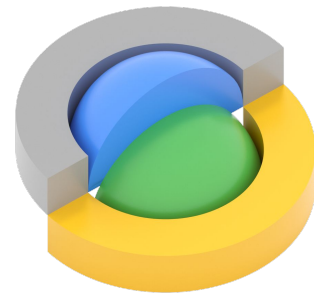


Performers

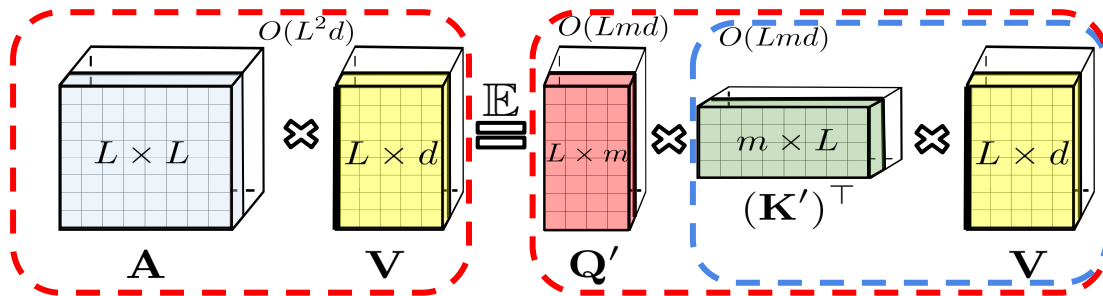


First Linear Softmax-Attention Transformers

Krzysztof Choromanski, Valerii Likhoshesterov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, Adrian Weller, Vikas Sindhvani



["Rethinking Attention with Performers"](#) [Google AI Blog Post](#)



Why we need better Memorization & Attention in ML?

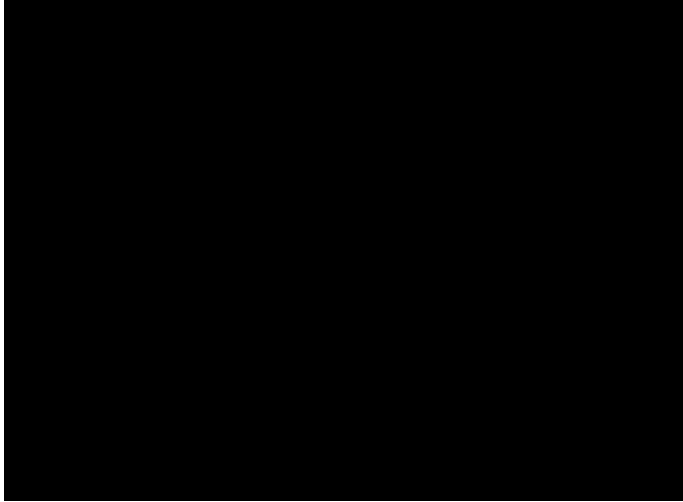


Fig. 1 DeepMind policy navigating simply by “sight”.

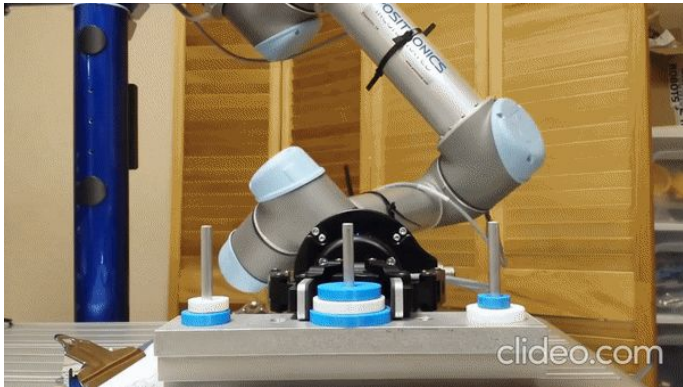


Fig. 2 Robotic arm “solving” Hanoi towers.

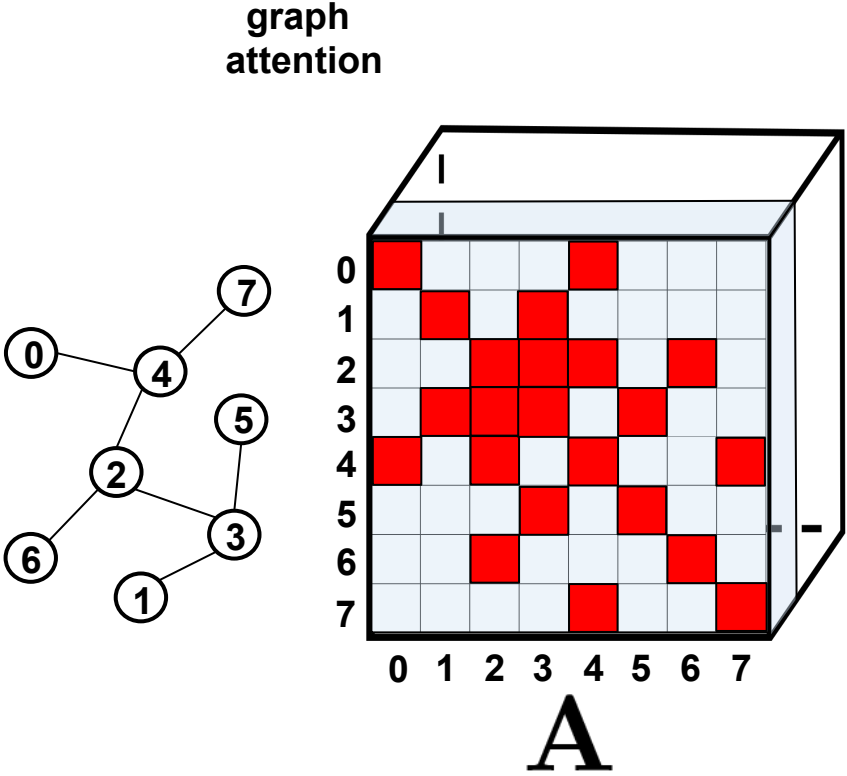
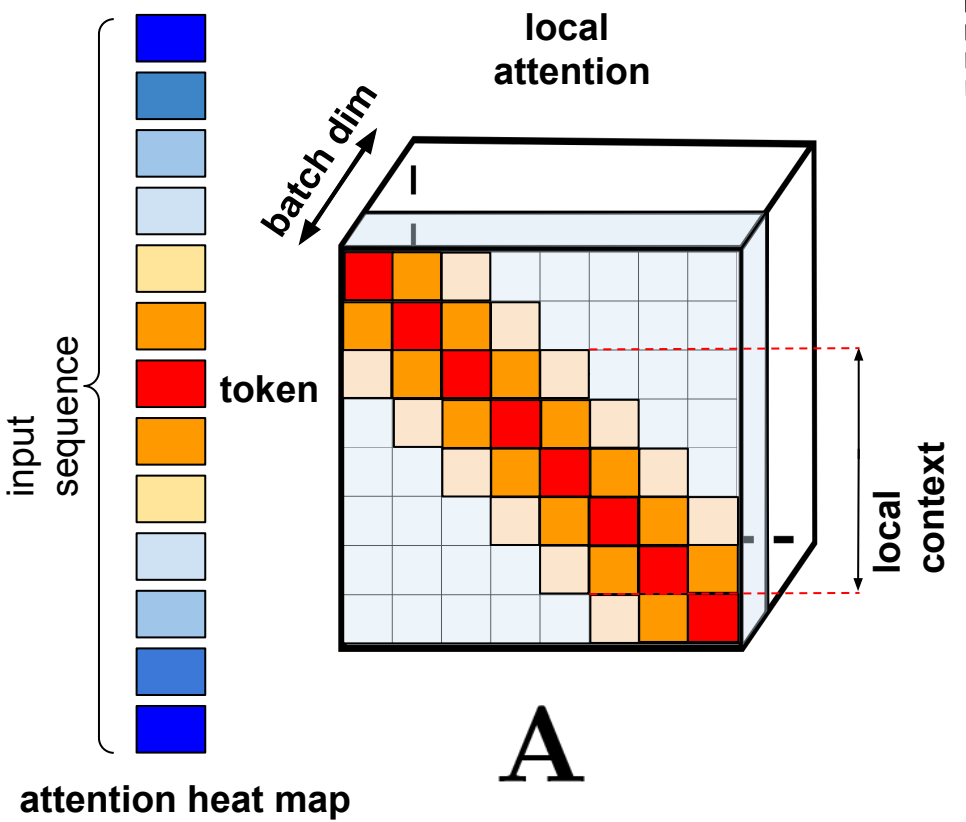
temporal attention

- *Lifelong Learning* requires going beyond purely-reactive Robotics tasks:
 - *“Developmental Robotics: A Complex Dynamical System with Several Spatiotemporal scales.”*
- Memory is a key to AI and currently existing sequential recurrent architectures fail to memorize well.
- Lets learn how to attend to the world - is **Attention** all you need ? Different attention dimensions: **spatial & temporal**.
- Standard attention mechanisms are effectively parallelizable and avoid catastrophic forgetting, but are not scalable.
 - *“It uses more memory and more computation per real interaction...”* [DeepMind nav by sight]
- Lifelong Learning Robotics requires **long range contexts** with **no attention priors**.

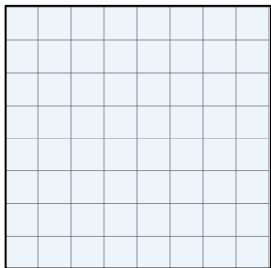


Performer's Backbone: FAVO(R)+

What we Do NOT DO: Sparsifying Attention

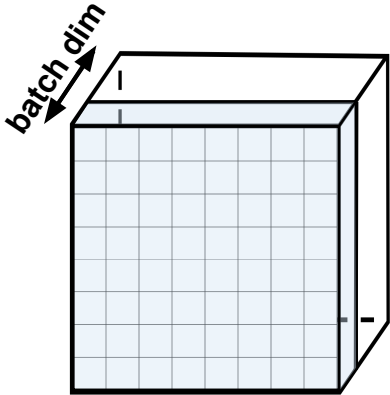


Attention is Kernelizable - A Tale of Random Maps



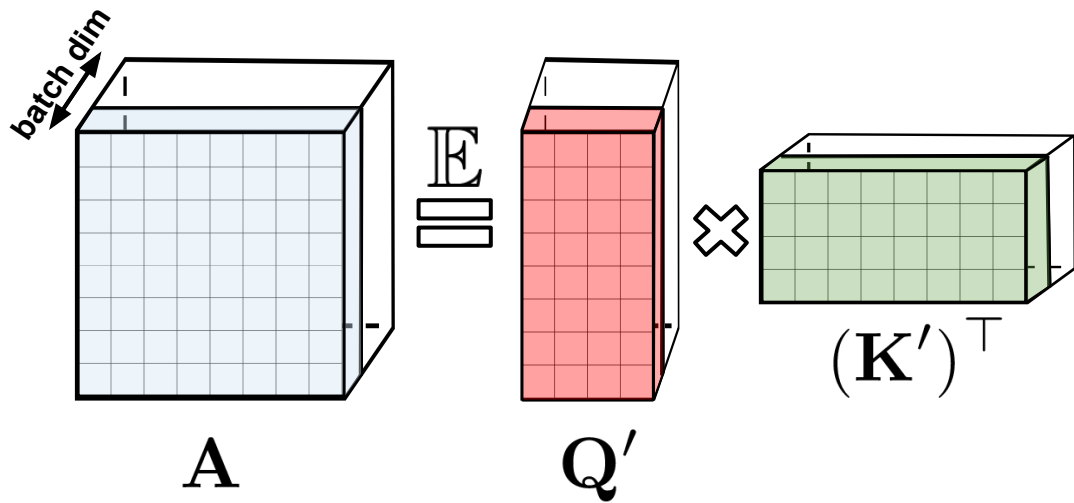
A

Attention is Kernelizable - A Tale of Random Maps

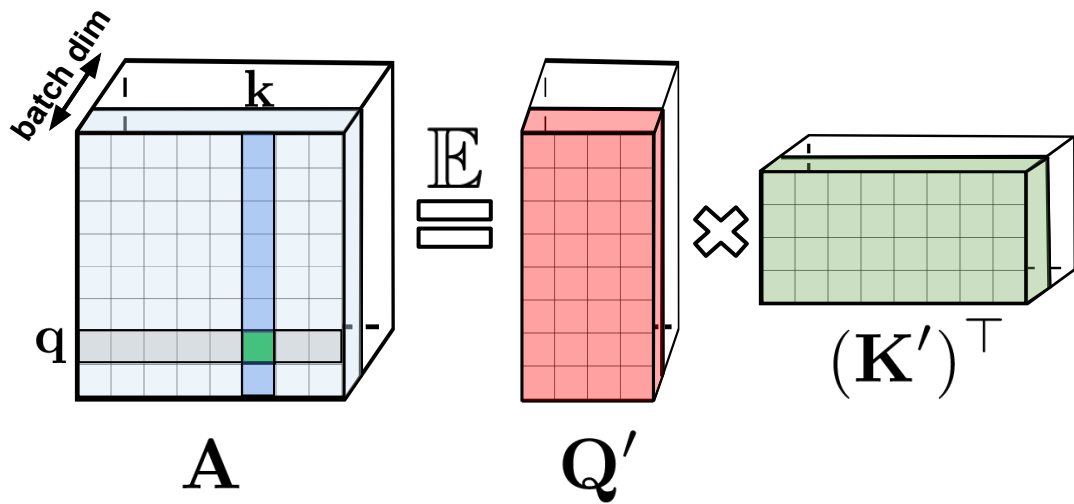


A

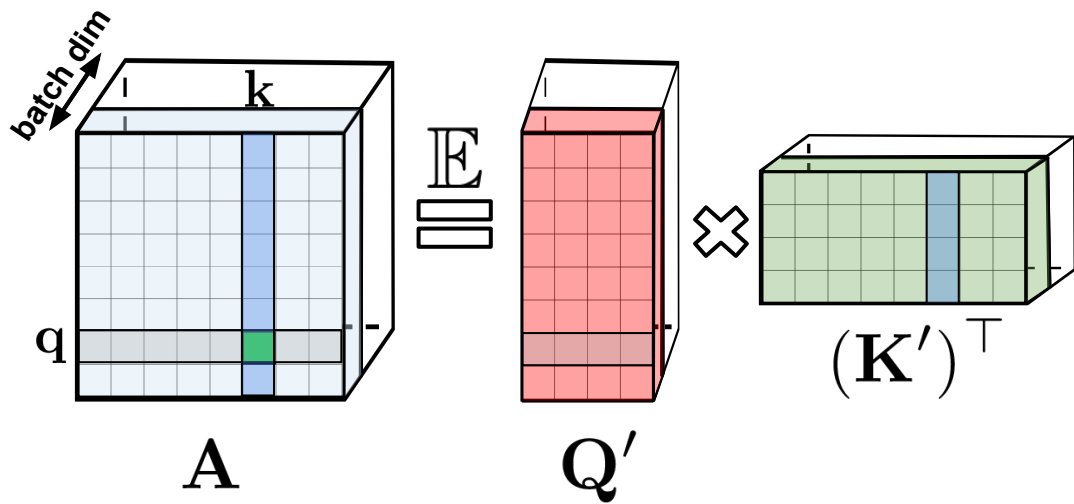
Attention is Kernelizable - A Tale of Random Maps



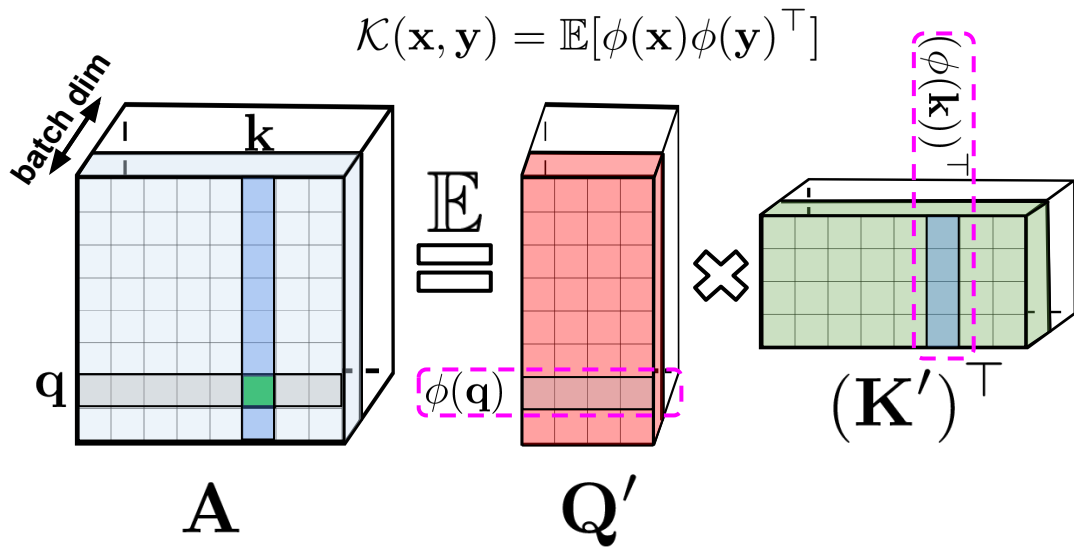
Attention is Kernelizable - A Tale of Random Maps



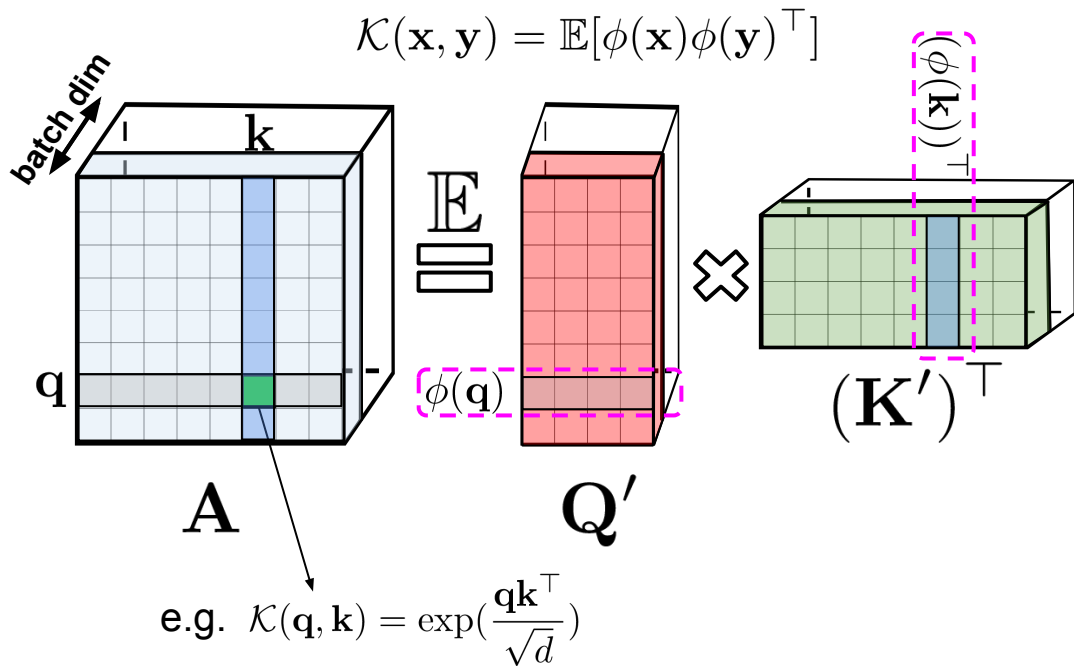
Attention is Kernelizable - A Tale of Random Maps



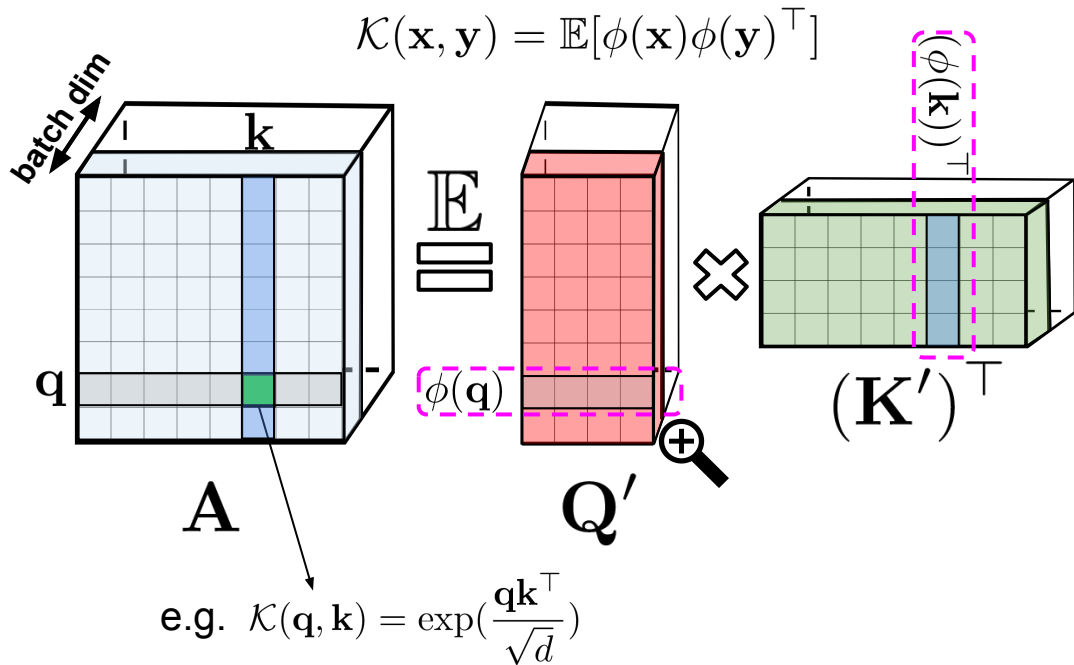
Attention is Kernelizable - A Tale of Random Maps



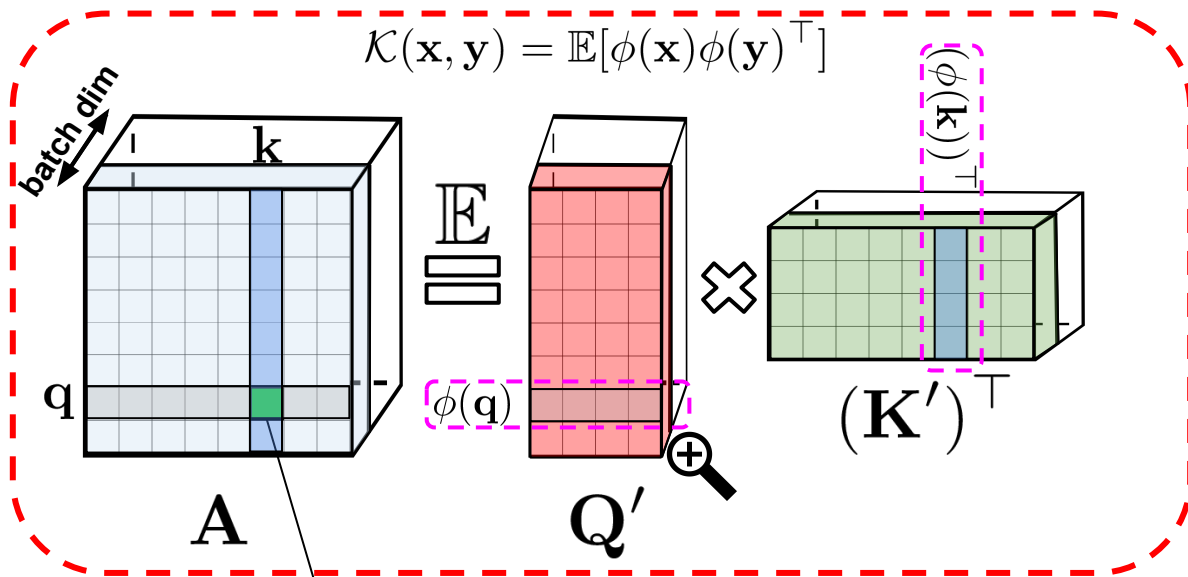
Attention is Kernelizable - A Tale of Random Maps



