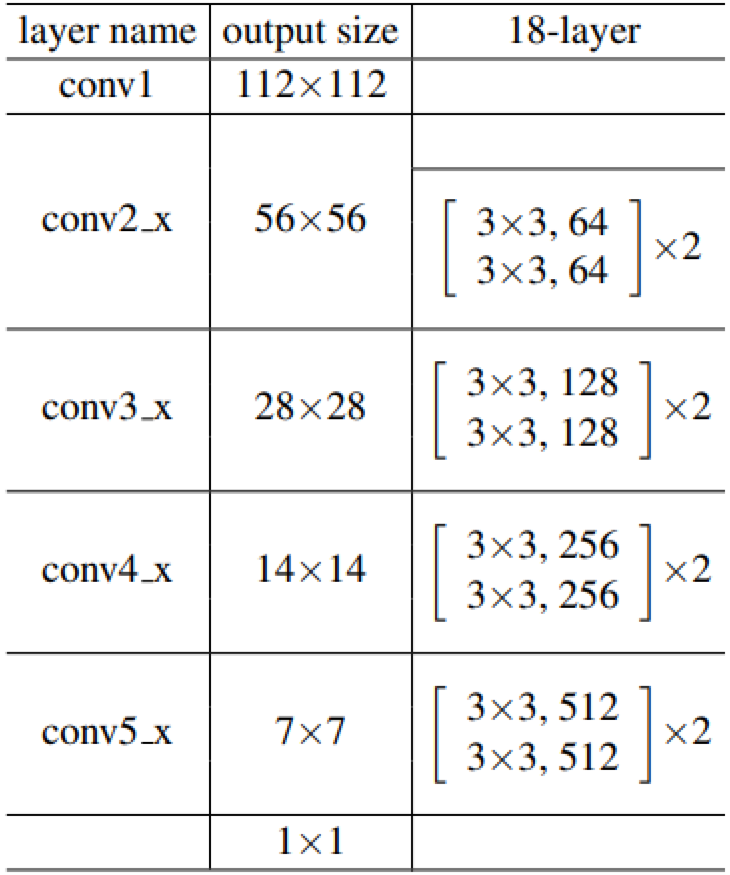
- Use **MSA Transformer** as encoder
  - Import `esm.pretrained.esm_msa1b_t12_100M_UR50S()`
  - Freeze weights of all layers for transfer learning
  - Save extracted embedding of each MSA for efficient training
- Result MSA embedding shape: `N × L × D`
  - MSA Transformer embedding dimension → `D = 768`
  - Add zero paddings → `L = 584`
  - Extract query reference → **`L × D` feature map** each MSA



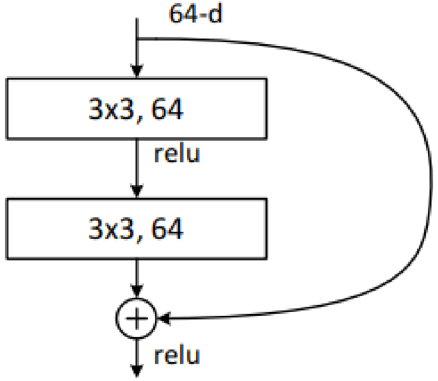
Methodology: Score Regression Network

- Use **Multi-Layer Perceptron (MLP)** for vector regression
  - Averaging pooling/extract <bos> embedding → 768-dim vector
  - Feed into 3 fully-connected layers
    - 768 → 128 → 32 → 1
    - with `leaky_relu` activations and `dropout` layers (except output layer)
  - Output one score and compute `MSELoss`
- Use **Convolutional Neural Network (CNN)** for feature map scoring
  - Each MSA is represented by a `1 × L × D` feature map (like an image)
  - We have investigated 2 different models:
    - **LeNet5**: Interleaving 2 convolution layers with `BatchNorm2d` and max-pooling layers
    - **ResNet18**: 1 convolution layer + 4 residue blocks + 1 average pooling + 1 FC layer

Methodology: Score Regression Network

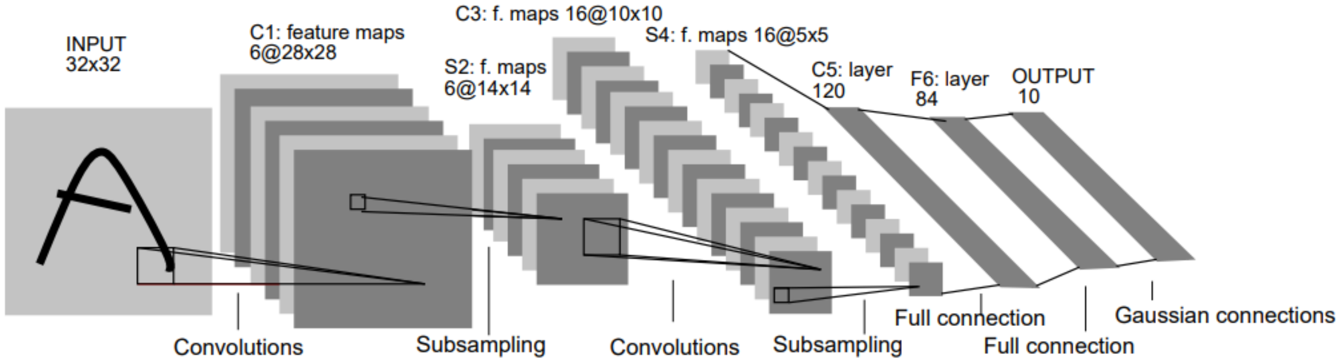


layer name	output size	18-layer
conv1	112×112	
conv2_x	56×56	
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$
	1×1	