Attention is Kernelizable - A Tale of Random Maps



Attention is Kernelizable - A Tale of Random Maps



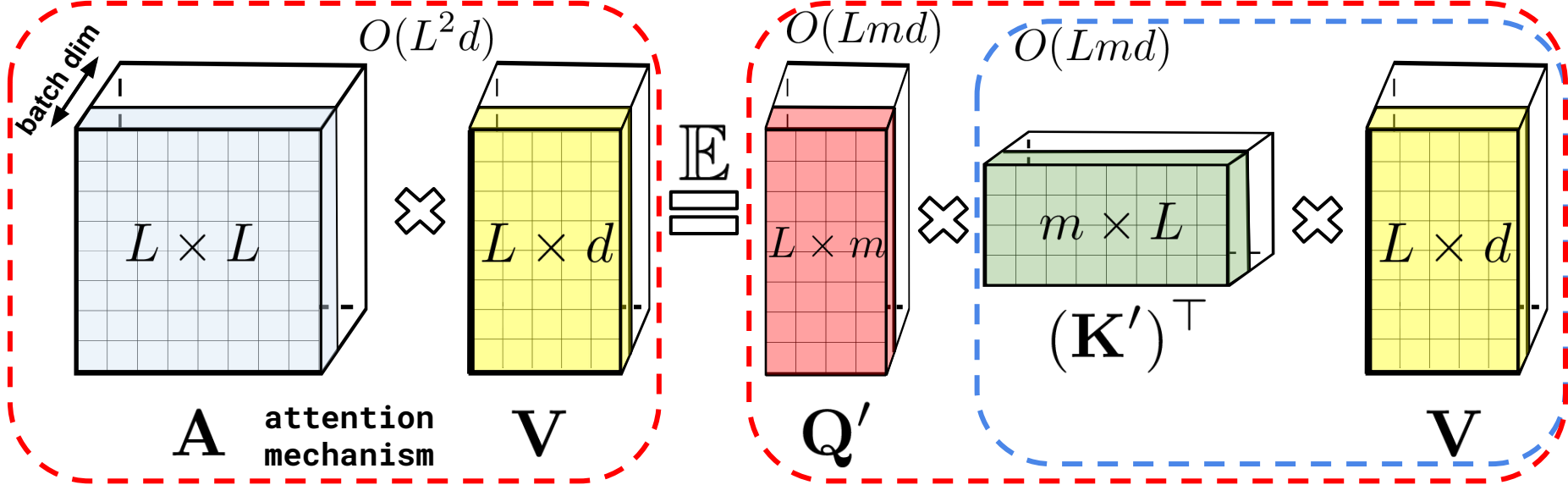
e.g. $\mathcal{K}(\mathbf{q}, \mathbf{k}) = \exp\left(\frac{\mathbf{q}\mathbf{k}^\top}{\sqrt{d}}\right)$

$$\left\{ \begin{array}{l} \phi(\mathbf{x}) = \frac{1}{\sqrt{m}} \underbrace{\exp\left(\frac{\|\mathbf{x}\|_2^2}{2\sqrt{d}}\right)}_{\text{deterministic feature}} \cdot \underbrace{\left(\sin\left(\frac{\mathbf{x}}{d^{1/4}}\omega_1^\top\right), \dots, \sin\left(\frac{\mathbf{x}}{d^{1/4}}\omega_m^\top\right), \cos\left(\frac{\mathbf{x}}{d^{1/4}}\omega_1^\top\right), \dots, \cos\left(\frac{\mathbf{x}}{d^{1/4}}\omega_m^\top\right)\right)}_{\text{random feature}} \\ \omega \sim \mathcal{N}(0, \mathbf{I}_d) \end{array} \right.$$

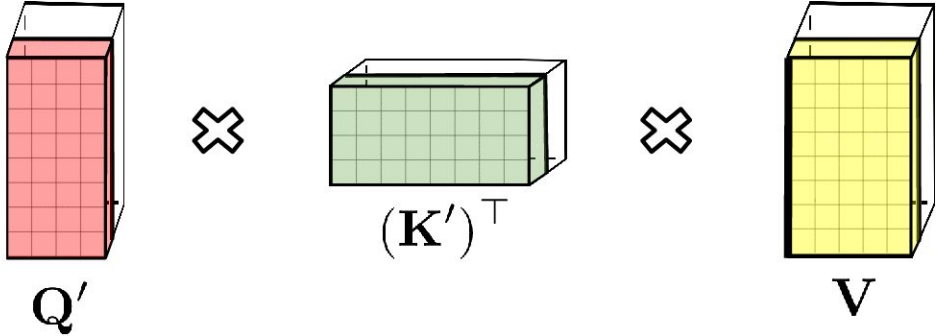
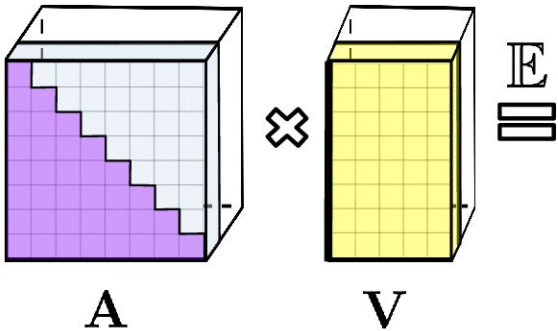
e.g. softmax kernel features

$\phi(\mathbf{q})$

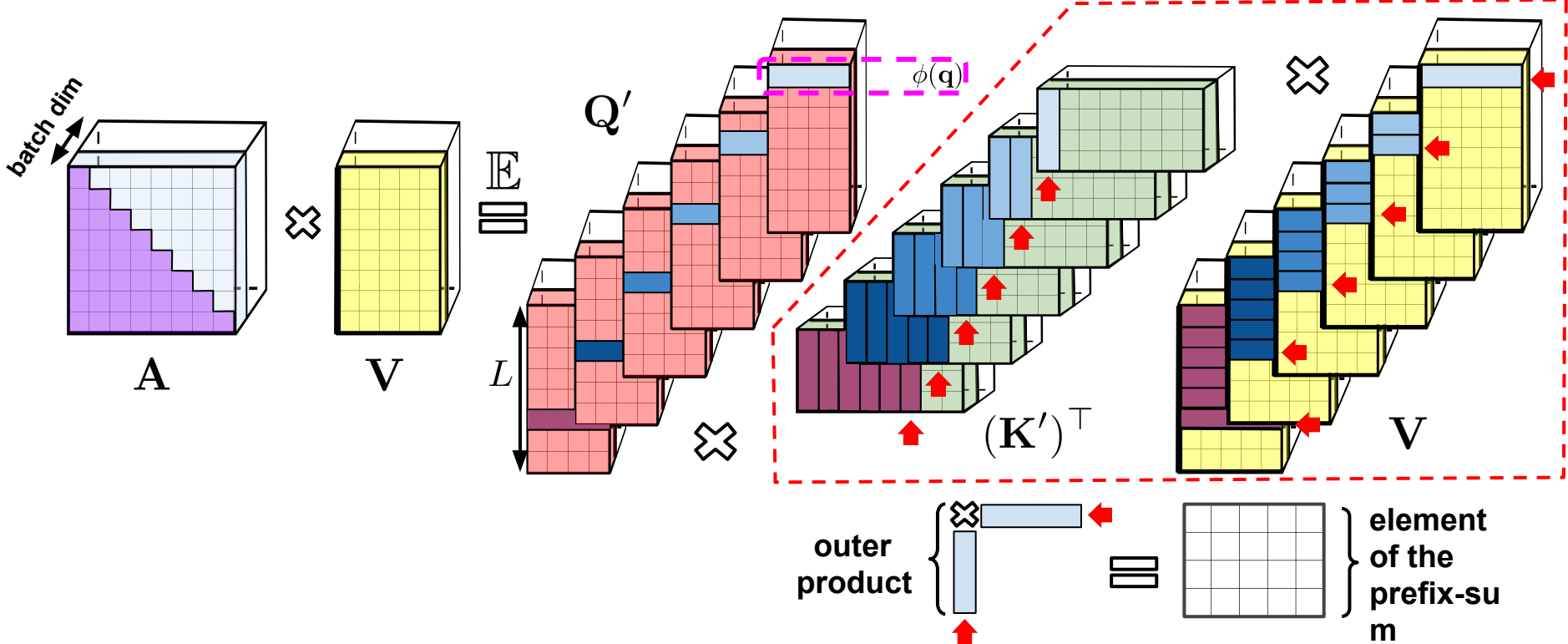
Associativity for Speedups and Space Compression



Causal Transformers as Prefix-Sums Calculators



Causal Transformers as Prefix-Sums Calculators



Strong Uniform Convergence Guarantees

.....

Theorem 1 (Uniform convergence of FAVOR). *Take the generalized attention mechanism defined by $g, h : \mathbb{R}^d \rightarrow \mathbb{R}$ (see: Sec. 2.2) and a radial basis function (RBF) kernel [11] K with corresponding spectral distribution Ω (e.g. Gaussian kernel for which $\Omega = \mathcal{N}(0, \mathbf{I}_d)$). Assume that the rows of matrices \mathbf{Q} and \mathbf{K} are taken from a ball $B(R)$ of radius R , centered at 0 (i.e. norms of queries and keys are upper-bounded by R). Define $l = Rd^{-\frac{1}{4}}$ and take $g^* = \max_{\mathbf{x} \in B(l)} |g(\mathbf{x})|$, $h^* = \max_{\mathbf{x} \in B(l)} |h(\mathbf{x})|$. Then for any $\epsilon > 0$, $\delta = \frac{\epsilon}{g^* h^*}$ and the number of random features $M = \Omega(\frac{d}{\delta^2} \log(\frac{4\sigma R}{\delta d^{\frac{1}{4}}}))$ for $\sigma = \mathbb{E}_{\omega \sim \Omega}[\omega^\top \omega]$ the following holds: $\|\hat{\mathbf{A}} - \mathbf{A}\|_1 \leq \epsilon$ with any constant probability, where $\hat{\mathbf{A}}$ approximates generalized attention matrix via FAVOR with R -ORFs.*

.....

Strong Uniform Convergence Guarantees

.....

Theorem 1 (Uniform convergence of FAVOR). *Take the generalized attention mechanism defined by $g, h : \mathbb{R}^d \rightarrow \mathbb{R}$ (see: Sec. 2.2) and a radial basis function (RBF) kernel [11] K with corresponding spectral distribution Ω (e.g. Gaussian kernel for which $\Omega = \mathcal{N}(0, \mathbf{I}_d)$). Assume that the rows of matrices \mathbf{Q} and \mathbf{K} are taken from a ball $B(R)$ of radius R , centered at 0 (i.e. norms of queries and keys are upper-bounded by R). Define $l = Rd^{-\frac{1}{4}}$ and take $g^* = \max_{\mathbf{x} \in B(l)} |g(\mathbf{x})|$, $h^* = \max_{\mathbf{x} \in B(l)} |h(\mathbf{x})|$. Then for any $\epsilon > 0$, $\delta = \frac{\epsilon}{g^* h^*}$ and the number of random features $M = \Omega(\frac{d}{\delta^2} \log(\frac{4\sigma R}{\delta d^{\frac{1}{4}}}))$ for $\sigma = \mathbb{E}_{\omega \sim \Omega}[\omega^\top \omega]$ the following holds: $\|\hat{\mathbf{A}} - \mathbf{A}\|_1 \leq \epsilon$ with any constant probability, where $\hat{\mathbf{A}}$ approximates generalized attention matrix via FAVOR with R-ORFs.*

.....

Standard uniform convergence from [Recht & Rahimi](#) if iid, but for R-ORFs some fun: [negative dependence](#)

Towards hyperbolic random features

Lemma 1 (Positive Random Features (PRFs) for Softmax). *For $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, $\mathbf{z} = \mathbf{x} + \mathbf{y}$ we have:*

$$\boxed{\text{SM}(\mathbf{x}, \mathbf{y})} = \mathbb{E}_{\omega \sim \mathcal{N}(0, \mathbf{I}_d)} \left[\exp(\omega^\top \mathbf{x} - \frac{\|\mathbf{x}\|^2}{2}) \exp(\omega^\top \mathbf{y} - \frac{\|\mathbf{y}\|^2}{2}) \right] = \Lambda \mathbb{E}_{\omega \sim \mathcal{N}(0, \mathbf{I}_d)} \cosh(\omega^\top \mathbf{z}),$$

where $\Lambda = \exp(-\frac{\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2}{2})$ and \cosh is a hyperbolic cosine.

Towards hyperbolic random features

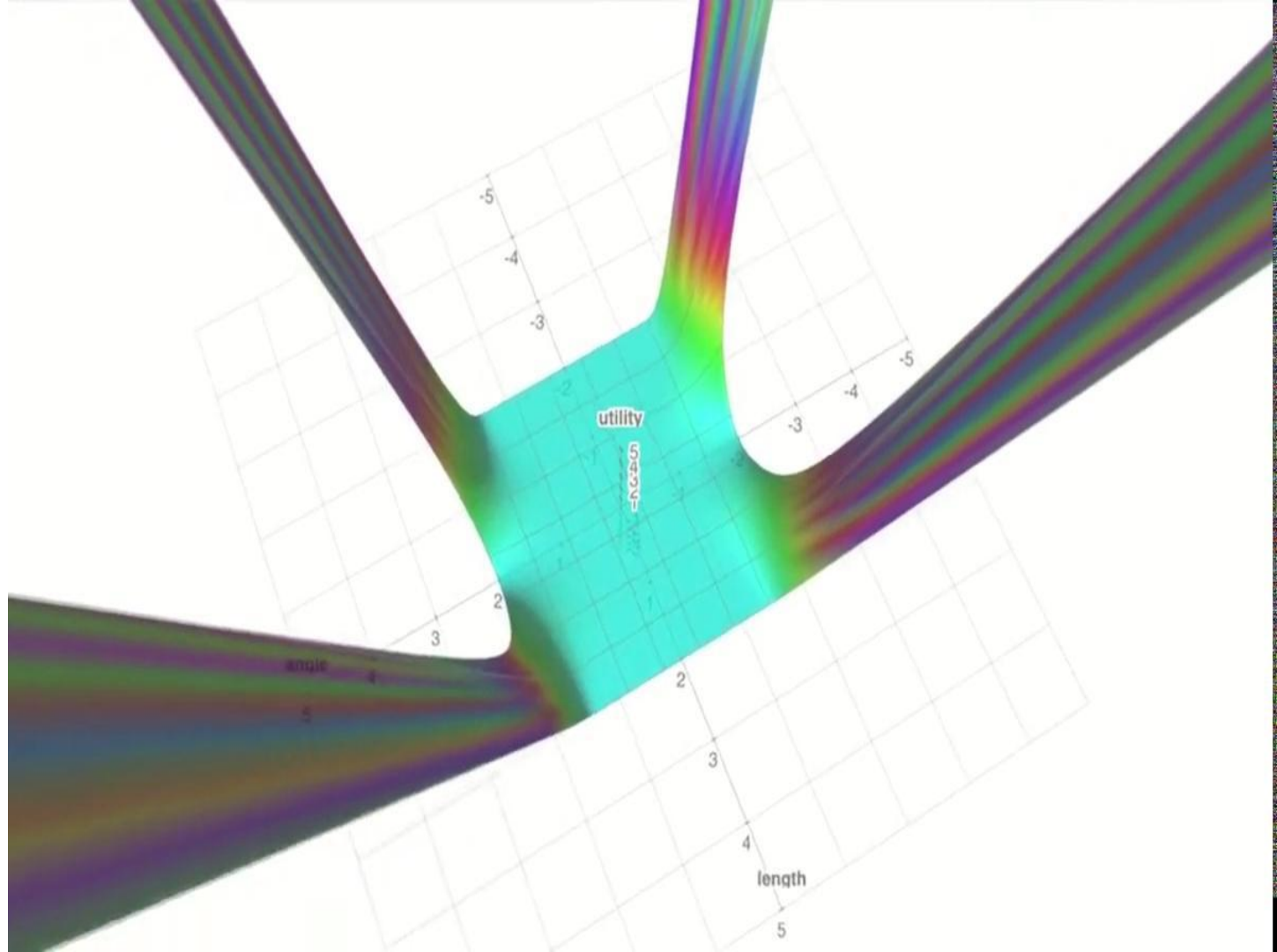
Lemma 1 (Positive Random Features (PRFs) for Softmax). *For $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, $\mathbf{z} = \mathbf{x} + \mathbf{y}$ we have:*

$$\boxed{\text{SM}(\mathbf{x}, \mathbf{y})} = \mathbb{E}_{\omega \sim \mathcal{N}(0, \mathbf{I}_d)} \left[\exp(\omega^\top \mathbf{x} - \frac{\|\mathbf{x}\|^2}{2}) \exp(\omega^\top \mathbf{y} - \frac{\|\mathbf{y}\|^2}{2}) \right] = \Lambda \mathbb{E}_{\omega \sim \mathcal{N}(0, \mathbf{I}_d)} \cosh(\omega^\top \mathbf{z}),$$

where $\Lambda = \exp(-\frac{\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2}{2})$ and \cosh is a hyperbolic cosine. Consequently, softmax-kernel admits a positive random feature map unbiased approximation with $h(\mathbf{x}) = \exp(-\frac{\|\mathbf{x}\|^2}{2})$, $l = 1$, $f_1 = \exp$ and $\mathcal{D} = \mathcal{N}(0, \mathbf{I}_d)$ or: $h(\mathbf{x}) = \frac{1}{\sqrt{2}} \exp(-\frac{\|\mathbf{x}\|^2}{2})$, $l = 2$, $f_1(u) = \exp(u)$, $f_2(u) = \exp(-u)$ and the same \mathcal{D} (the latter for further variance reduction). We call related estimators: $\boxed{\widehat{\text{SM}}_m^+ \text{ and } \widehat{\text{SM}}_m^{\text{hyp}+}}$.

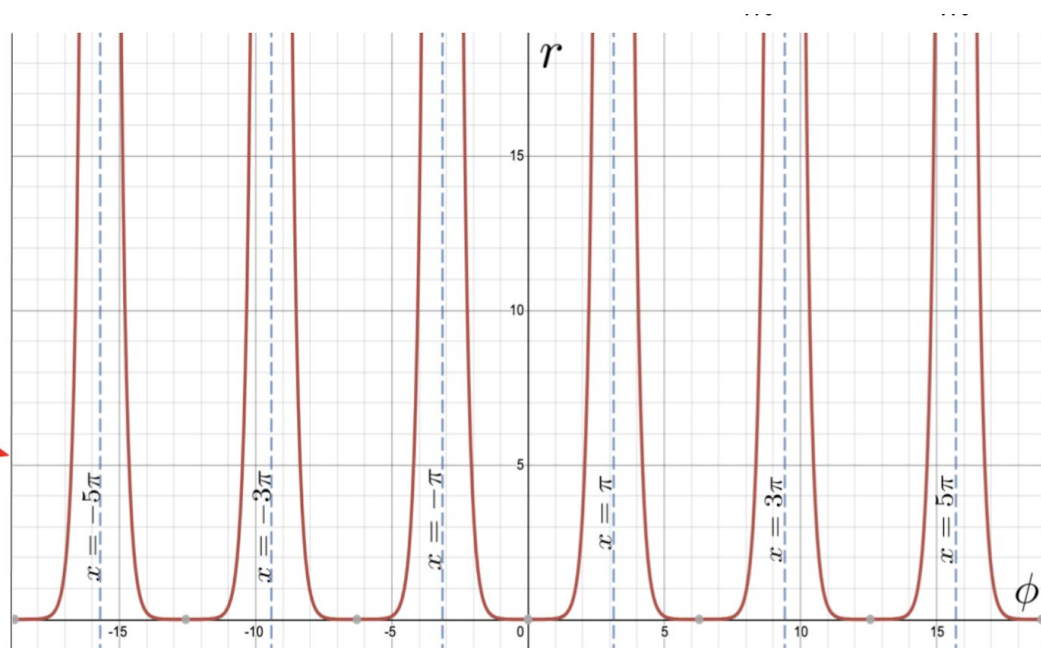
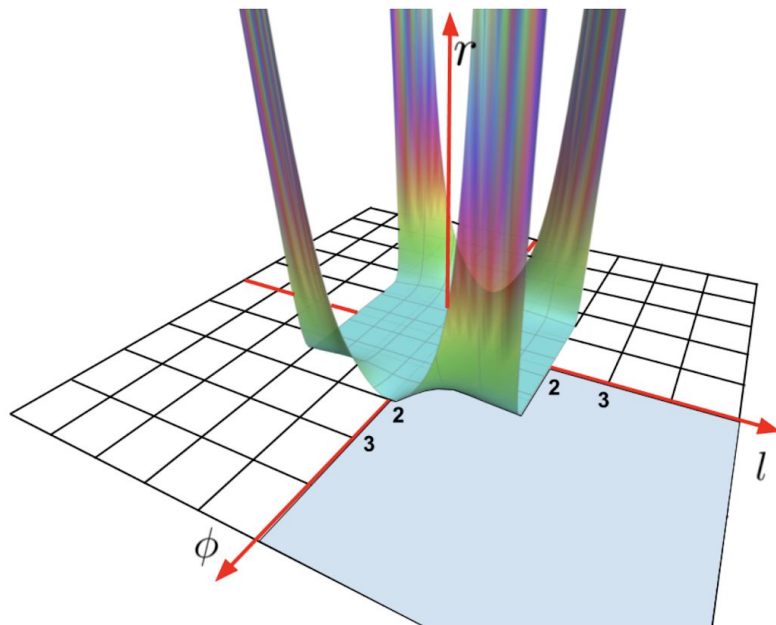
**Do we need to go beyond
trigonometric features ?**

$$\text{utility} = \frac{\text{MSE}(\text{SM}^{\text{trig}}(\mathbf{x}, \mathbf{y}))}{\text{MSE}(\text{SM}^+(\mathbf{x}, \mathbf{y}))}$$



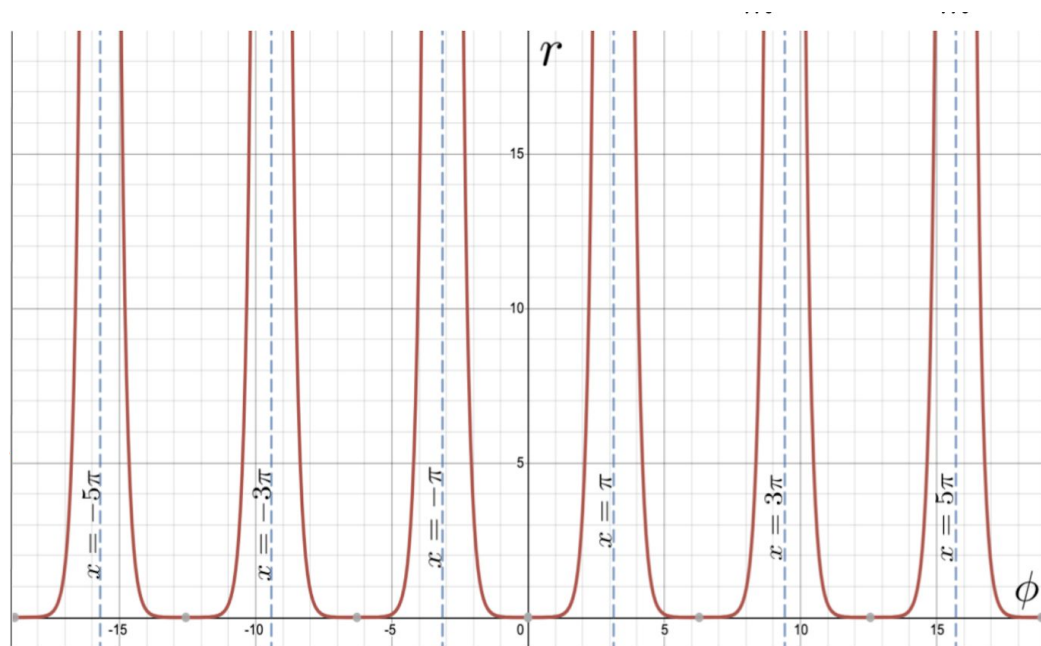
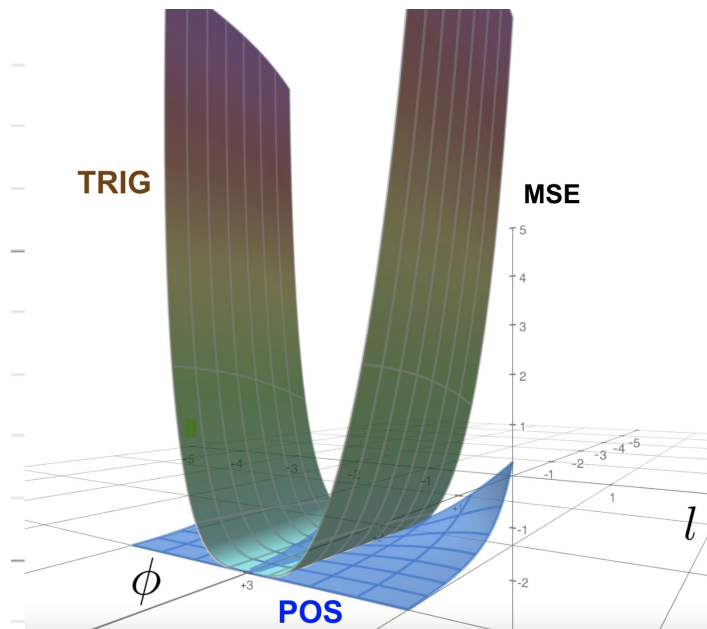
$$\text{length} = \|\mathbf{x}\| = \|\mathbf{y}\|$$

Do we need to go beyond trigonometric features ?