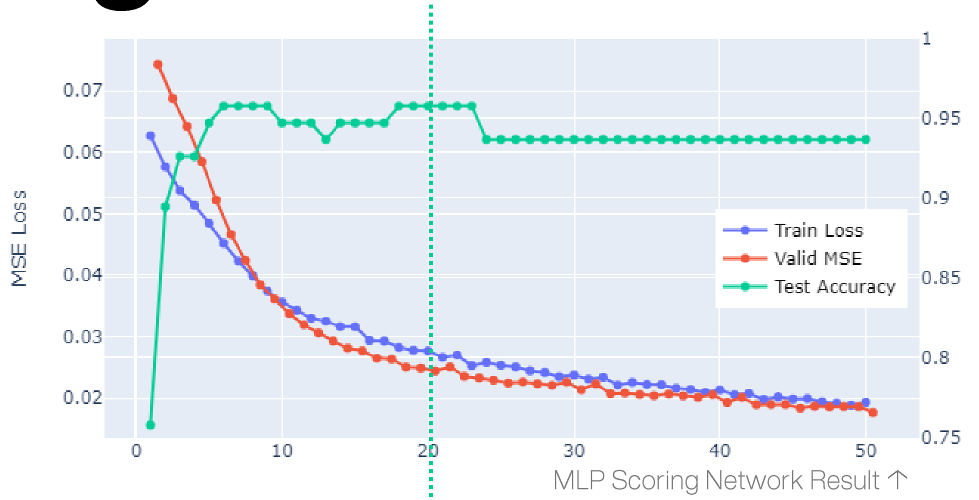
← ResNet18, He et al. 2015

↓ LeNet5, LeCun et al. 1998



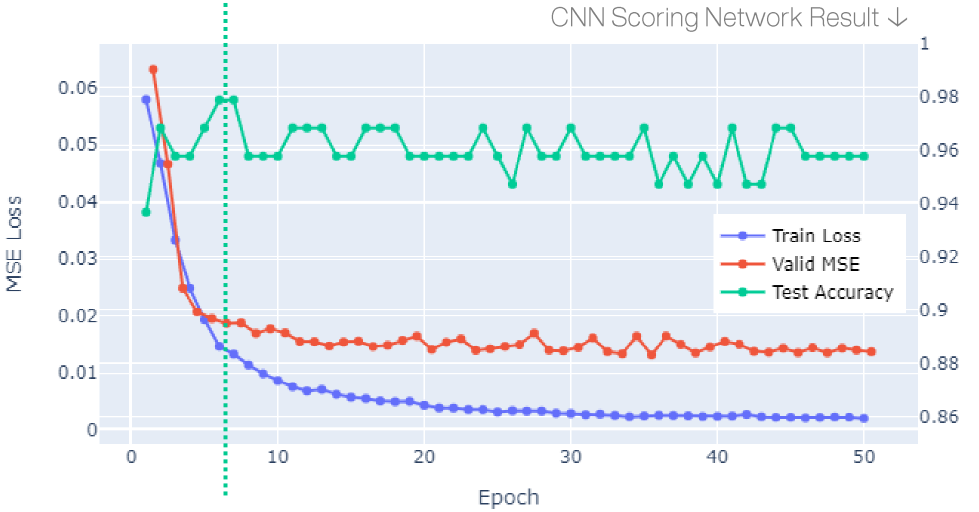
Results: Pointwise Scoring Network

- MLP (extract <bos> embedding vector)
  - `lr=1e-4 batch_size=64`
  - Reach 0.9579 test accuracy after 20 epochs
  - Reach 0.0177 test MSE after 50 epochs
- CNN (LeNet)
  - `lr=1e-4 batch_size=32`
  - Reach 0.9789 test accuracy after 6 epochs
  - Reach 0.0131 test MSE after 35 epochs
- Problem:
  - Test accuracy drops as test MSE decreases
  - Directly fit each MSA with its score
  - Compare scores of MSAs with different query sequence is meaningless!

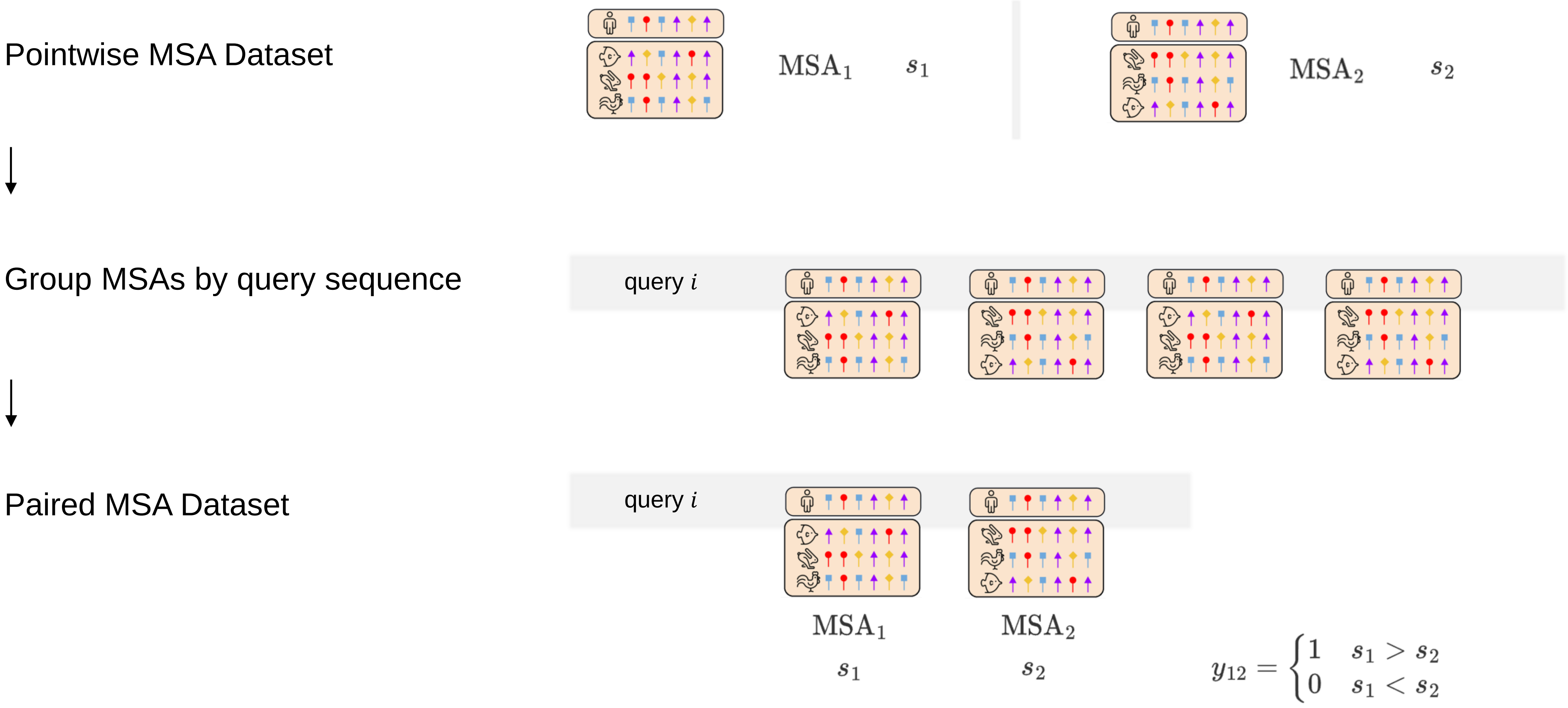


MLP Scoring Network Result ↑

CNN Scoring Network Result ↓



Methodology: Siamese Framework



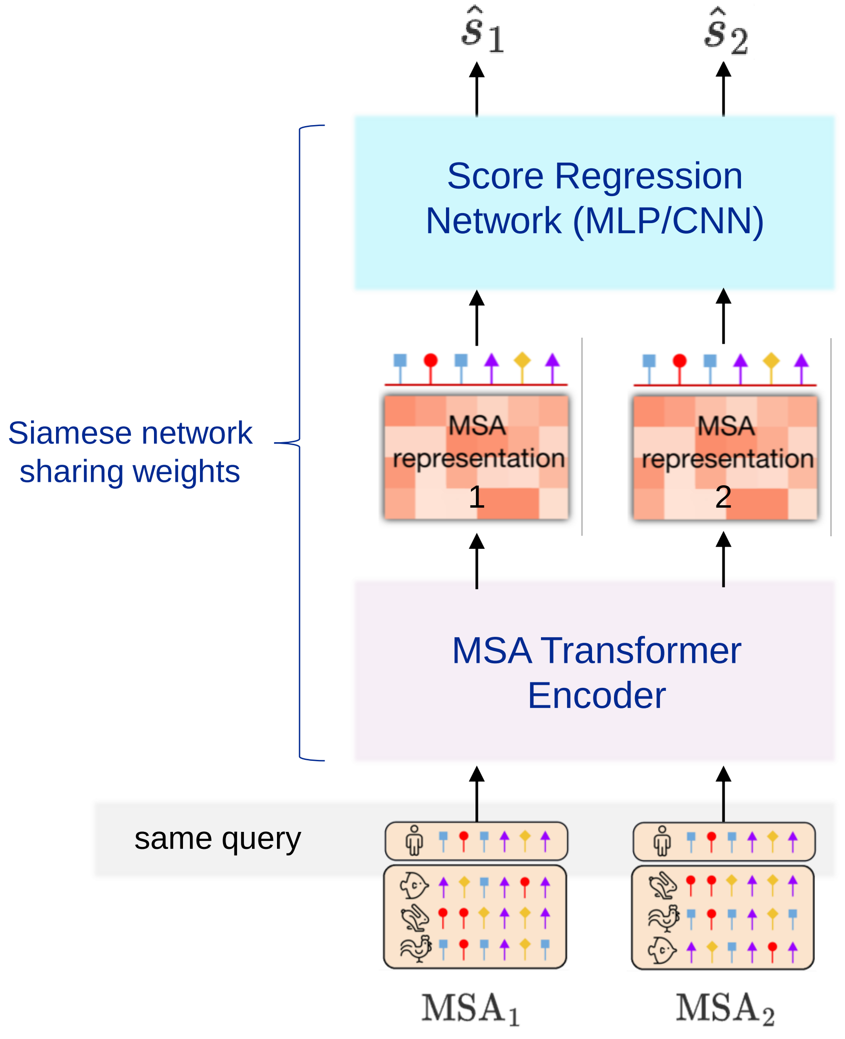
Pointwise MSA Dataset

Group MSAs by query sequence

Paired MSA Dataset

$$y_{12} = \begin{cases} 1 & s_1 > s_2 \\ 0 & s_1 < s_2 \end{cases}$$

Methodology: Siamese Framework



$\hat{P}_{12}$: possibility that MSA_1 is of higher quality than MSA_2

$$\hat{P}_{12} = \mathrm{sigmoid}(\hat{s}_1 - \hat{s}_2) = \frac{1}{1+e^{-(\hat{s}_1-\hat{s}_2)}}$$ [1]