Left: Symmetrized (around origin) utility function r (defined as a ratio of the mean squared errors of estimators built on: trigonometric and positive random features) as a function of the angle ϕ (in radians) between input feature vectors and their lengths l . Larger values indicate regions of (ϕ, l) -space with better performance of positive random features. We see that for critical regions with ϕ large enough (small enough softmax-kernel values) our method is arbitrarily more accurate than trigonometric random features. Plot presented for domain $[-\pi, \pi] \times [-2, 2]$. **Right:** The slice of function r for fixed $l = 1$ and varying angle ϕ .

Do we need to go beyond trigonometric features ?



Left: Comparison of the mean squared errors (MSEs) of the estimators applying trigonometric random features (TRIG) and the one leveraging the mechanism of positive random features (POS) in the region of small softmax-kernel values.

Right: The slice of function r for fixed $l = 1$ and varying angle ϕ .

Positive (hyperbolic) features provide also strong theory

Lemma (positive (hyperbolic) versus trigonometric random features). *The following is true:*

$$\text{MSE}(\widehat{\text{SM}}_m^{\text{trig}}(\mathbf{x}, \mathbf{y})) = \frac{1}{2m} \exp(\|\mathbf{x} + \mathbf{y}\|^2) \text{SM}^{-2}(\mathbf{x}, \mathbf{y}) (1 - \exp(-\|\mathbf{x} - \mathbf{y}\|^2))^2,$$

$$\text{MSE}(\widehat{\text{SM}}_m^+(\mathbf{x}, \mathbf{y})) = \frac{1}{m} \exp(\|\mathbf{x} + \mathbf{y}\|^2) \text{SM}^2(\mathbf{x}, \mathbf{y}) (1 - \exp(-\|\mathbf{x} + \mathbf{y}\|^2)),$$

$$\text{MSE}(\widehat{\text{SM}}_m^{\text{hyp}+}(\mathbf{x}, \mathbf{y})) = \frac{1}{2} (1 - \exp(-\|\mathbf{x} + \mathbf{y}\|^2)) \text{MSE}(\widehat{\text{SM}}_m^+(\mathbf{x}, \mathbf{y})).$$

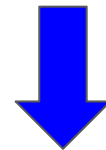
for independent random samples ω_i and where MSE stands for the mean squared error.

Optimal combination: Positive Orthogonal Random Features

Theorem If $\widehat{\text{SM}}_m^{\text{ort}+}(\mathbf{x}, \mathbf{y})$ stands for the version of $\widehat{\text{SM}}_m^+(\mathbf{x}, \mathbf{y})$ with orthogonal random features (and thus for $m \leq d$), then the following holds for any $d > 0$:

$$\text{MSE}(\widehat{\text{SM}}_m^{\text{ort}+}(\mathbf{x}, \mathbf{y})) \leq \text{MSE}(\widehat{\text{SM}}_m^+(\mathbf{x}, \mathbf{y})) - \left(1 - \frac{1}{m}\right) \frac{2}{d+2} \text{SM}^2(\mathbf{x}, \mathbf{y}).$$

Furthermore, completely analogous result holds for the regularized softmax-kernel SMREG.



accuracy gains

Further improvements by regularizing softmax

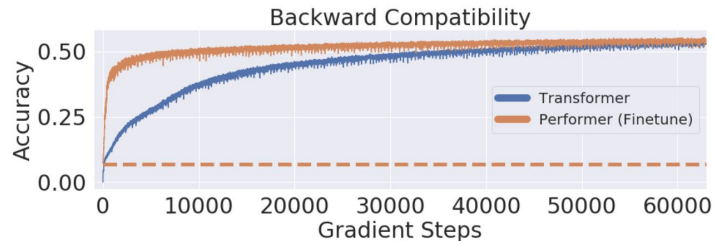
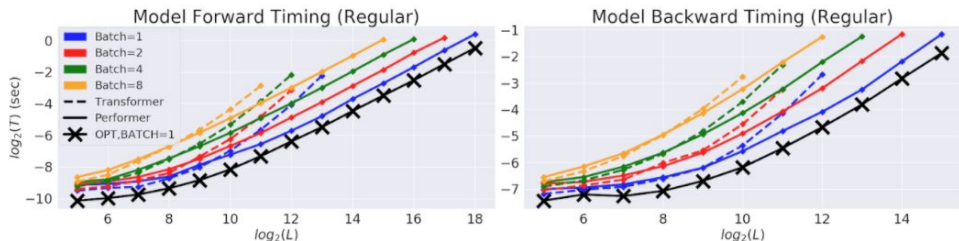
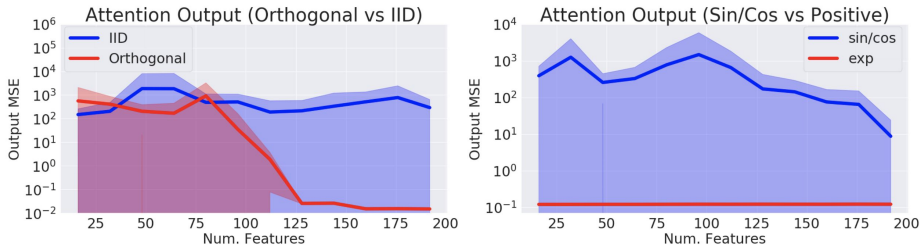
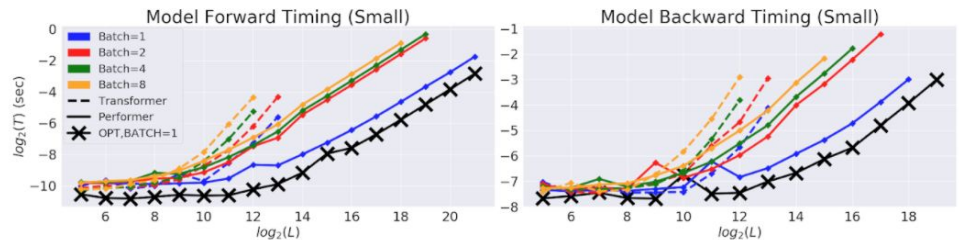
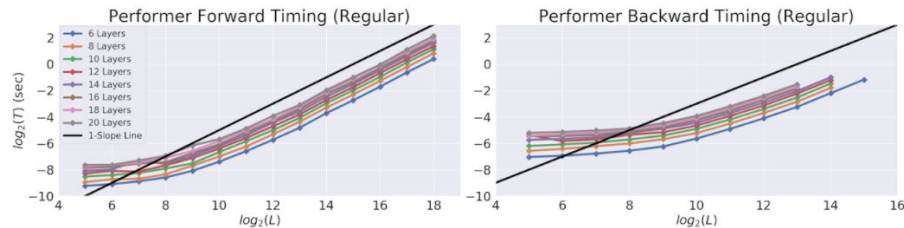
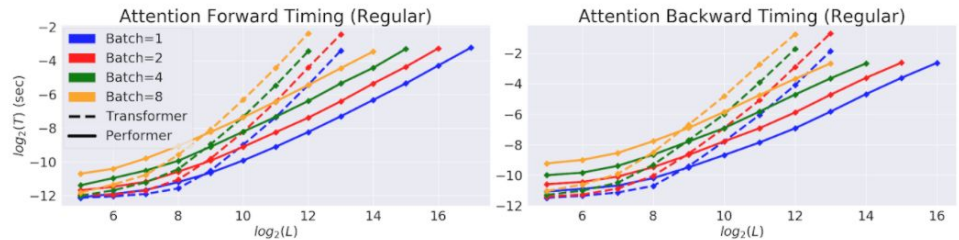
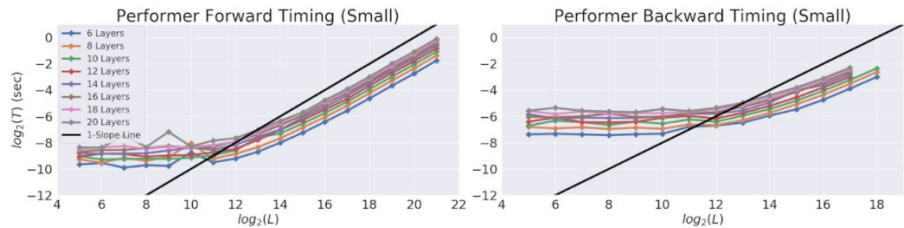
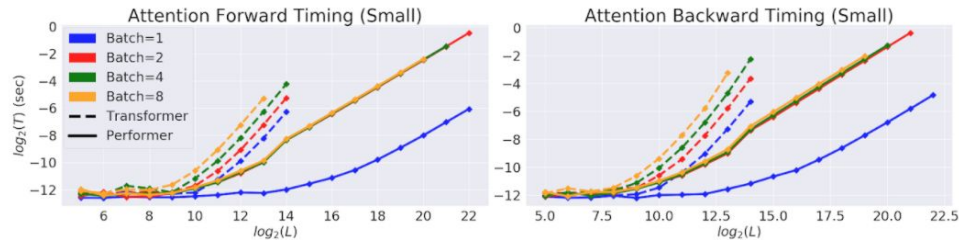
Theorem (regularized versus softmax-kernel). Assume that the L_∞ -norm of the attention matrix for the softmax-kernel satisfies: $\|\mathbf{A}\|_\infty \leq C$ for some constant $C \geq 1$. Denote by \mathbf{A}^{reg} the corresponding attention matrix for the regularized softmax-kernel. The following holds:

$$\inf_{i,j} \frac{\mathbf{A}^{\text{reg}}(i,j)}{\mathbf{A}(i,j)} \geq 1 - \frac{2}{d^{\frac{1}{3}}} + o\left(\frac{1}{d^{\frac{1}{3}}}\right), \text{ and } \sup_{i,j} \frac{\mathbf{A}^{\text{reg}}(i,j)}{\mathbf{A}(i,j)} \leq 1.$$

Furthermore, the latter holds for $d \geq 2$ even if L_∞ -norm condition is not satisfied, i.e. regularized softmax-kernel is a universal lower bound for the softmax-kernel.

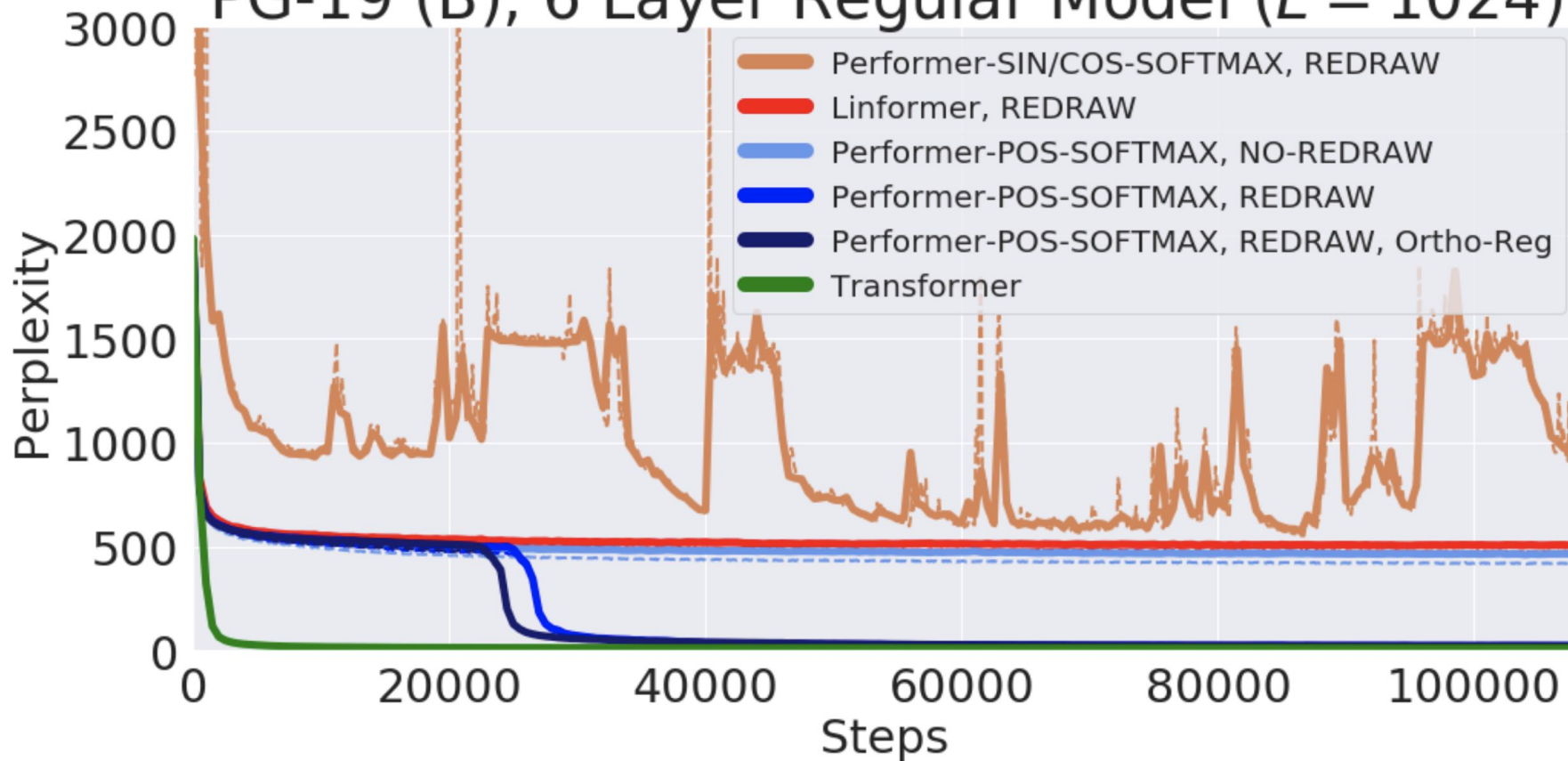


Benchmarking Performers

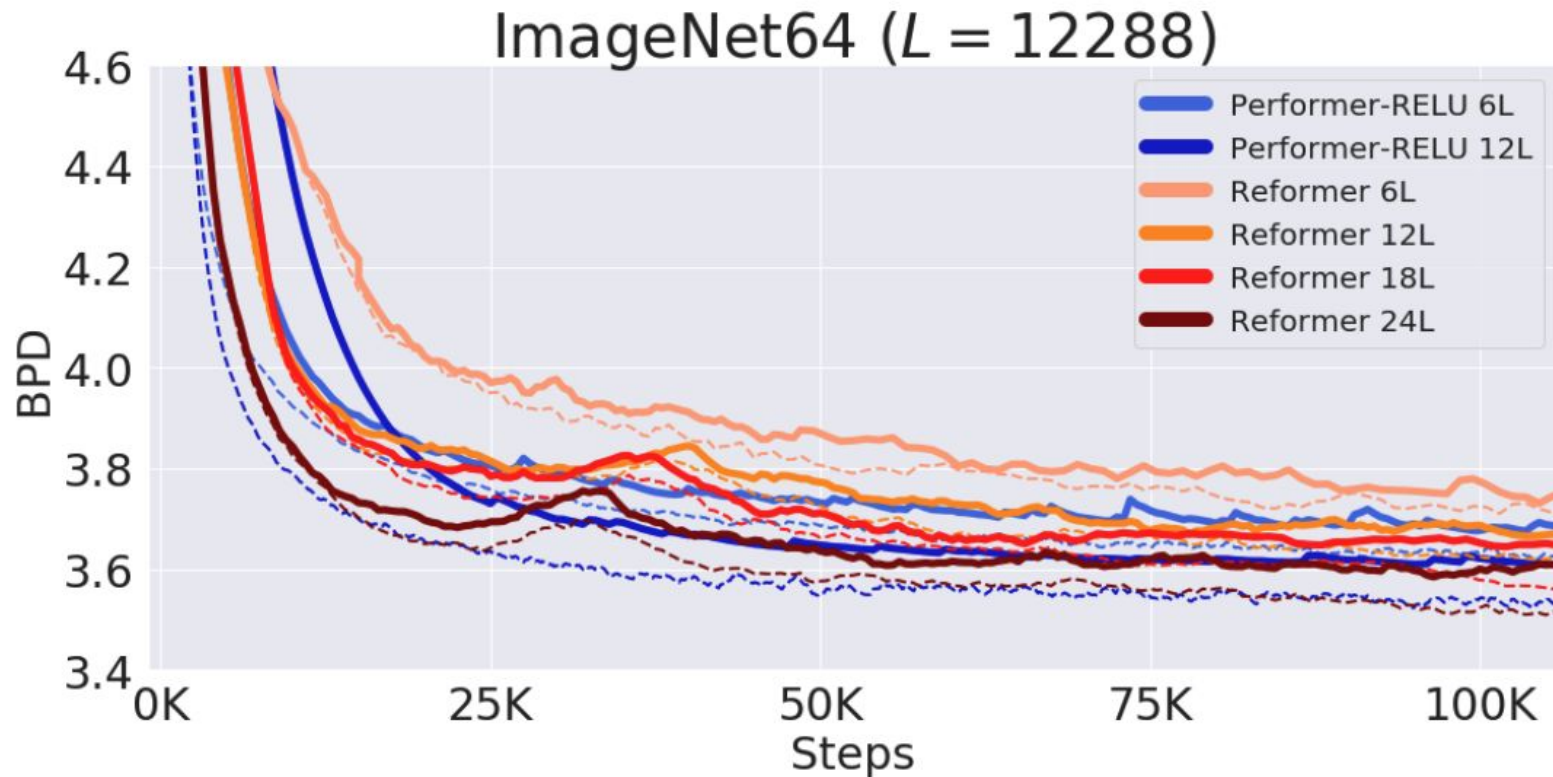


Positive vs trigonometric random features in practice

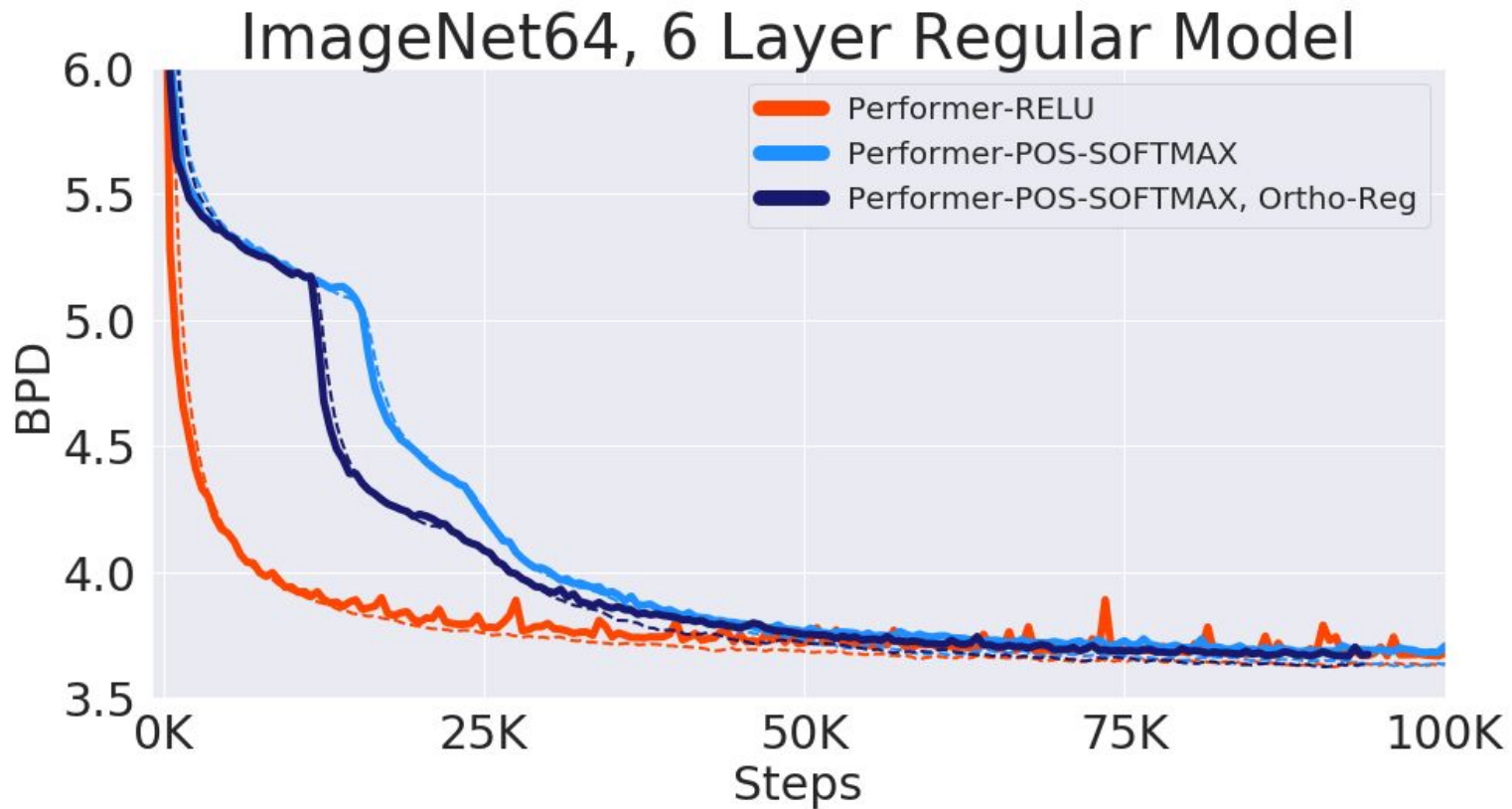
PG-19 (B), 6 Layer Regular Model ($L = 1024$)

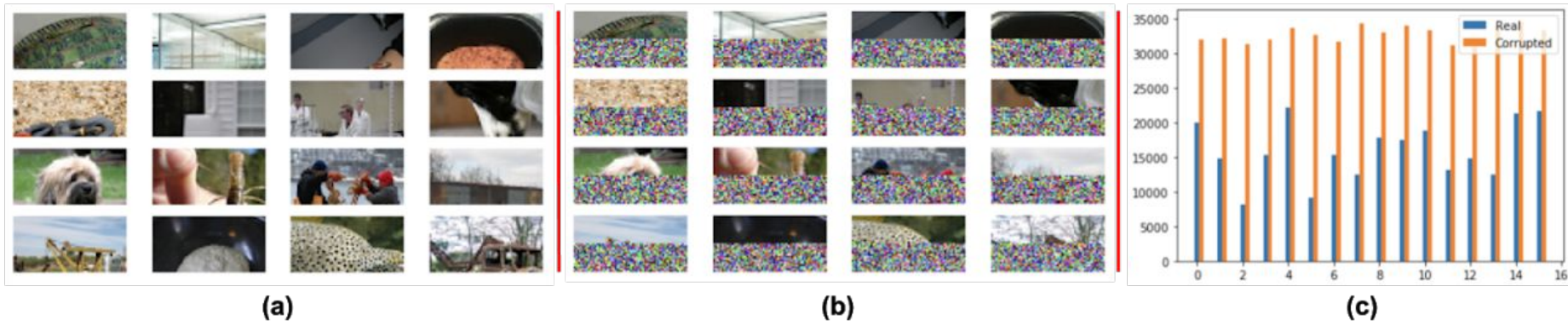


Performers on ImageNet64 - pixel predictions models



Performers on ImageNet64 - Approx. Softmax vs ReLU





(a): ImageNet64 (validation) images

(b): Partially Corrupted ImageNet64 images

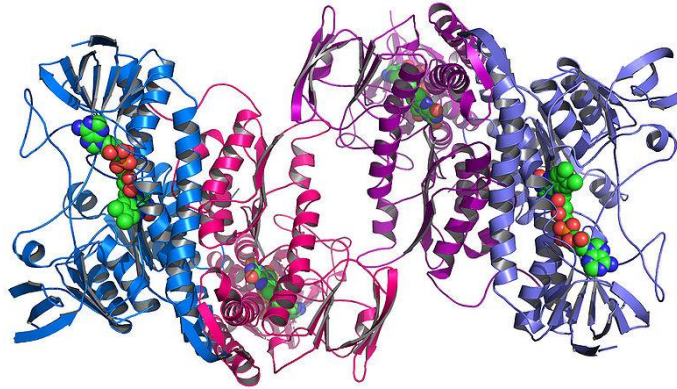
(c): Negative log prob between (a) and (b)

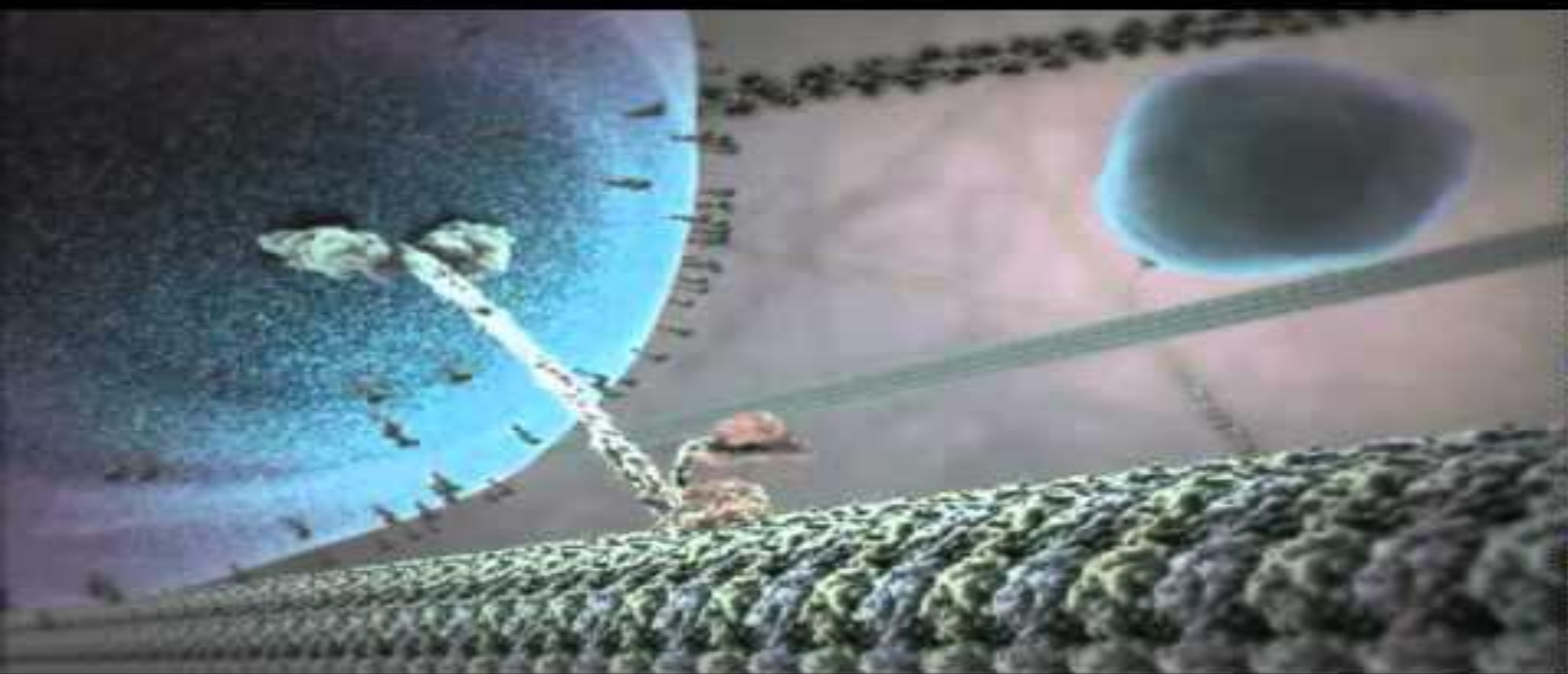
Model correctly distinguishes b/w actual and corrupted images.

Applications

Decoding Protein Language

Aka On the Hunt for Holy Grail of Modern Science...

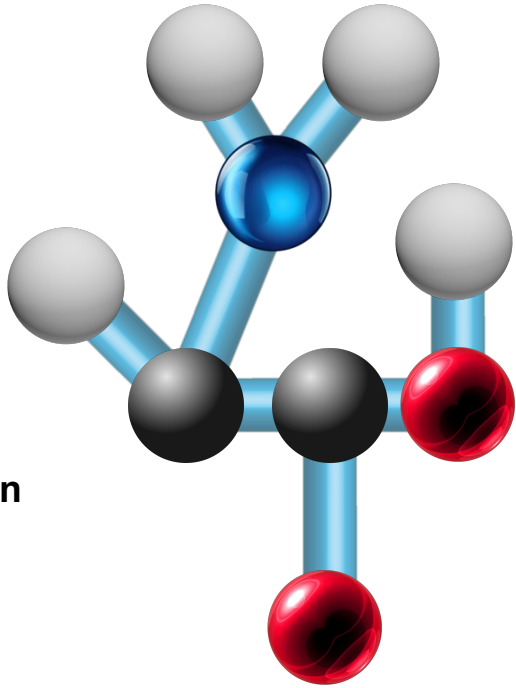




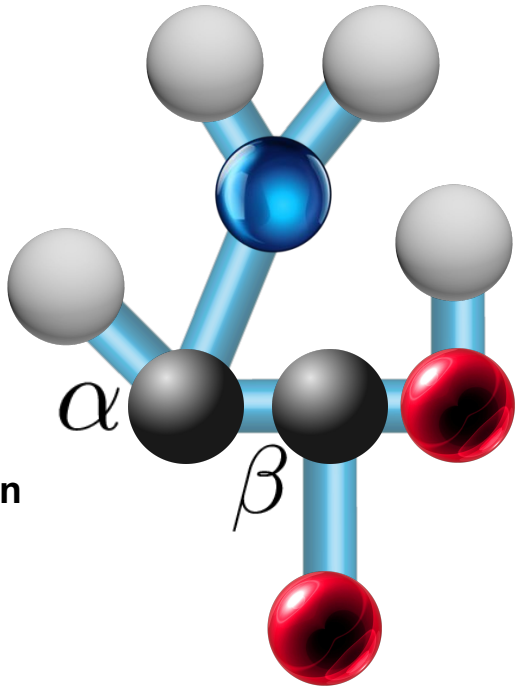
Primary Structure



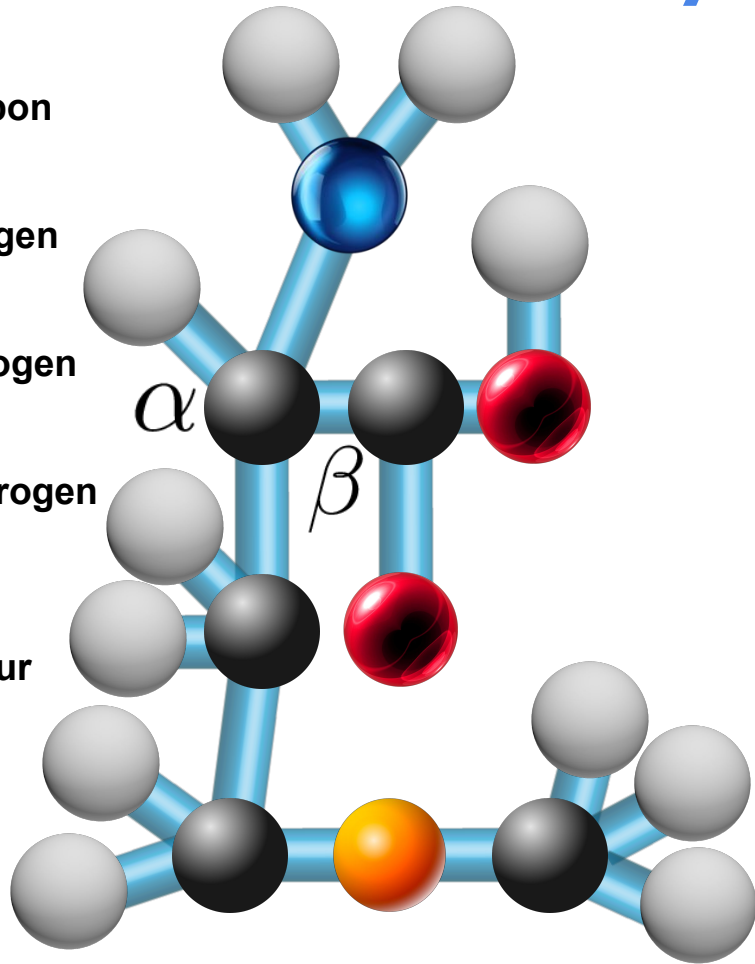
Primary Structure



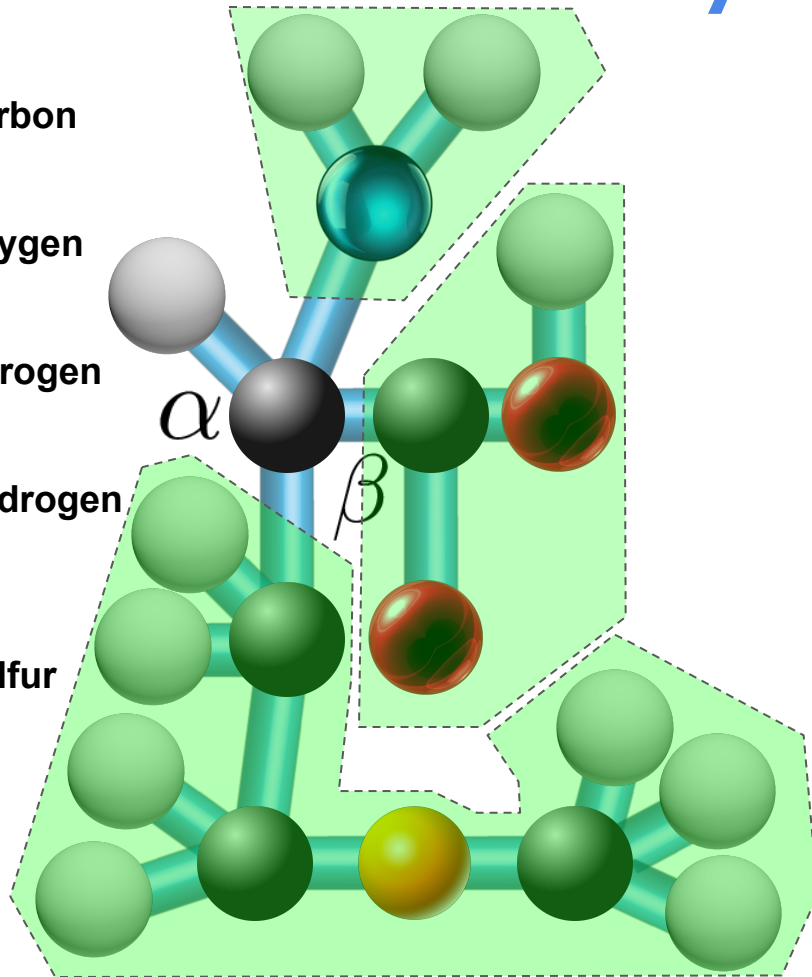
Primary Structure



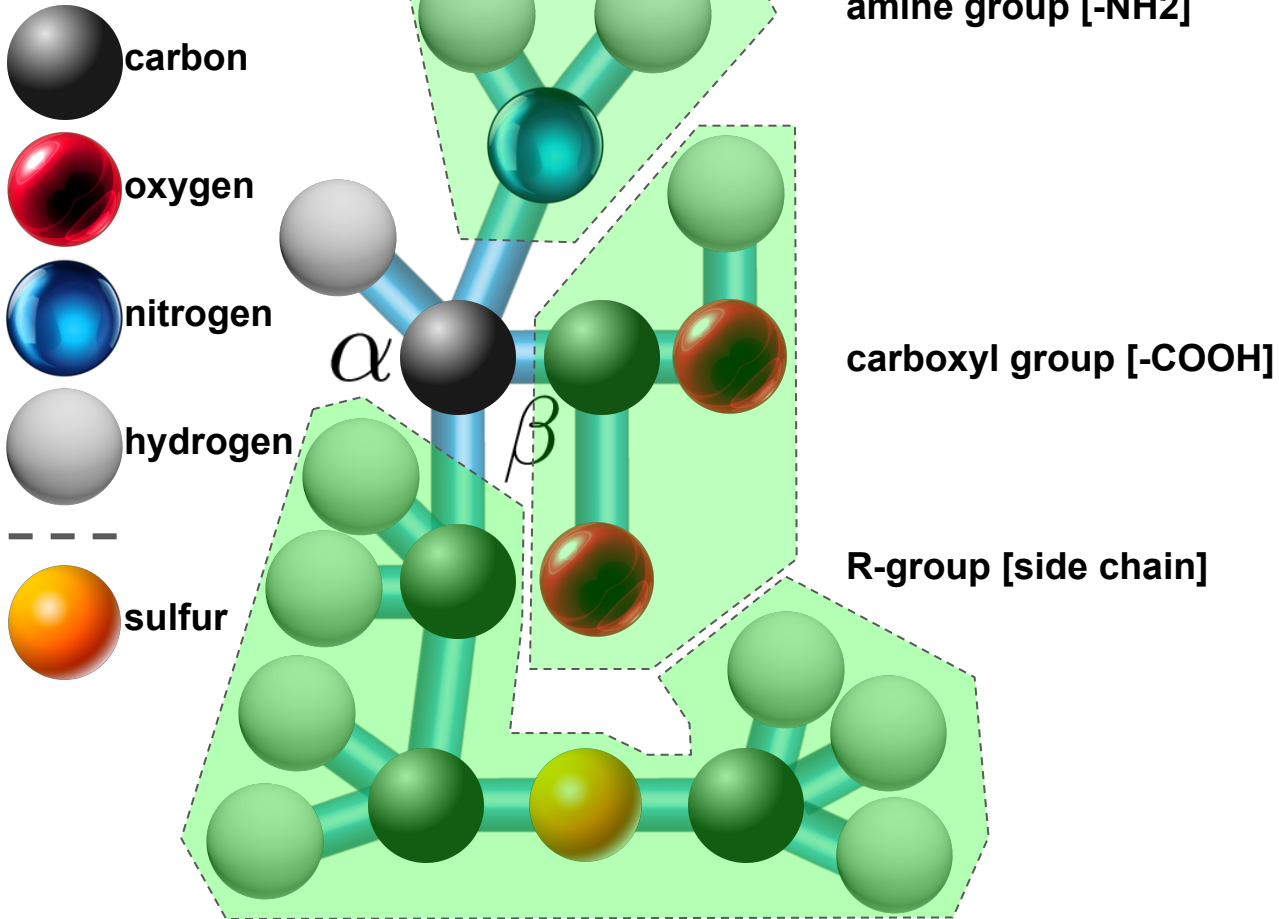
Primary Structure



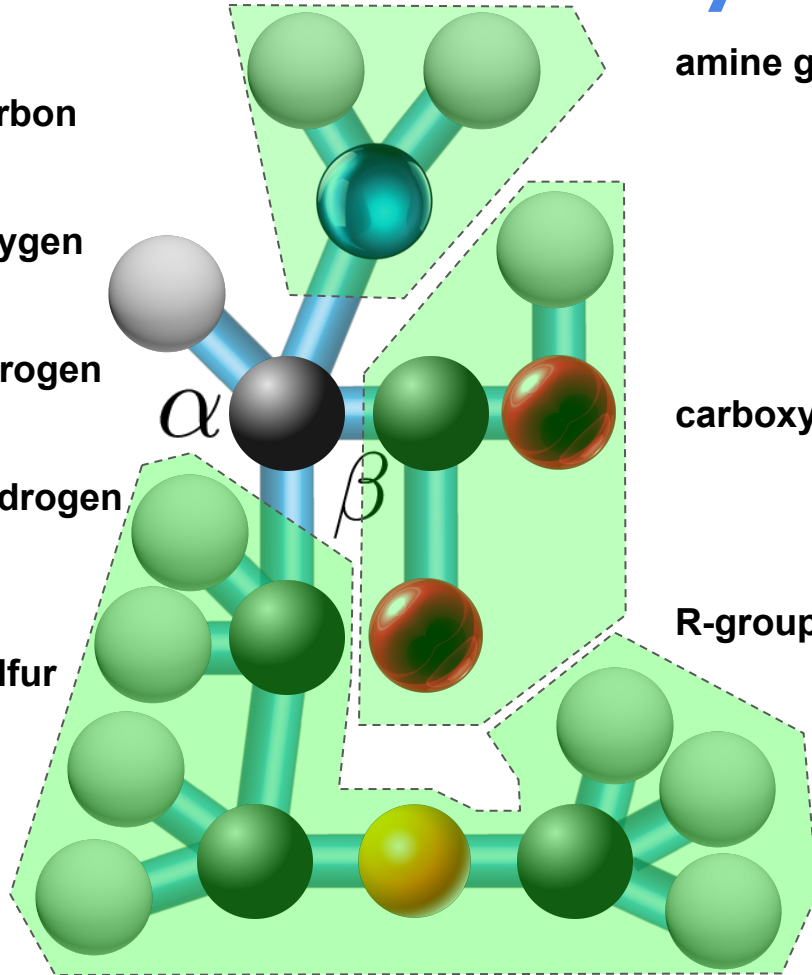
Primary Structure



Primary Structure



Primary Structure

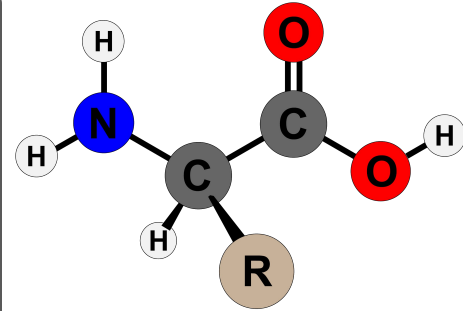


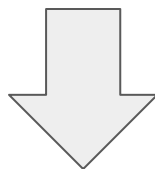
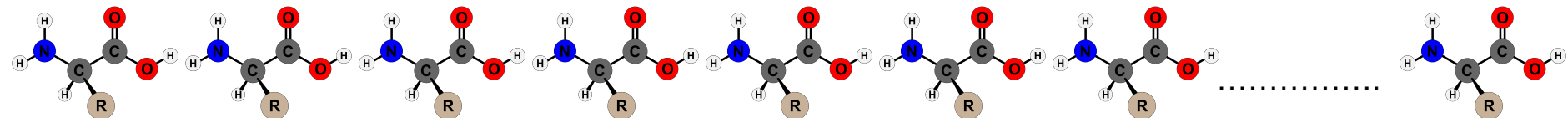
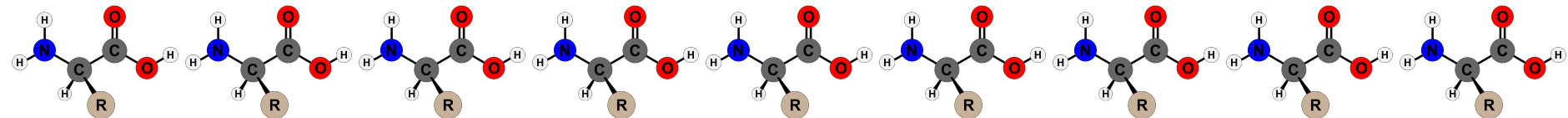
amine group [-NH₂]

carboxyl group [-COOH]