Pairwise Binary Cross Entropy loss:

$$\ell_{\mathrm{BCE}} = -[y_{12} \cdot \log \hat{P}_{12} + (1 - y_{12}) \log(1 - \hat{P}_{12})]$$

$$y_{12} = \begin{cases} 1 & s_1 > s_2 \\ 0 & s_1 < s_2 \end{cases}$$

Mean Squared Error loss:

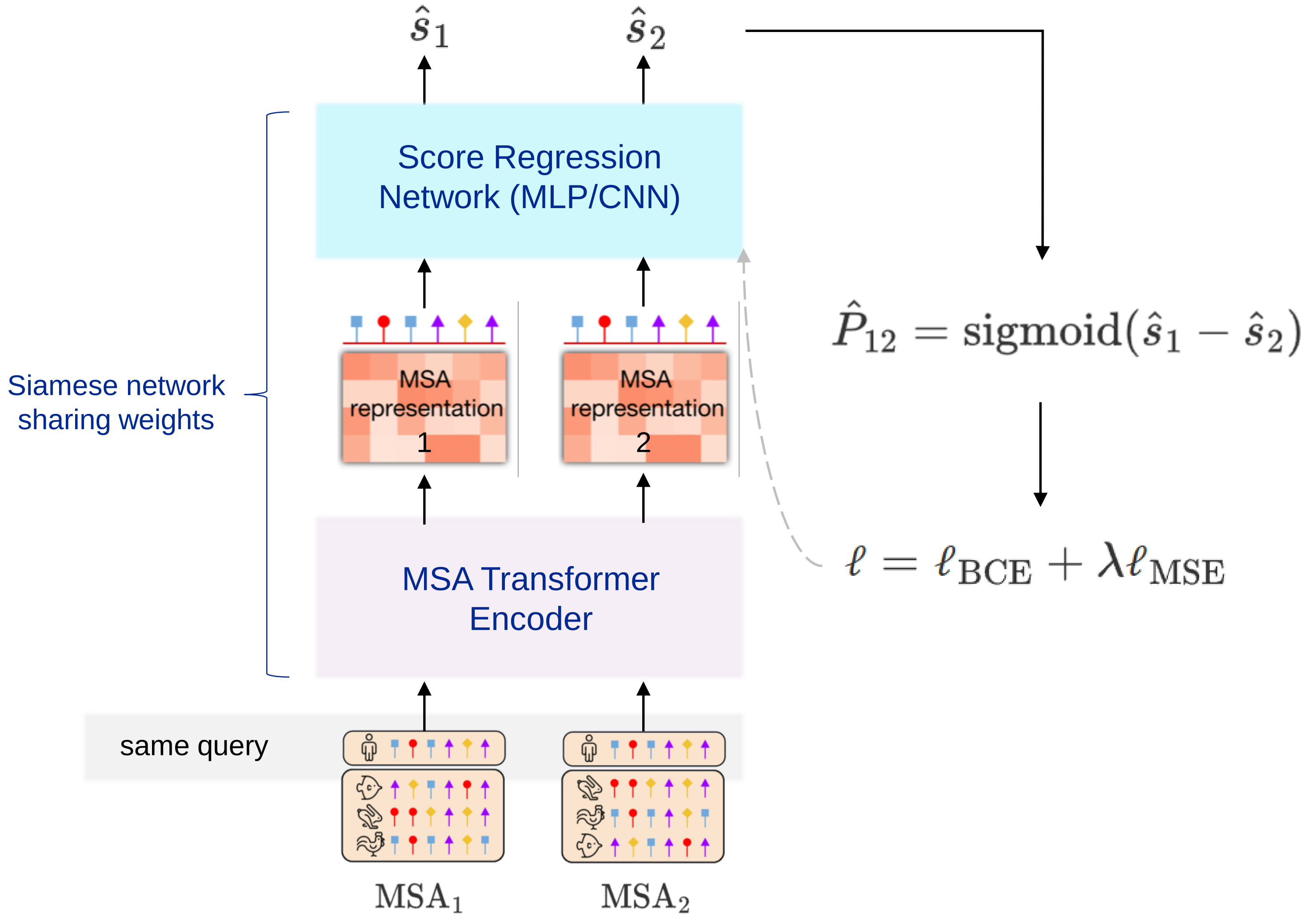
$$\ell_{\mathrm{MSE}} = \frac{1}{2}[(\hat{s}_1 - s_1)^2 + (\hat{s}_2 - s_2)^2]$$