R-group [side chain]

core

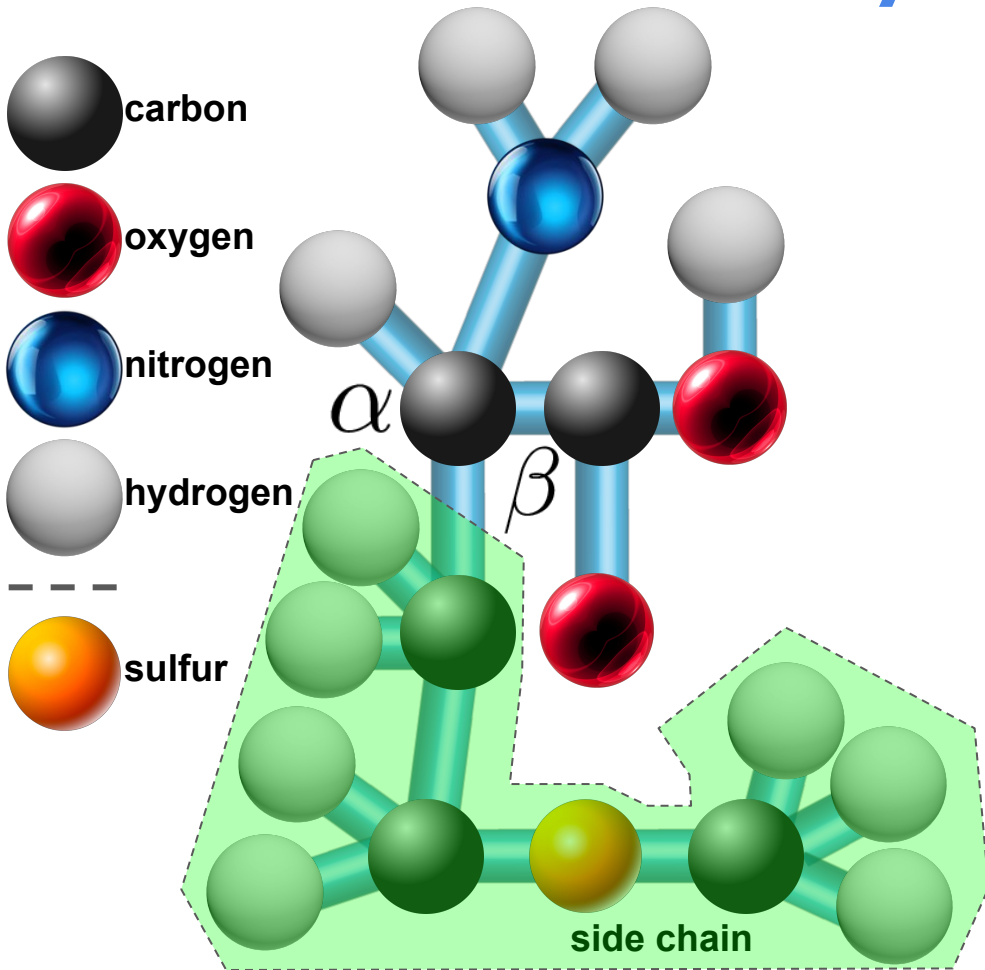










R T S D C F E L K

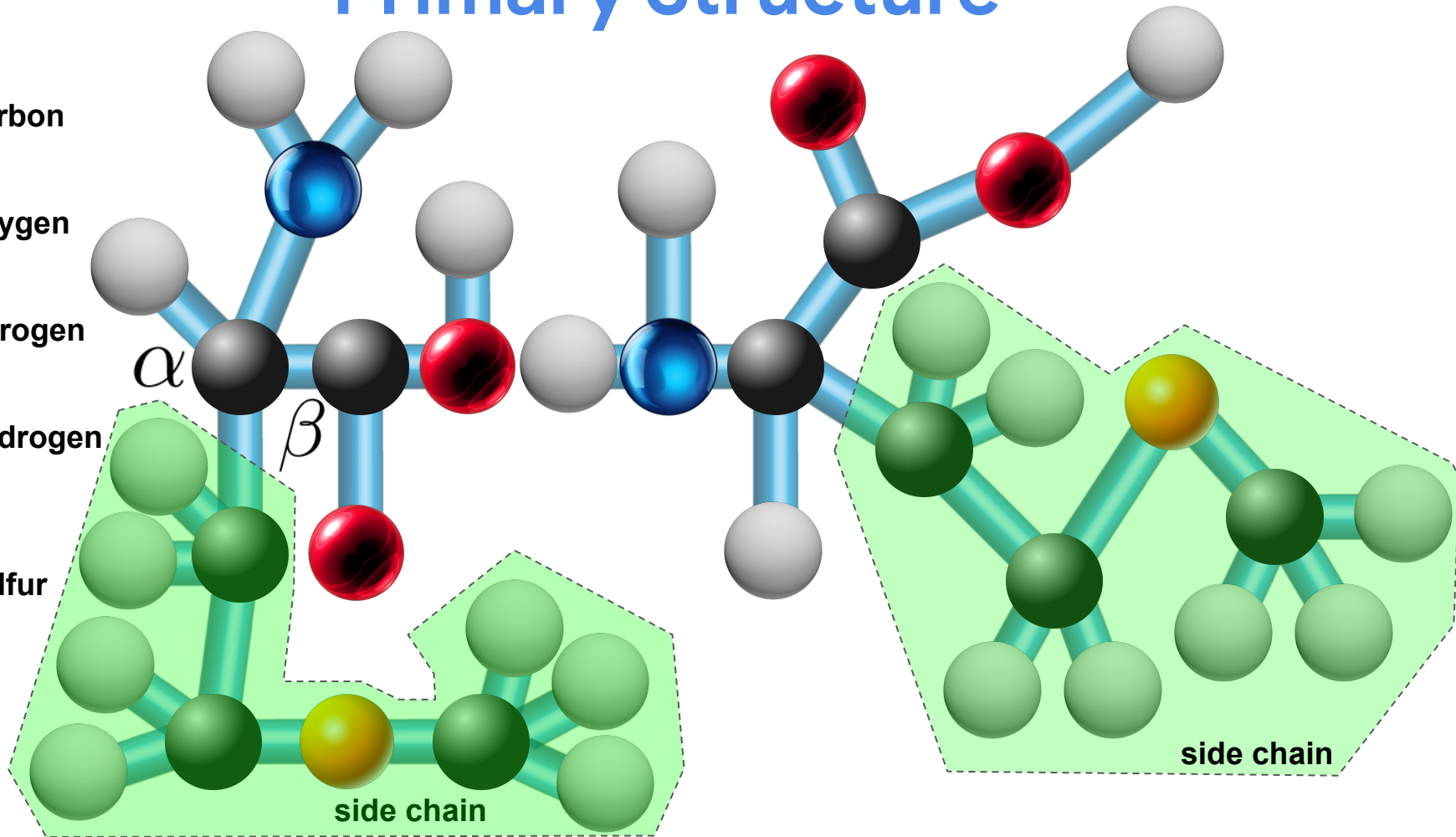
K L Y W Q R Y X Q

Primary Structure

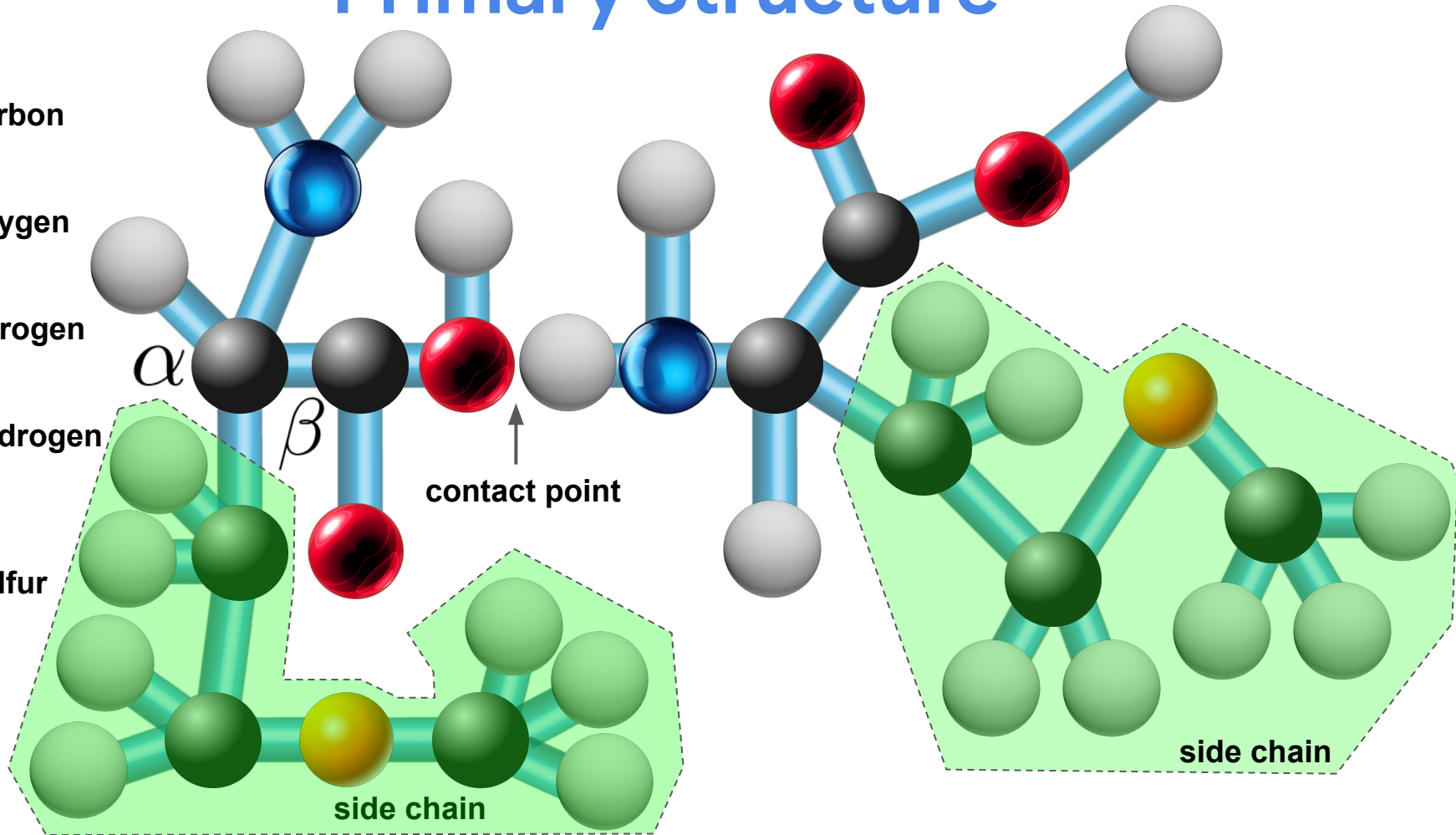
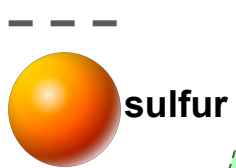


Primary Structure

-  carbon
-  oxygen
-  nitrogen
-  hydrogen
-  - - -
-  sulfur



Primary Structure



Primary Structure

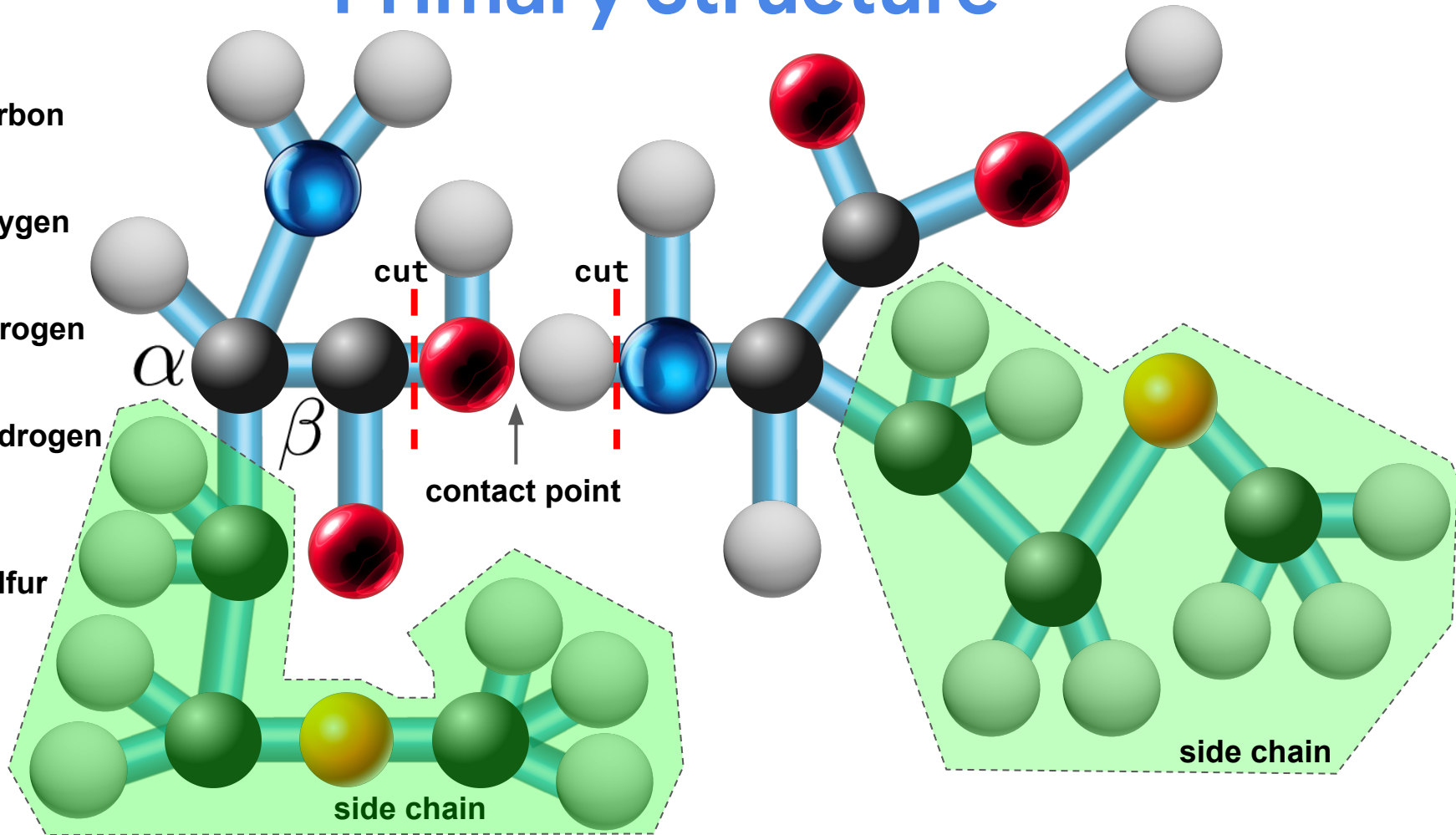
carbon

oxygen

nitrogen

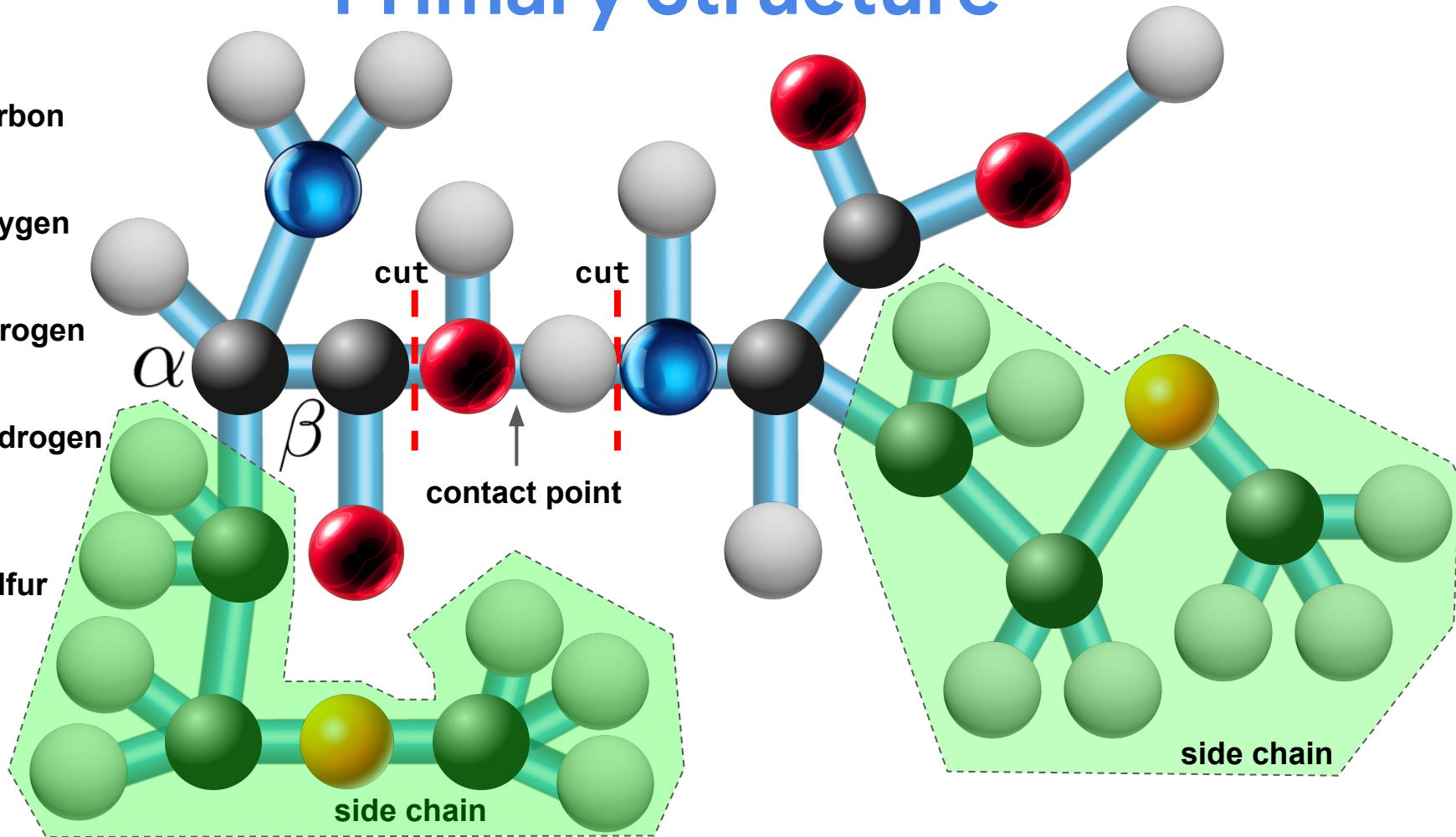
hydrogen

sulfur



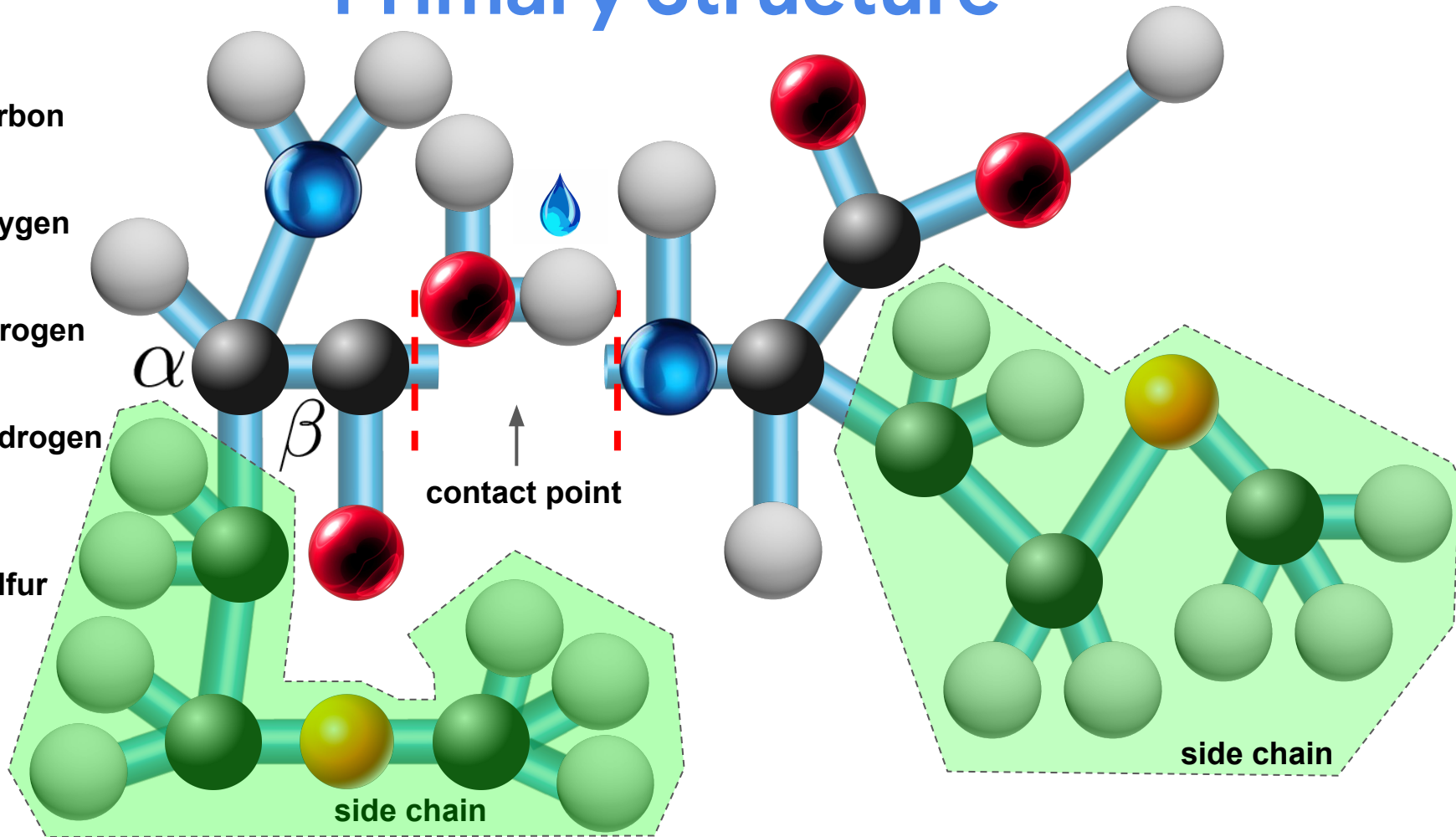
Primary Structure

- carbon
- oxygen
- nitrogen
- hydrogen
- sulfur



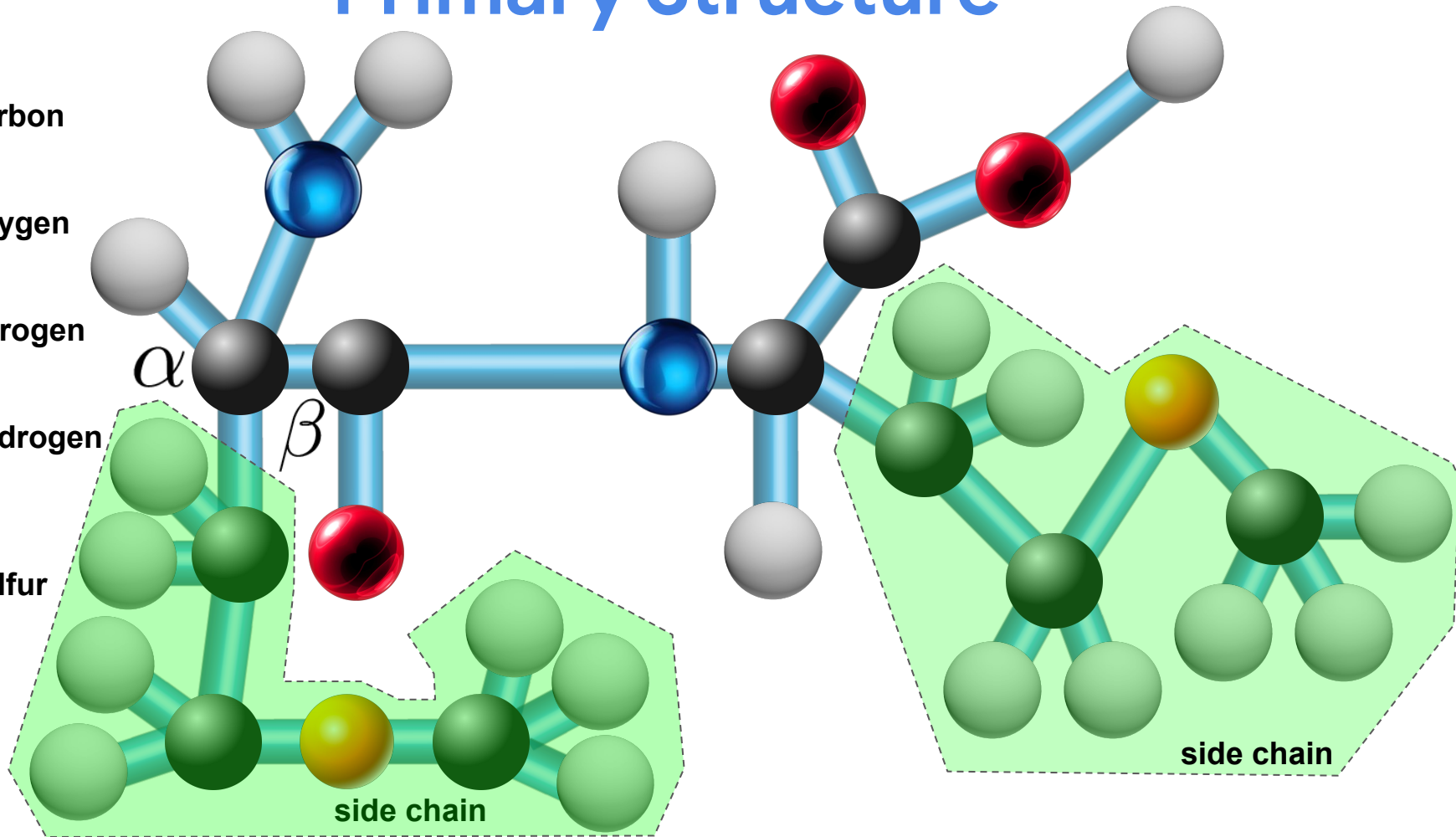
Primary Structure

- carbon
- oxygen
- nitrogen
- hydrogen
- sulfur

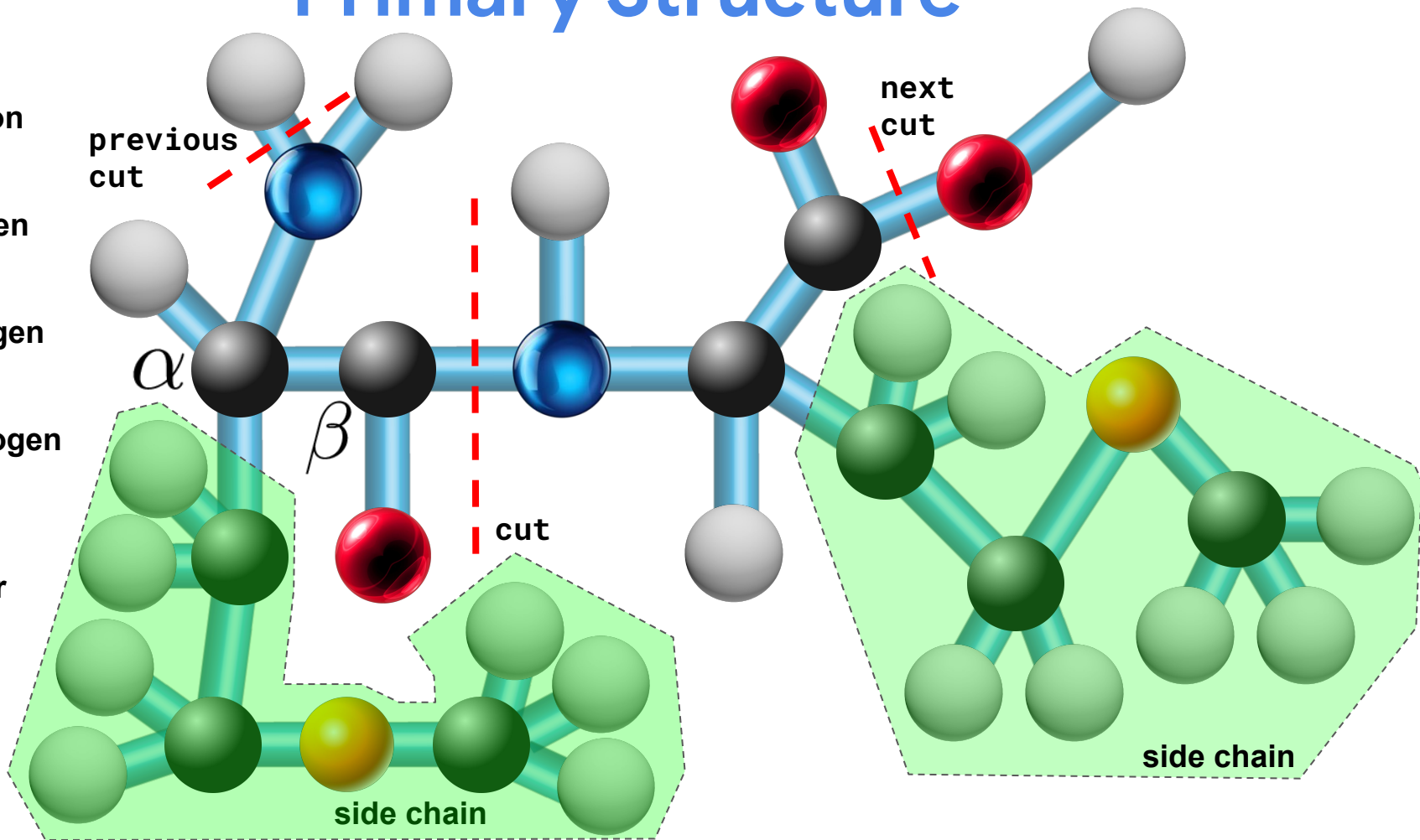
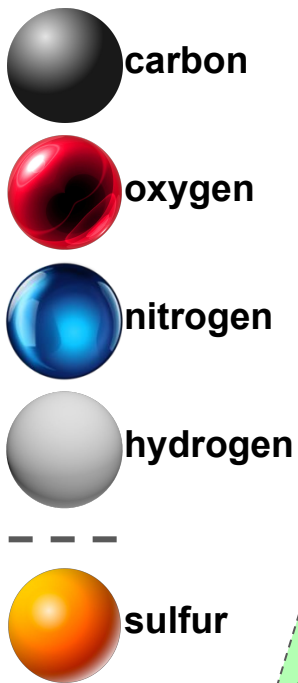


Primary Structure

- carbon
- oxygen
- nitrogen
- hydrogen
- sulfur



Primary Structure



Primary Structure

protein backbone

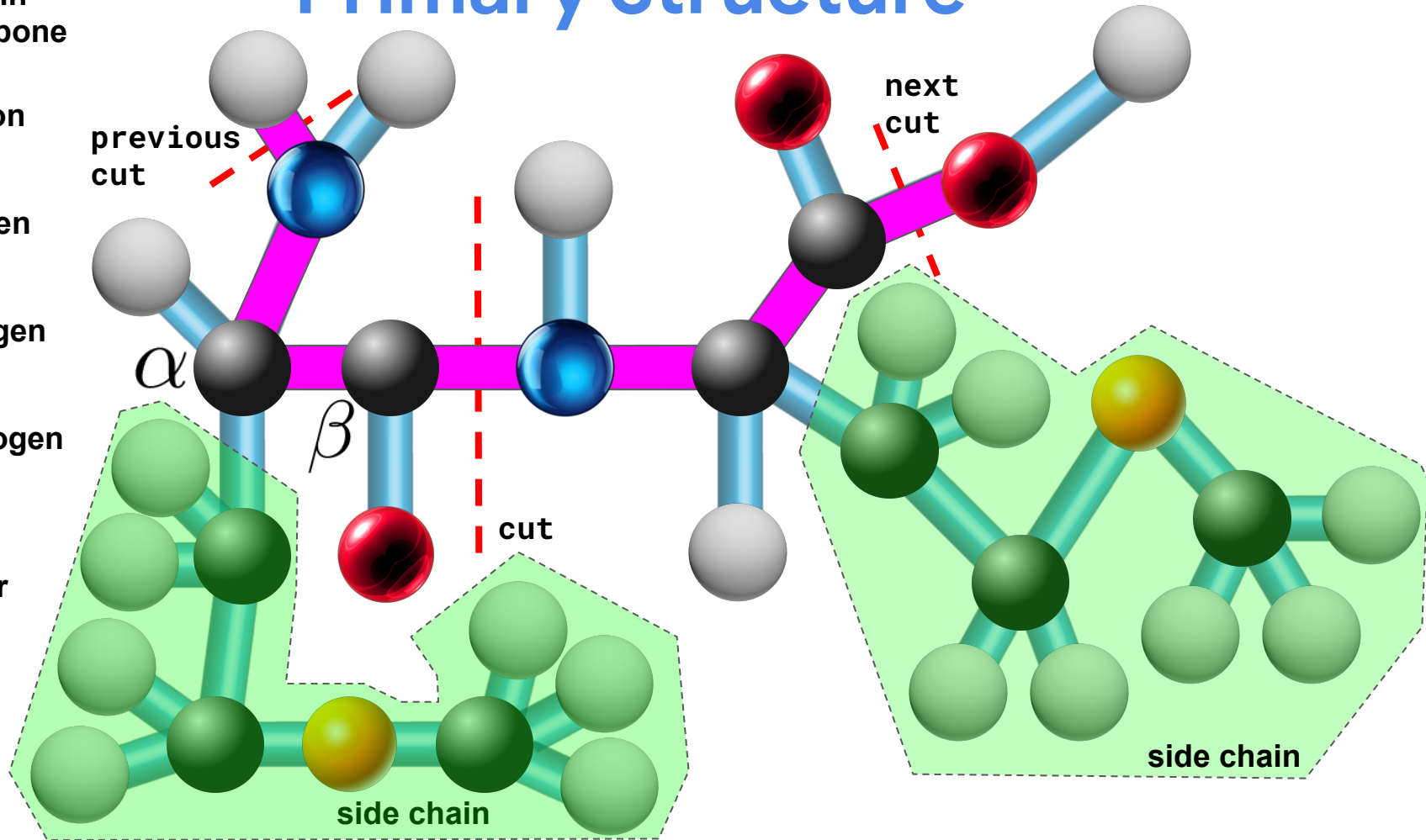
carbon

oxygen

nitrogen

hydrogen

sulfur



Tertiary Structure

protein backbone

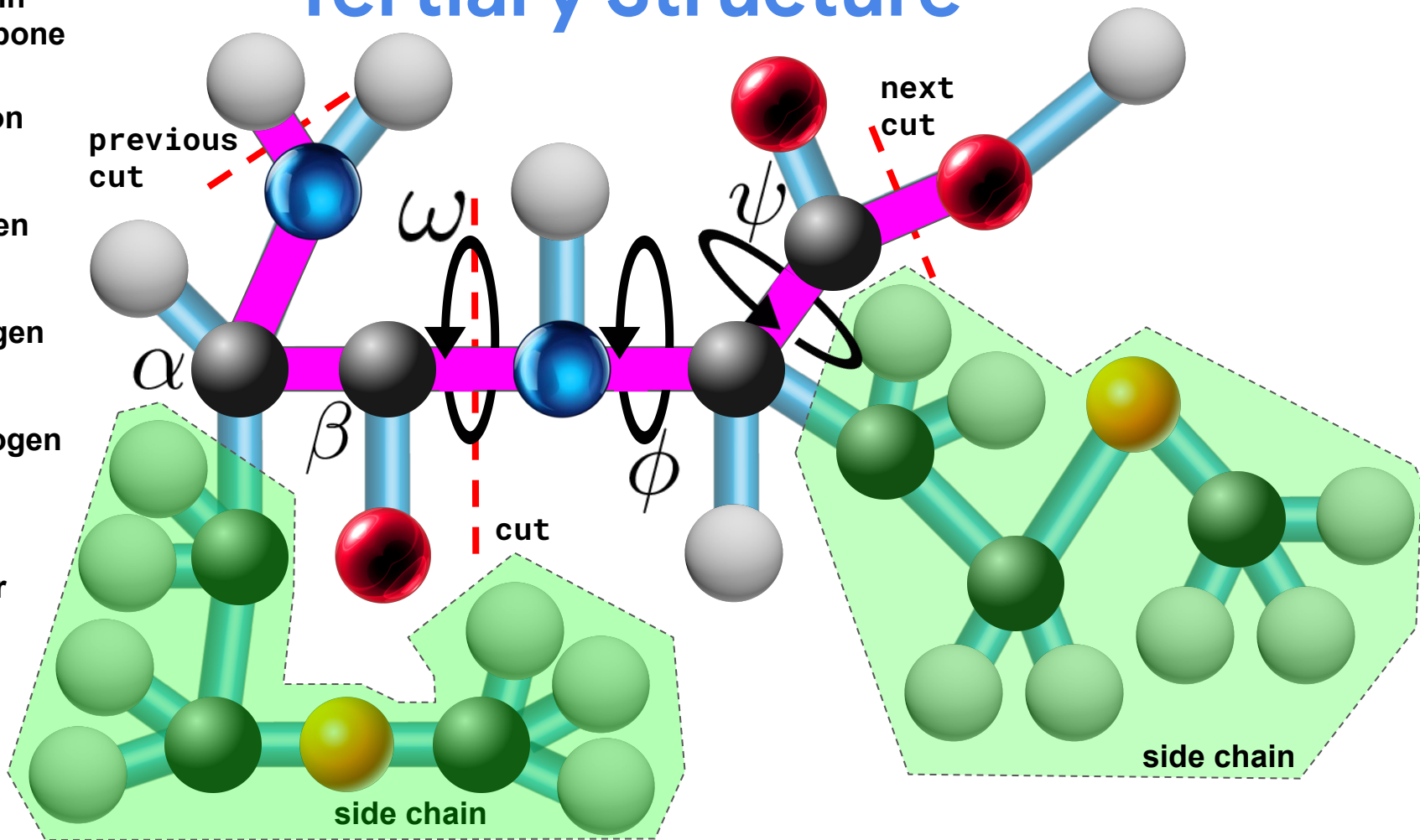
carbon

oxygen

nitrogen

hydrogen

sulfur



Tertiary Structure

protein backbone

carbon

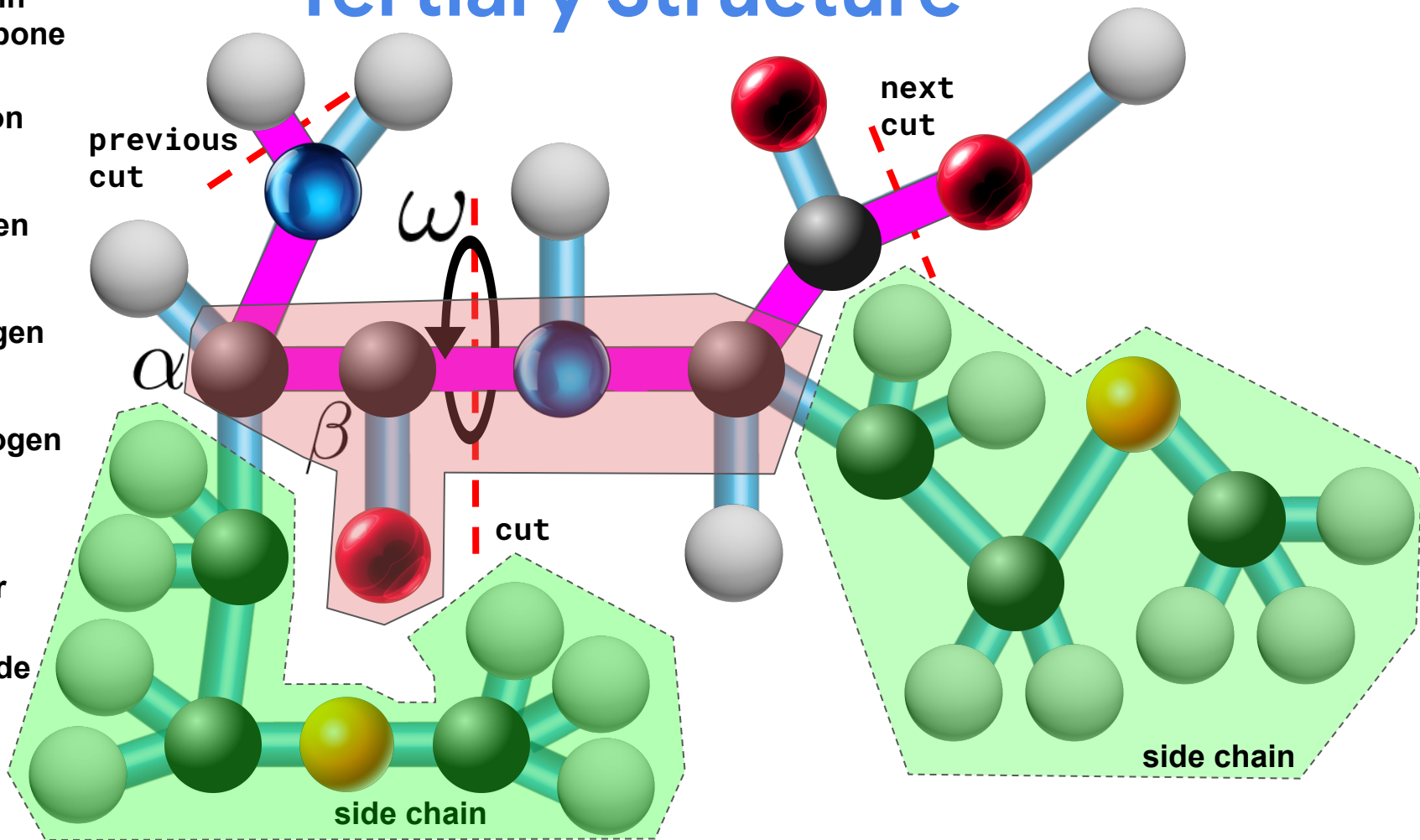
oxygen

nitrogen

hydrogen

sulfur

peptide unit



Tertiary Structure

protein backbone

carbon

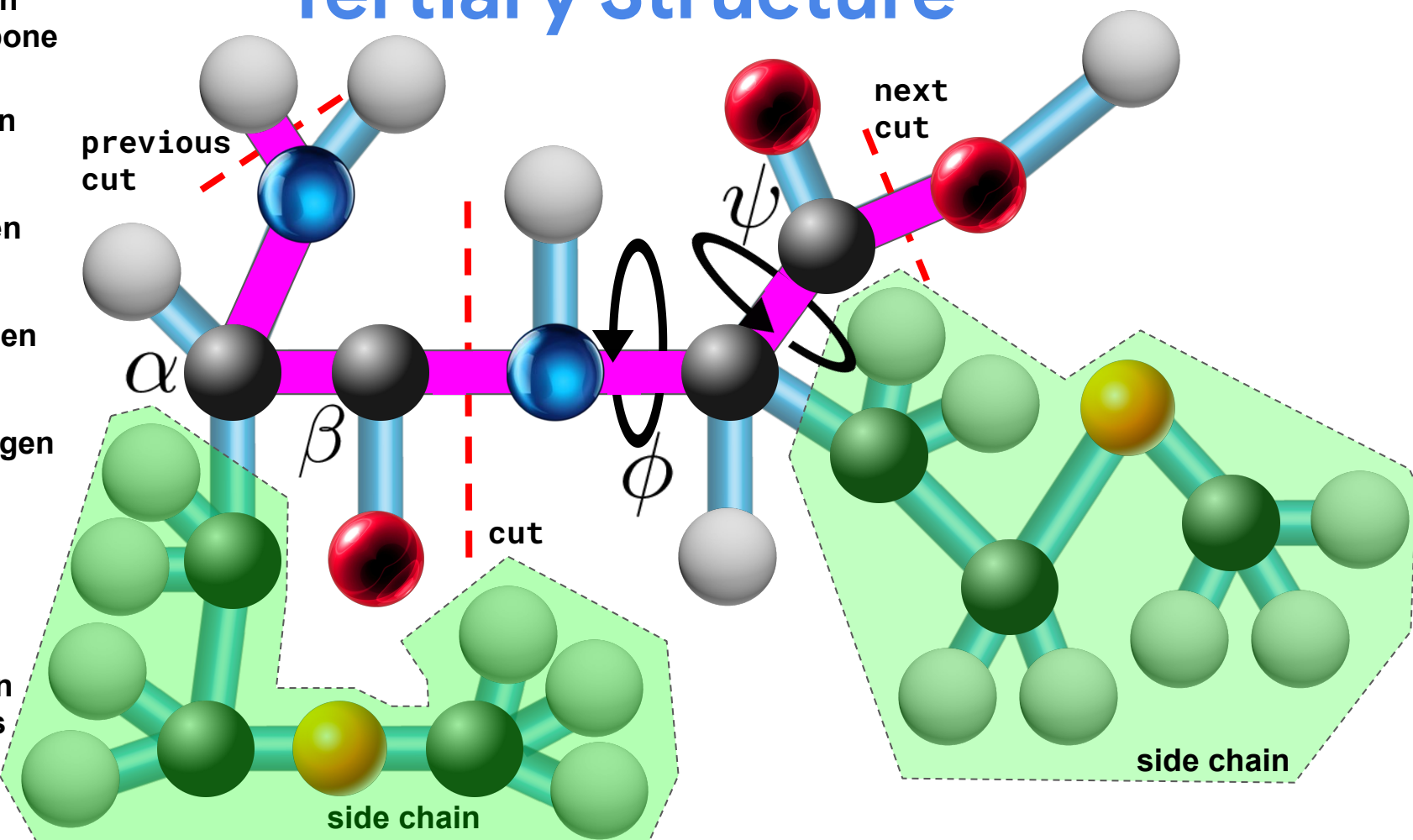
oxygen

nitrogen

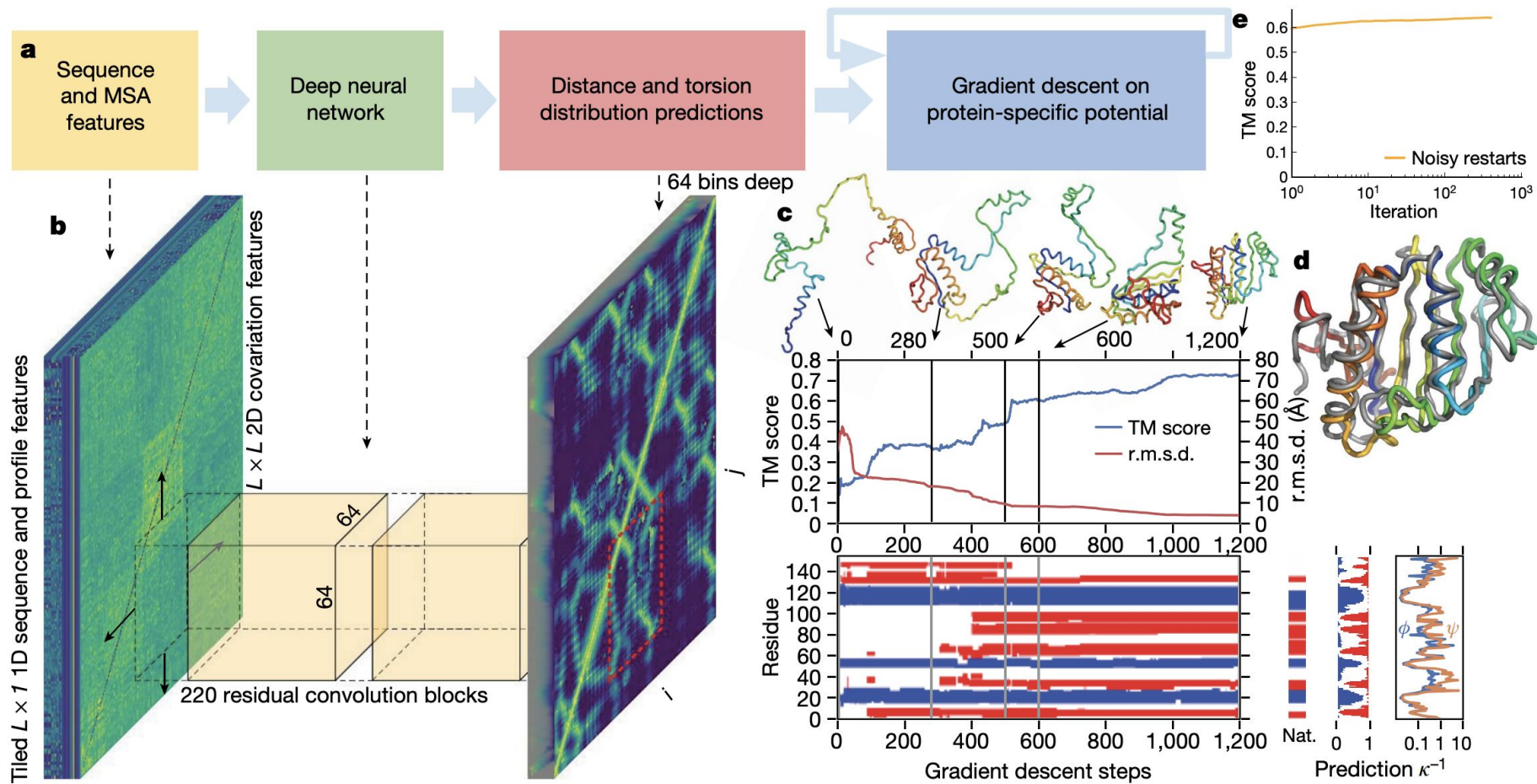
hydrogen

sulfur

torsion angles
 ψ
 ϕ



AlphaFold

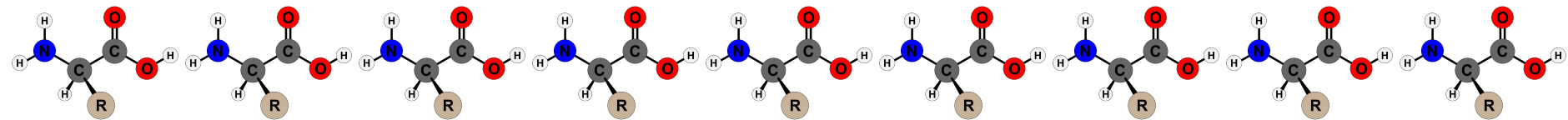


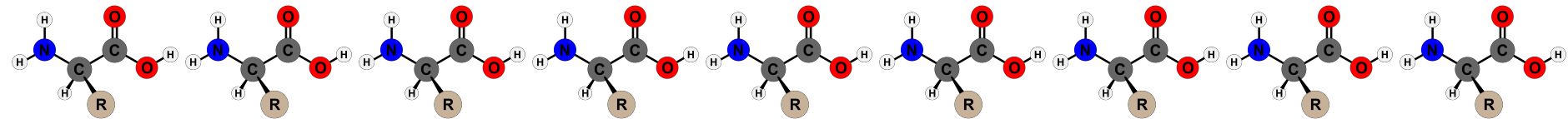


Performers for Protein Design

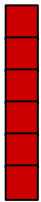
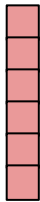