Weighted total loss for back propagation:

$$\ell = \ell_{\mathrm{BCE}} + \lambda \ell_{\mathrm{MSE}}$$

[1] Burges C . From RankNet to LambdaRank to LambdaMART: An Overview[J]. learning, 2010.

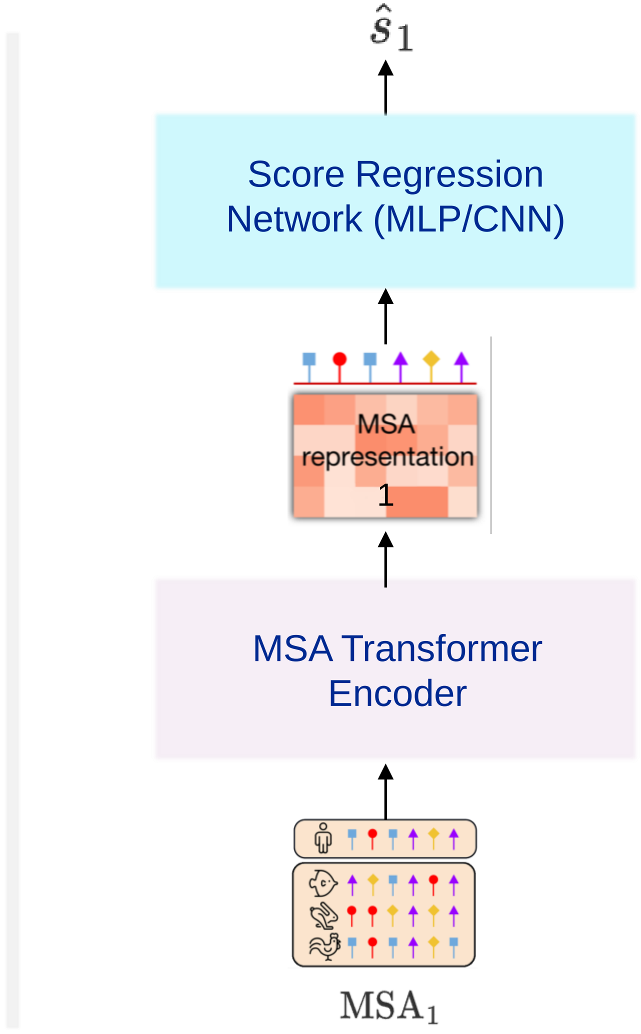
Methodology: Siamese Framework



$$\hat{P}_{12} = \mathrm{sigmoid}(\hat{s}_1 - \hat{s}_2)$$

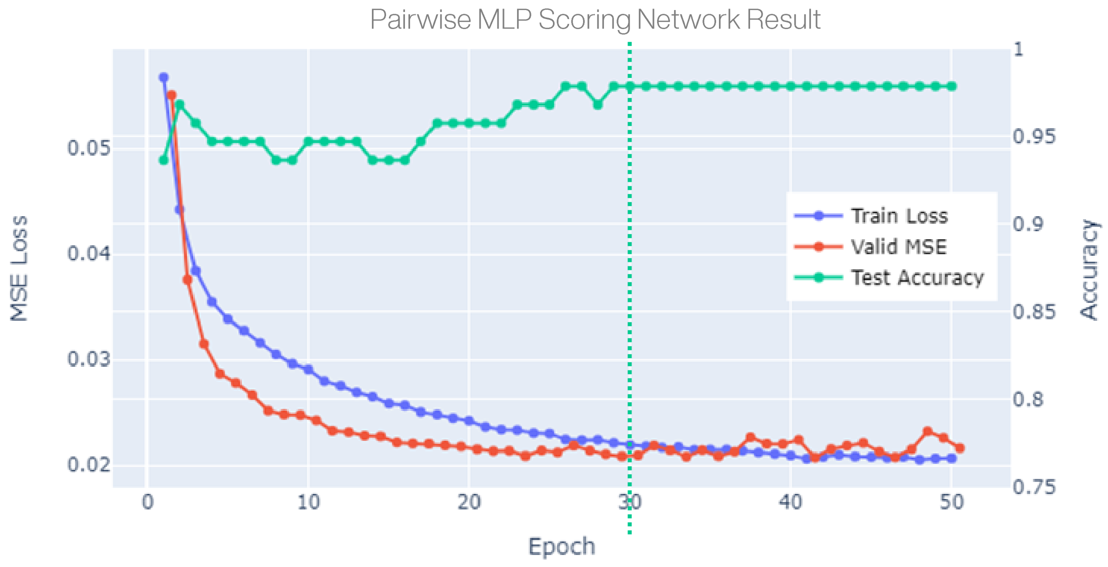
$$\ell = \ell_{\mathrm{BCE}} + \lambda \ell_{\mathrm{MSE}}$$