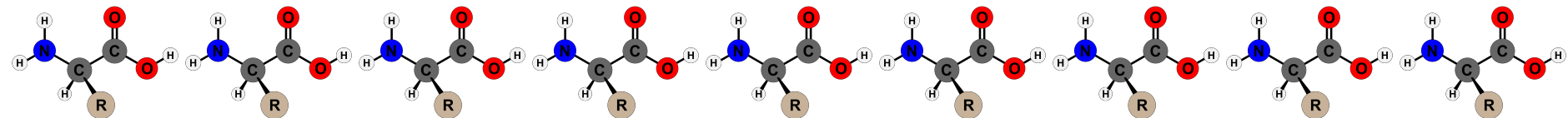
Performers for Protein Design



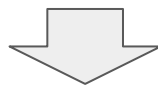
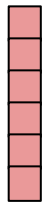


X_i

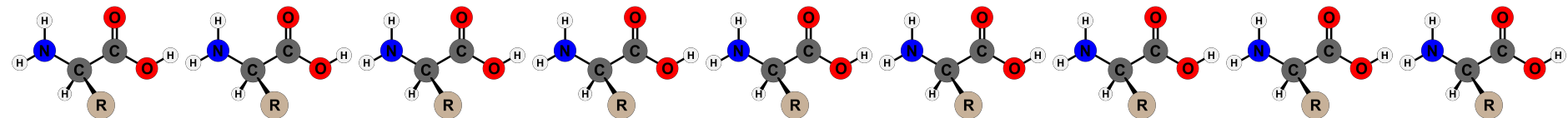




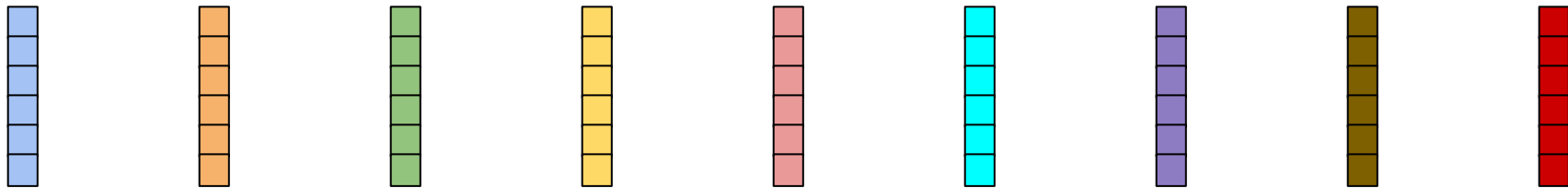
X_i



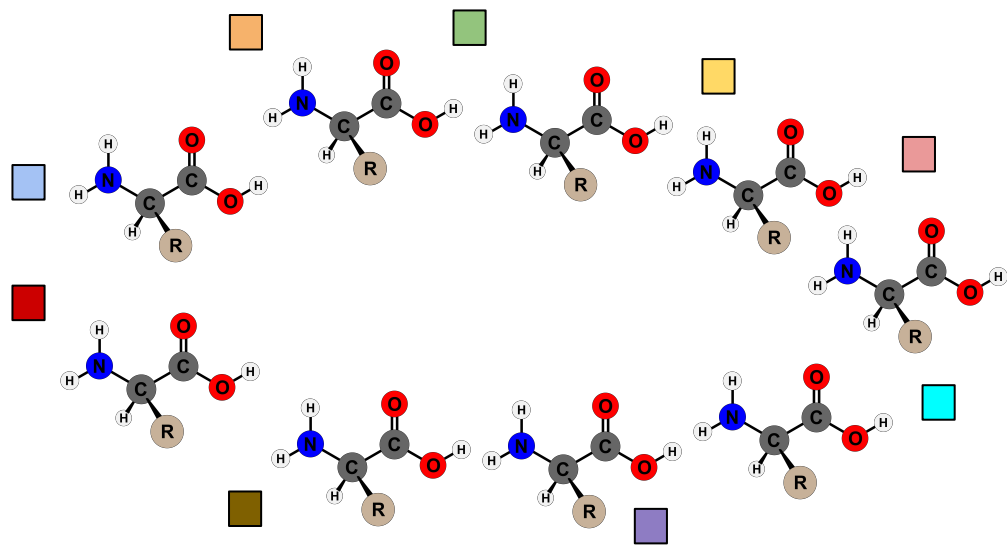
tertiary structure

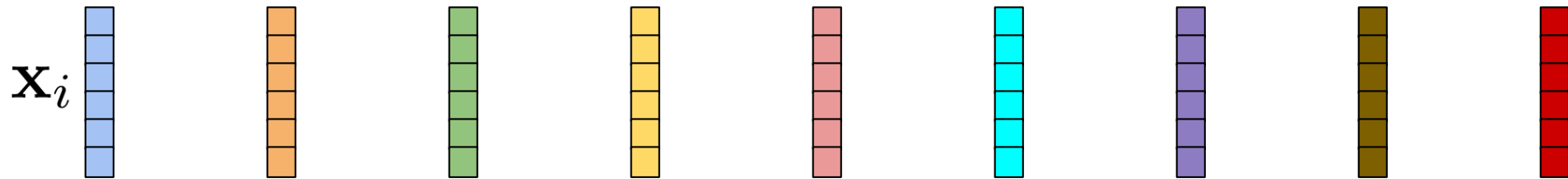
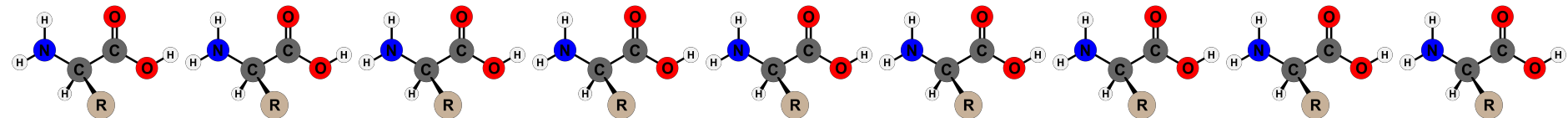


X_i

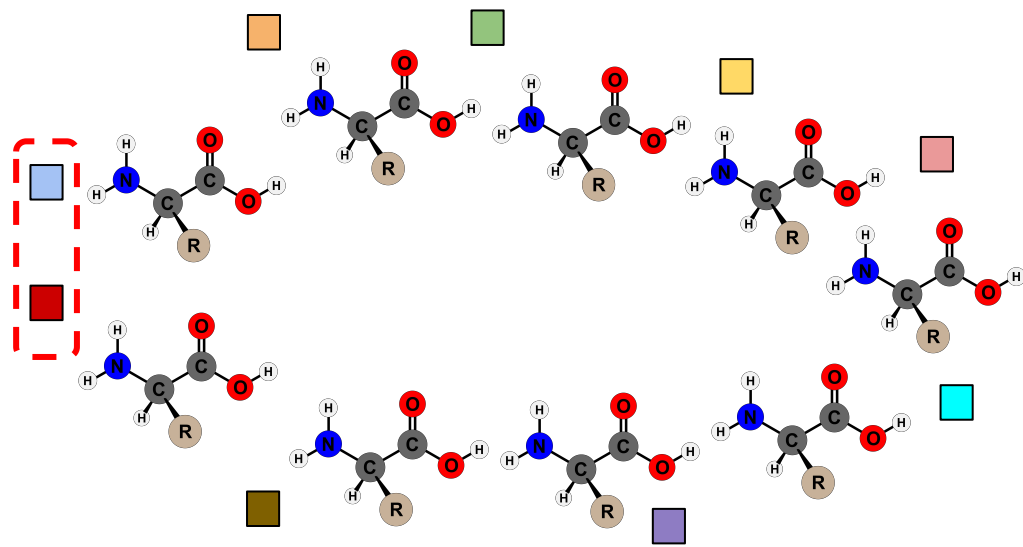


tertiary structure





tertiary structure

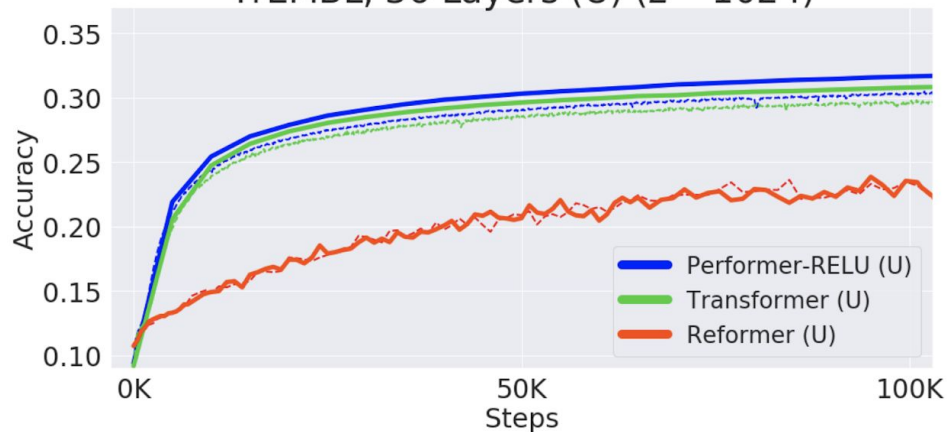


What we have already done...

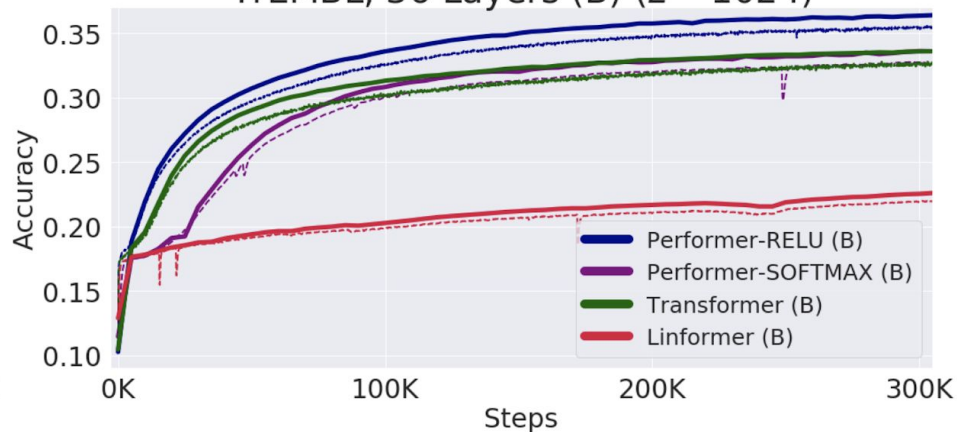


Performers on moderate-size biological sequences

TrEMBL, 36 Layers (U) ($L = 1024$)

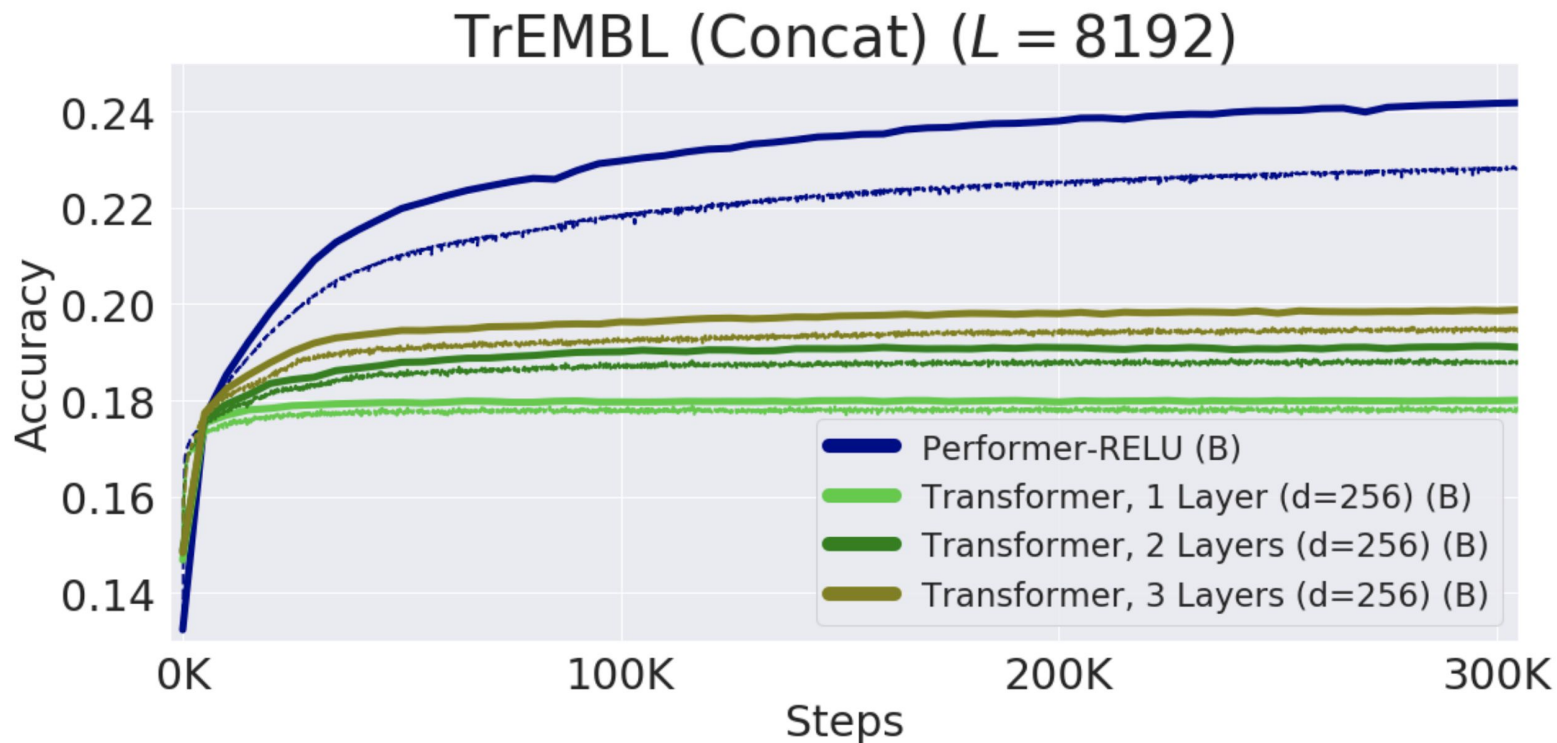


TrEMBL, 36 Layers (B) ($L = 1024$)

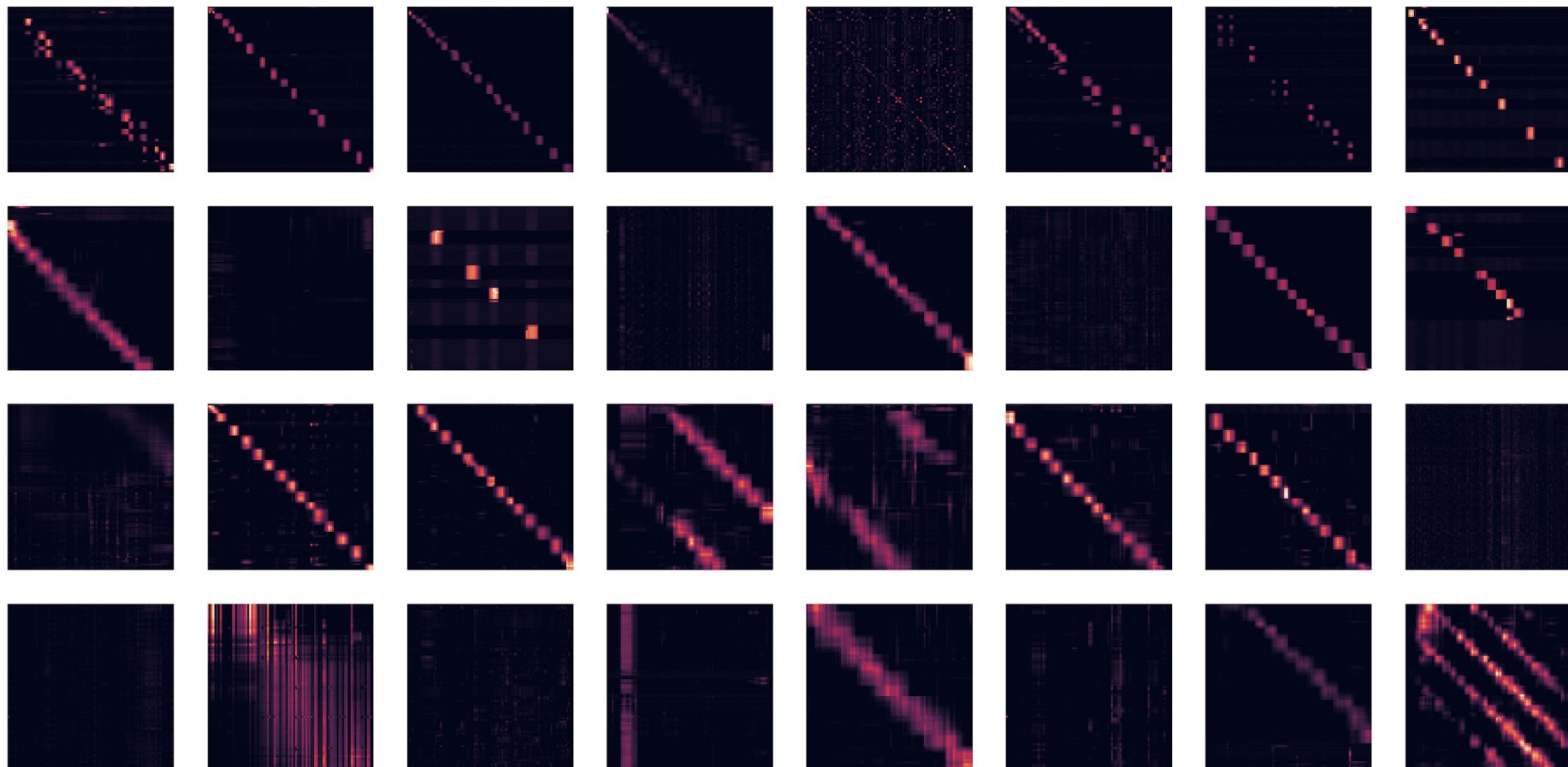


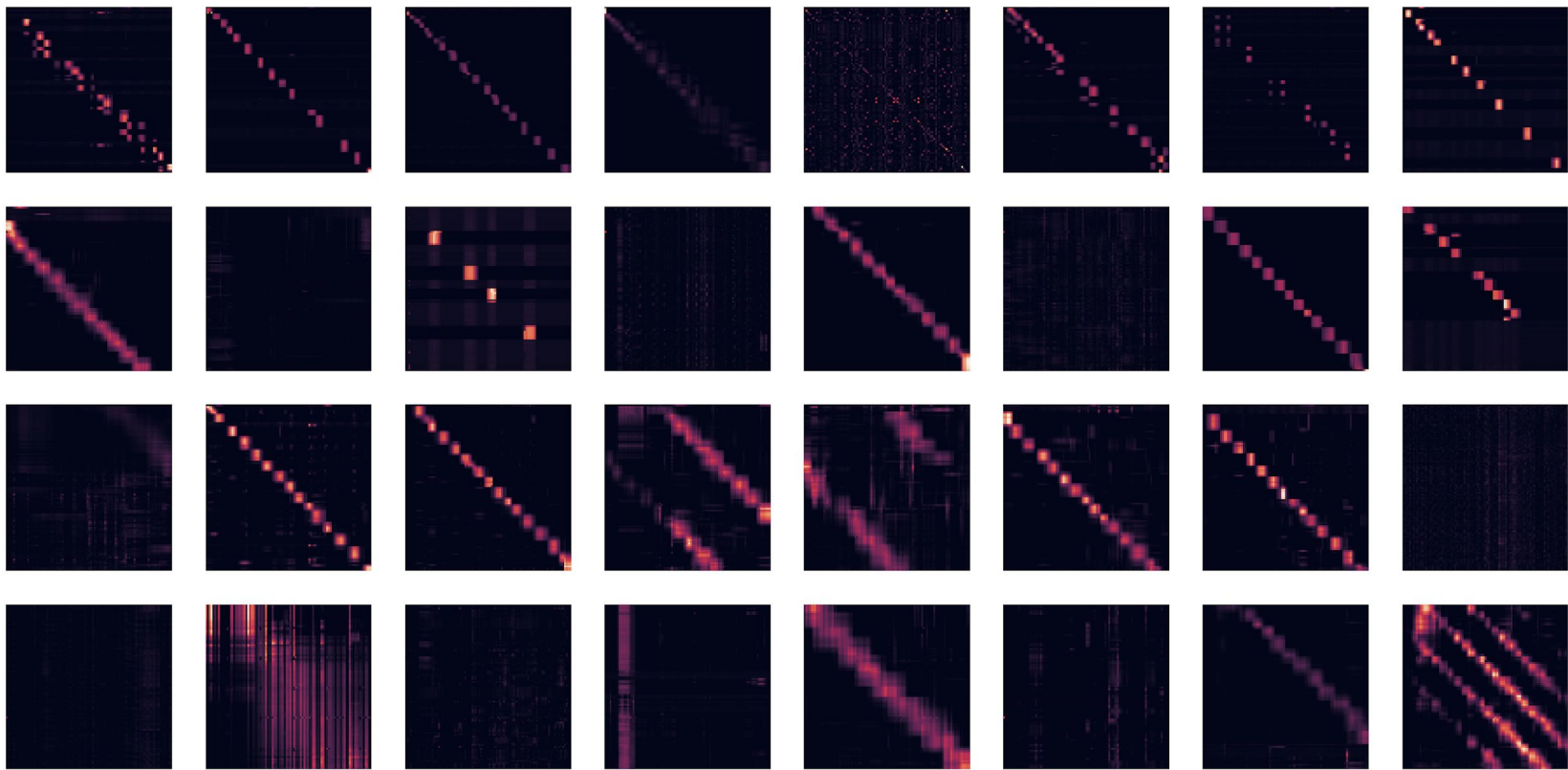
Train = Dashed, Validation = Solid, Unidirectional = (U), Bidirectional = (B). For TrEMBL, we used the exact same model parameters ($n_{heads}, n_{layers}, d_{ff}, d$) = (8, 36, 1024, 512) from (Madani et al., 2020) for all runs. For fairness, all TrEMBL experiments used 16x16 TPU-v2's. Batch sizes were maximized for each separate run given the compute constraints. Hyperparameters & extended results including dataset statistics, out of distribution evaluations, and visualizations will appear soon in the extended version of the paper.