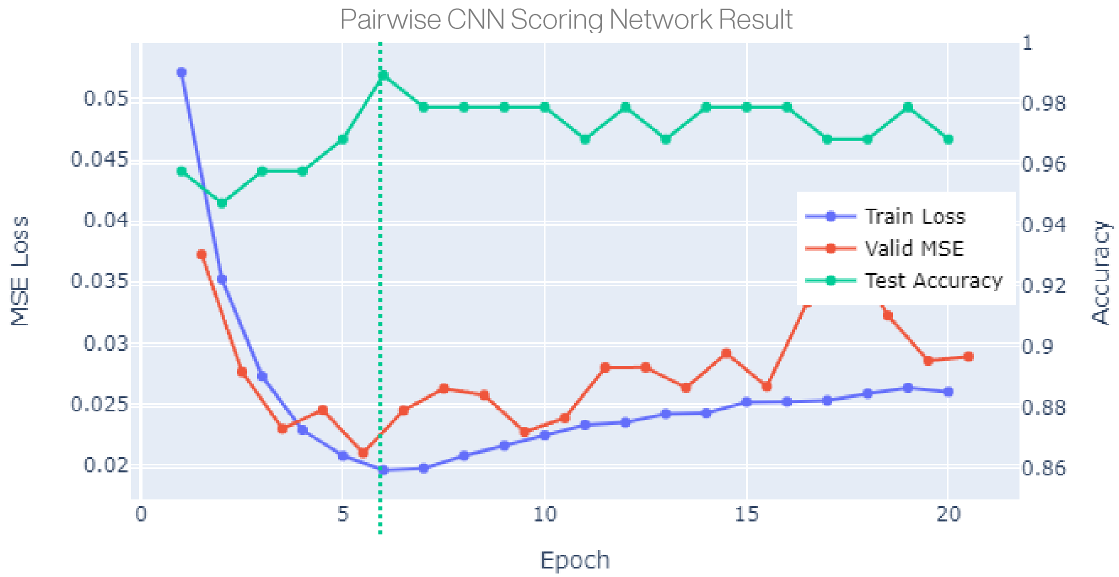
Training stage



Inference stage

Results: Siamese Network + Pairwise Loss



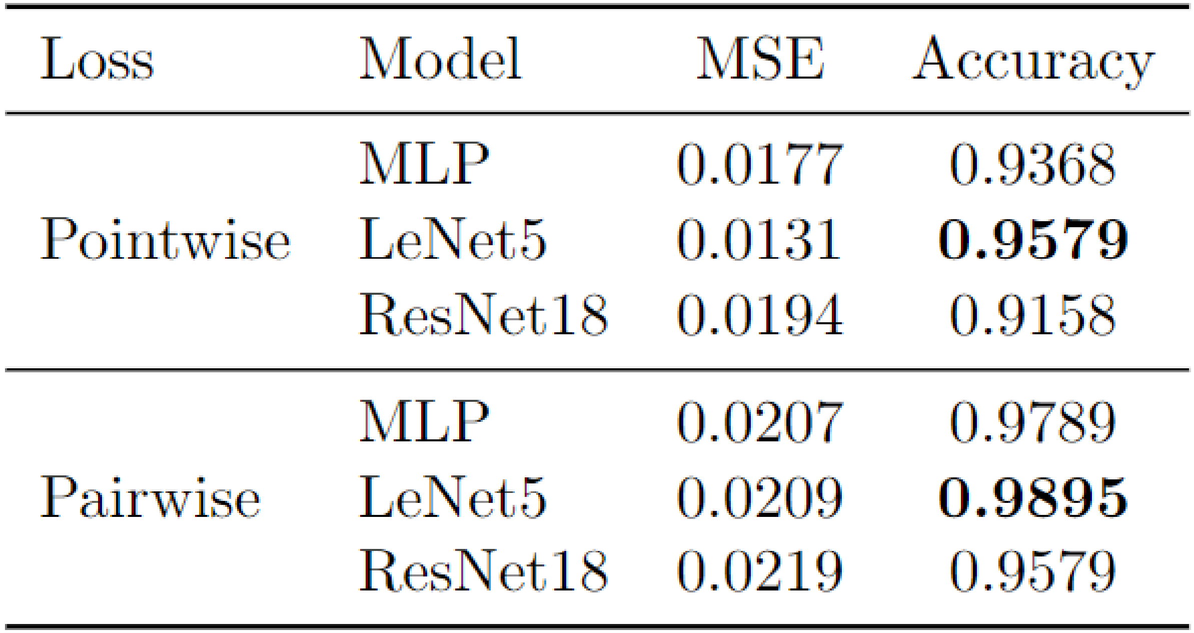


- Pairwise MLP
  - `lr=1e-4 batch_size=64 λ=0.5`
  - Reach `0.9789` test accuracy after `30` epochs
  - Reach `0.0207` test MSE after `41` epochs
- Pairwise CNN
  - `lr=1e-4 batch_size=32 λ=2.5`
  - Reach `0.9895` test accuracy after `6` epochs
  - Reach `0.0210` test MSE after `6` epochs

Results & Conclusion

MLP vs. CNN & Pointwise vs. Pairwise & Metric Reports

Metric Report



Loss	Model	MSE	Accuracy
Pointwise	MLP	0.0177	0.9368
	LeNet5	0.0131	**0.9579**
	ResNet18	0.0194	0.9158
Pairwise	MLP	0.0207	0.9789
	LeNet5	0.0209	**0.9895**
	ResNet18	0.0219	0.9579

* results are at convergence or early-stopped epoch

Ranking Example

- Ground Truth from AlphaFold:

T1024-D1_rand10_fm	T1024-D1_aug_fm	T1024-D1_deduplicated_fm	T1024-D1_meta_fm
0.56865	0.70208	0.88213	0.96243

Ranking Example

- Ground Truth from AlphaFold:

T1024-D1_rand10_fm	T1024-D1_aug_fm	T1024-D1_deduplicated_fm	T1024-D1_meta_fm
0.56865	0.70208	0.88213	0.96243

- Ranked by our pairwise model:

	T1024-D1_rand10_fm		T1024-D1_aug_fm		T1024-D1_deduplicated_fm		T1024-D1_meta_fm
CNN	0.40373	<	0.79878	<	0.85233	<	0.95667
MLP	0.57246	<	0.63600	<	0.98783	>	0.95985

Contributions

- **Transfer Learning** Application of `MSA-Transformer` for `AlphaFold`
- Apply Image Recognition CNNs to MSA Embedding Feature Map
  - eg. `LeNet5`, `ResNet18`
- Combine with `RankNet` and MSA Scoring/Ranking
  - Design pairwise dataset and pairwise loss to compare a pair of MSA inputs of the same reference sequence
  - For the reference sequence and several MSA inputs, possible to rank MSA inputs for better `AlphaFold` prediction

Thanks for your Attention!

MSA Scoring & Ranking Based on MSA Transformer

Group lianyh, tengyue & yangxch