Performers on long biological sequences: towards modeling complexes of proteins - proof of concept

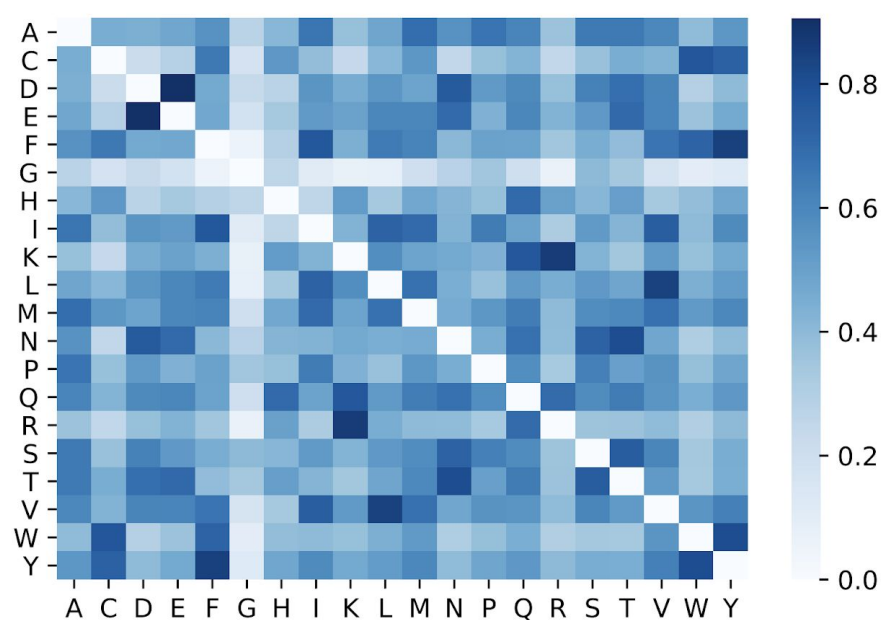
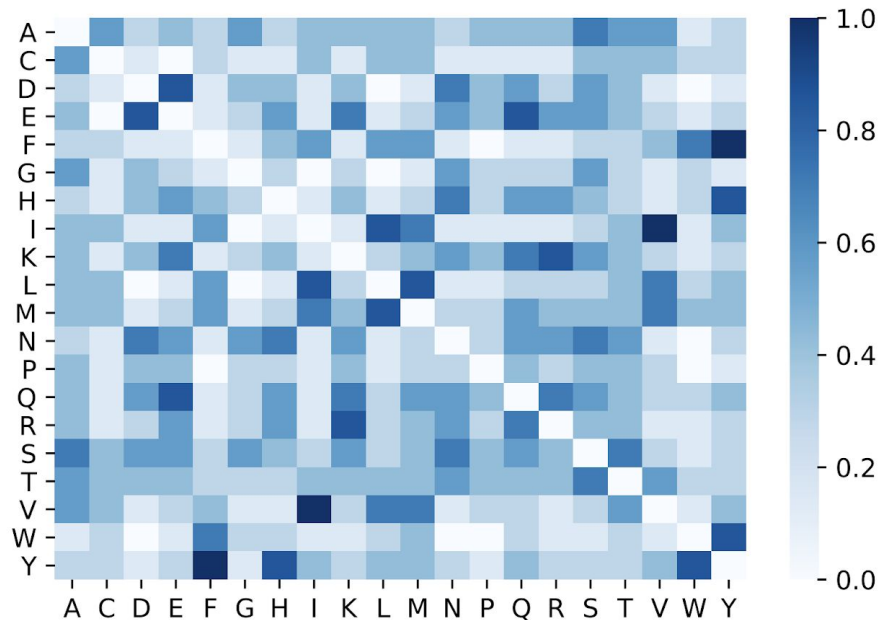


What do Performers attend to ?





We show the attention matrices for the first 4 layers and all 8 heads (each row is a layer, each column is head index, each cell contains the attention matrix across the entire BPT1_BOVIN protein sequence). Note that many heads show a diagonal pattern, where each node attends to its neighbors, and some heads show a vertical pattern, where each head attends to the same fixed positions.



Amino acid similarity matrix estimated from attention matrices aggregated across a small subset of sequences, as described in Vig et al. (Vig et al., 2020). The sub-figures correspond respectively to: (1) the normalized BLOSUM matrix, (2) the amino acid similarity estimated via a trained Performer model. Note that the Performer recognizes highly similar amino acid pairs such as (D, E) and (F, Y).

Performers performing on the Long Range Arena

Long Range Arena Paper



Model	ListOps	Text	Retrieval	Image	Pathfinder	Path-X	Avg
Transformer	36.37	64.27	57.46	42.44	71.40	FAIL	<u>54.39</u>
Local Attention	15.82	52.98	53.39	41.46	66.63	FAIL	46.06
Sparse Trans.	17.07	63.58	59.59	44.24	71.71	FAIL	51.24
Longformer	35.63	62.85	56.89	42.22	69.71	FAIL	53.46
Linformer	35.70	53.94	52.27	38.56	<u>76.34</u>	FAIL	51.36
Reformer	37.27	56.10	53.40	38.07	68.50	FAIL	50.67
Sinkhorn Trans.	33.67	61.20	53.83	41.23	67.45	FAIL	51.39
Synthesizer	<u>36.99</u>	61.68	54.67	41.61	69.45	FAIL	52.88
BigBird	36.05	64.02	<u>59.29</u>	40.83	74.87	FAIL	55.01
Linear Trans.	16.13	65.90	53.09	42.34	75.30	FAIL	50.55
Performer	18.01	<u>65.40</u>	53.82	<u>42.77</u>	77.05	FAIL	<u>51.41</u>
Task Avg (Std)	29 (9.7)	61 (4.6)	55 (2.6)	41 (1.8)	72 (3.7)	FAIL	52 (2.4)

Table 1: Experimental results on Long-Range Arena benchmark. Best model is in boldface and second best is underlined. All models do not learn anything on Path-X task, contrary to the Pathfinder task and this is denoted by FAIL. This shows that increasing the sequence length can cause serious difficulties for model training. We leave Path-X on this benchmark for future challengers but do not include it on the Average score as it has no impact on relative performance.

Model	Steps per second				Peak Memory Usage (GB)			
	1K	2K	3K	4K	1K	2K	3K	4K
Transformer	8.1	4.9	2.3	1.4	0.85	2.65	5.51	9.48
Local Attention	9.2 (1.1x)	8.4 (1.7x)	7.4 (3.2x)	7.4 (5.3x)	0.42	0.76	1.06	1.37
Linformer	<u>9.3</u> (1.2x)	9.1 (1.9x)	8.5 (3.7x)	7.7 (5.5x)	0.37	0.55	0.99	0.99
Reformer	4.4 (0.5x)	2.2 (0.4x)	1.5 (0.7x)	1.1 (0.8x)	0.48	0.99	1.53	2.28
Sinkhorn Trans	9.1 (1.1x)	7.9 (1.6x)	6.6 (2.9x)	5.3 (3.8x)	0.47	0.83	1.13	1.48
Synthesizer	8.7 (1.1x)	5.7 (1.2x)	6.6 (2.9x)	1.9 (1.4x)	0.65	1.98	4.09	6.99
BigBird	7.4 (0.9x)	3.9 (0.8x)	2.7 (1.2x)	1.5 (1.1x)	0.77	1.49	2.18	2.88
Linear Trans.	9.1 (1.1x)	9.3 (1.9x)	<u>8.6</u> (3.7x)	<u>7.8</u> (5.6x)	0.37	<u>0.57</u>	0.80	<u>1.03</u>
Performer	9.5 (1.2x)	9.4 (1.9x)	8.7 (3.8x)	8.0 (5.7x)	0.37	0.59	<u>0.82</u>	1.06

Table 2: Benchmark results of all Xformer models with a consistent batch size of 32 across all models. We report relative speed increase/decrease in comparison with the vanilla Transformer in brackets besides the steps per second. Memory usage refers to per device memory usage across each TPU device. Benchmarks are run on 4x4 TPU V3 Chips.

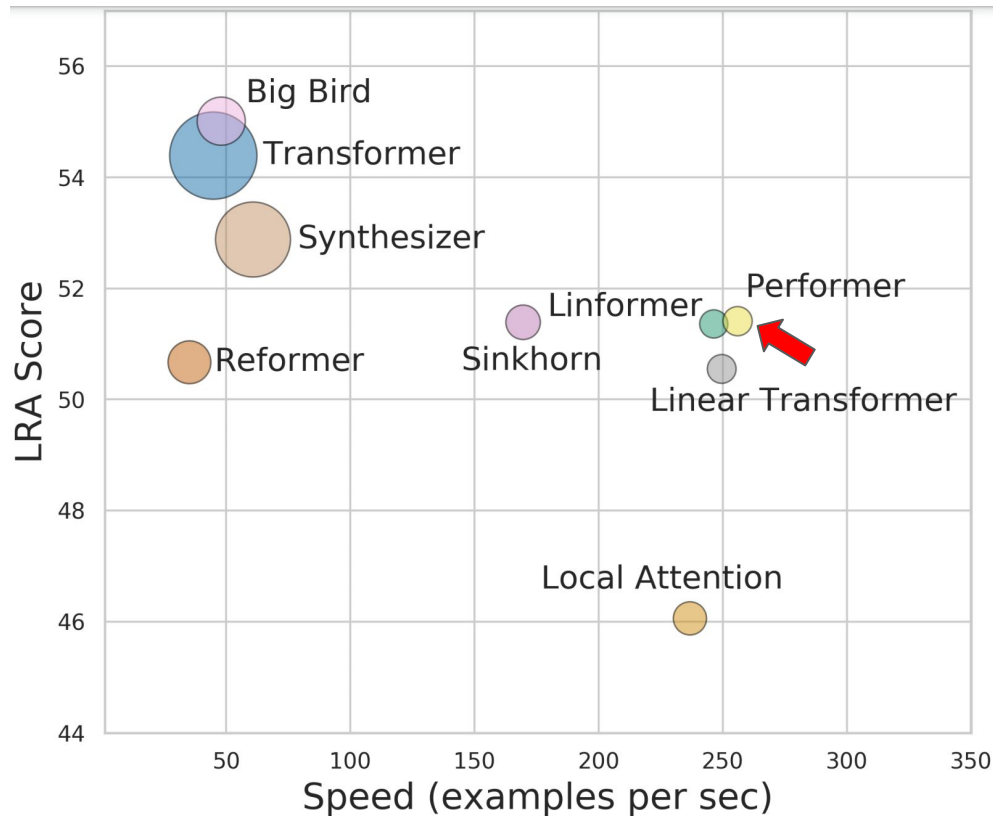


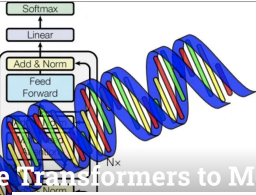
Figure 3: Performance (y axis), speed (x axis), and memory footprint (size of the circles) of different models.

Performers externally

Synced

AI TECHNOLOGY & INDUSTRY REVIEW

GLOBAL NEWS ▾ INDUSTRY ▾ COMPANY ▾ RESEARCH ▾ COMMUNITY ▾ NEWSLETTER CONTACT US ▾



Applying Linearly Scalable Transformers to Model Longer Protein Sequences

Researchers proposed a novel architecture for modeling longer protein sequences.

Synced

AI TECHNOLOGY & INDUSTRY REVIEW

GLOBAL NEWS ▾ INDUSTRY ▾ COMPANY ▾ RESEARCH ▾ COMMUNITY ▾ NEWSLETTER CONTACT US ▾



Google, Cambridge, DeepMind & Alan Turing Institute's 'Performer' Transformer Slashes Compute Costs

A team from Google, Cambridge, DeepMind, and the Alan Turing Institute has developed a new transformer architecture called Performer, which significantly reduces the compute costs of training large-scale models.



Accelerate your Machine Learning Journey with AWS
See How »

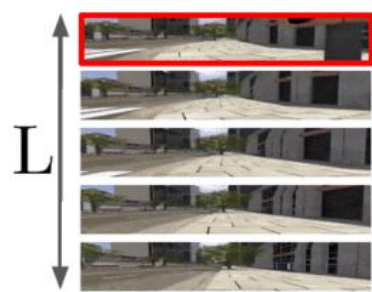
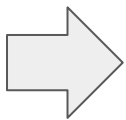
Google's Performer AI architecture could advance protein analysis and cut compute costs

Kyle Wiggins @KyleLWiggins June 4, 2022 8:10 AM AI

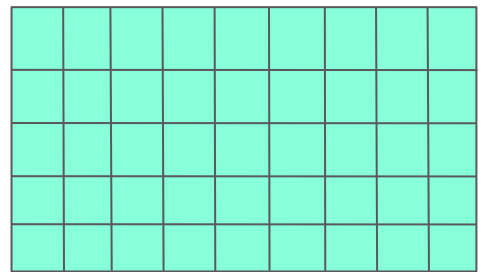
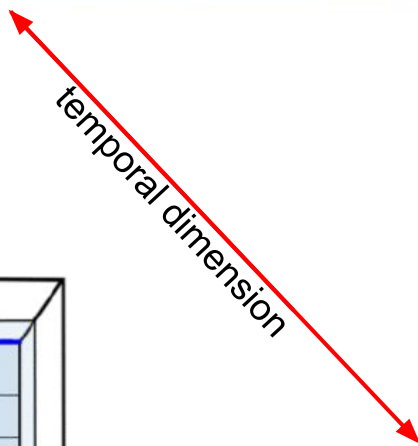
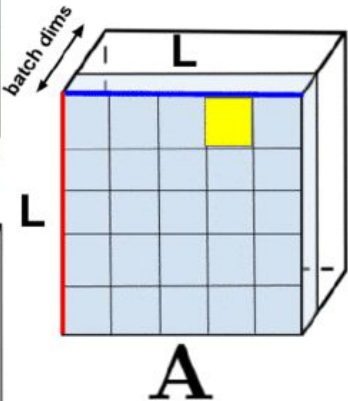
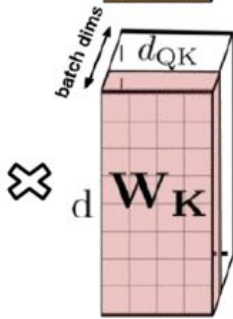
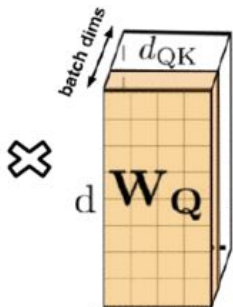
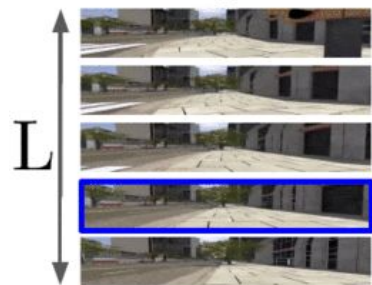
f t in



Image credit: Photo: Michael J. Gorman / Shutterstock.com

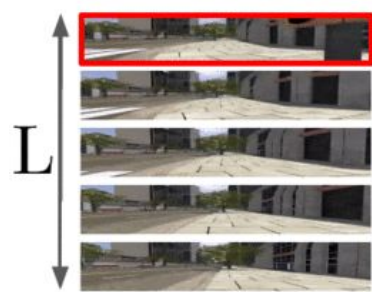
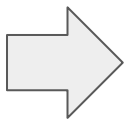


d

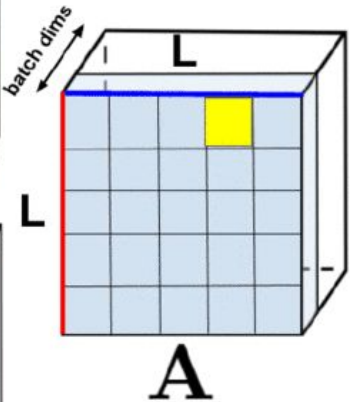
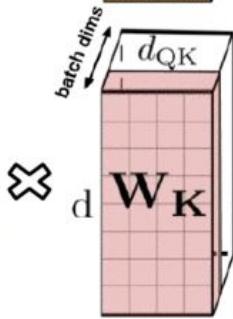
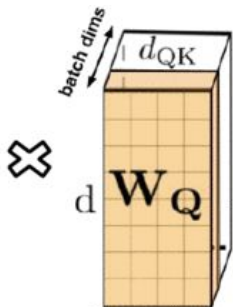
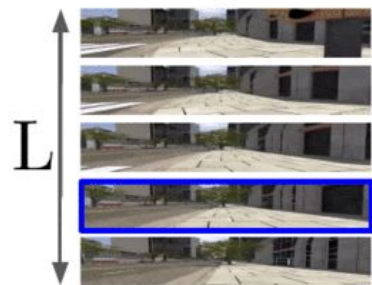


spatial dimension

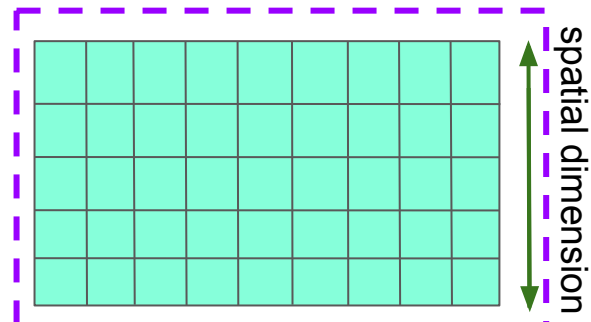
spatial dimension



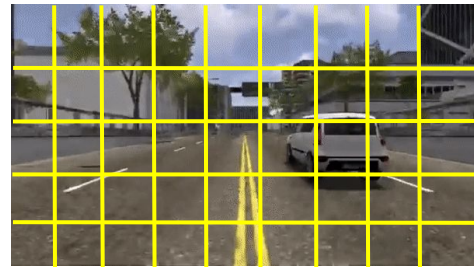
d

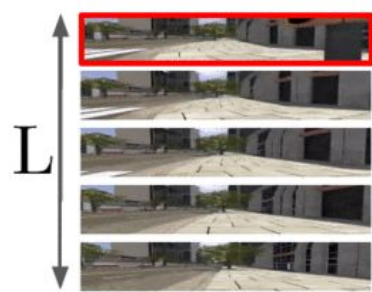
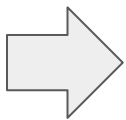


temporal dimension

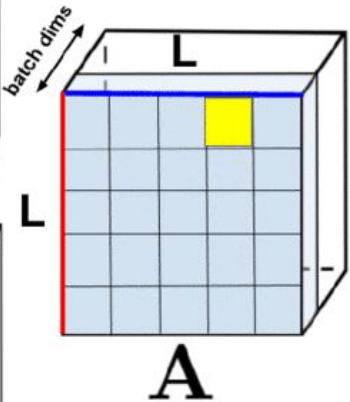
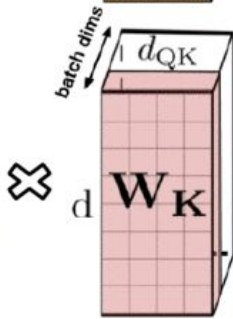
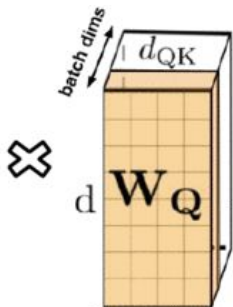
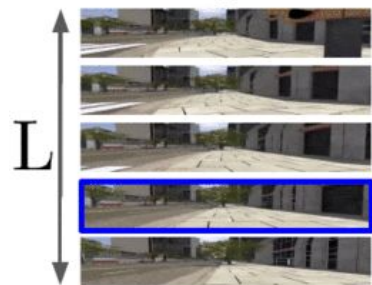


spatial dimension

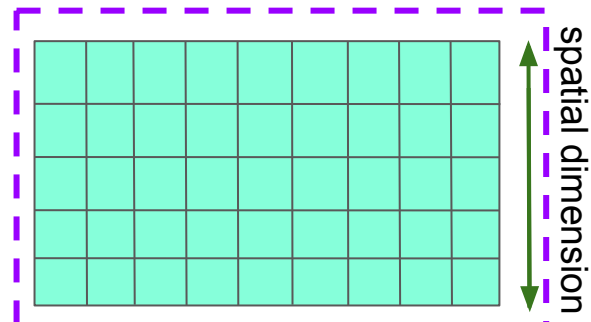




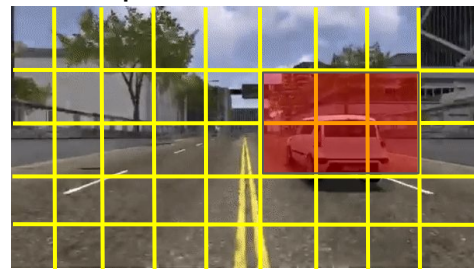
d

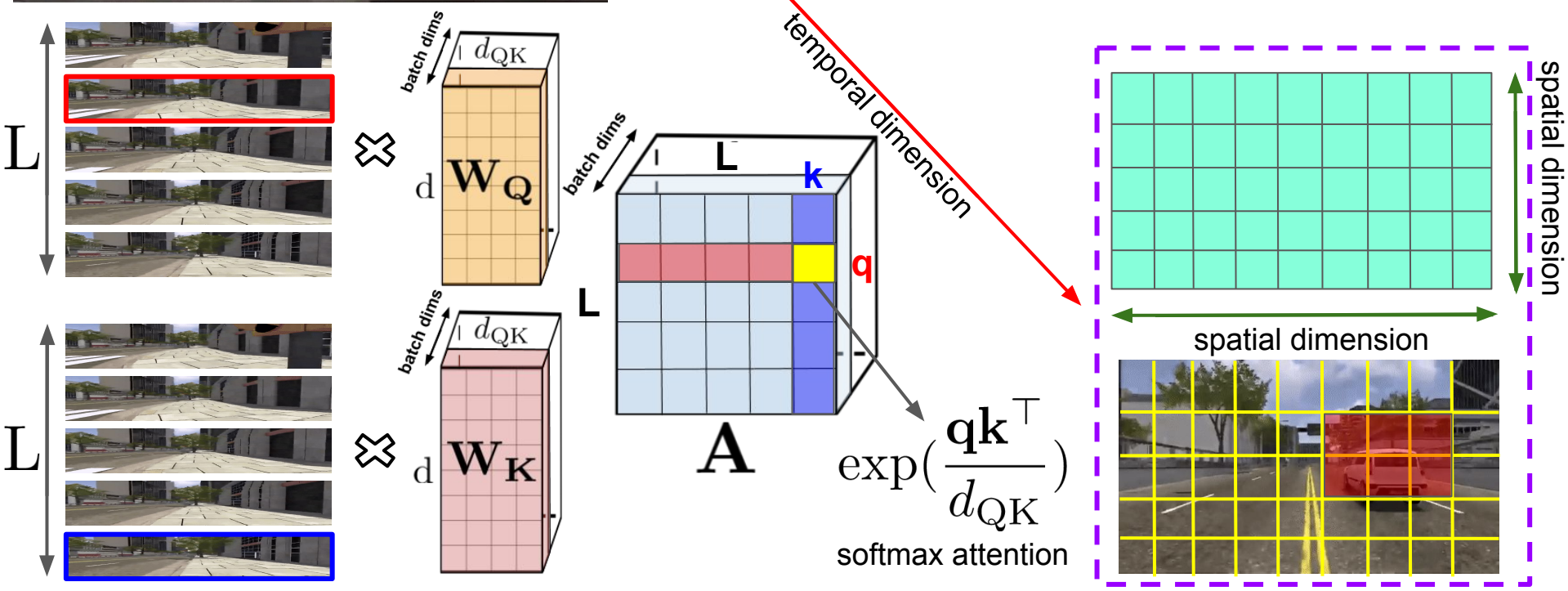
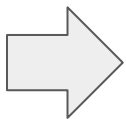


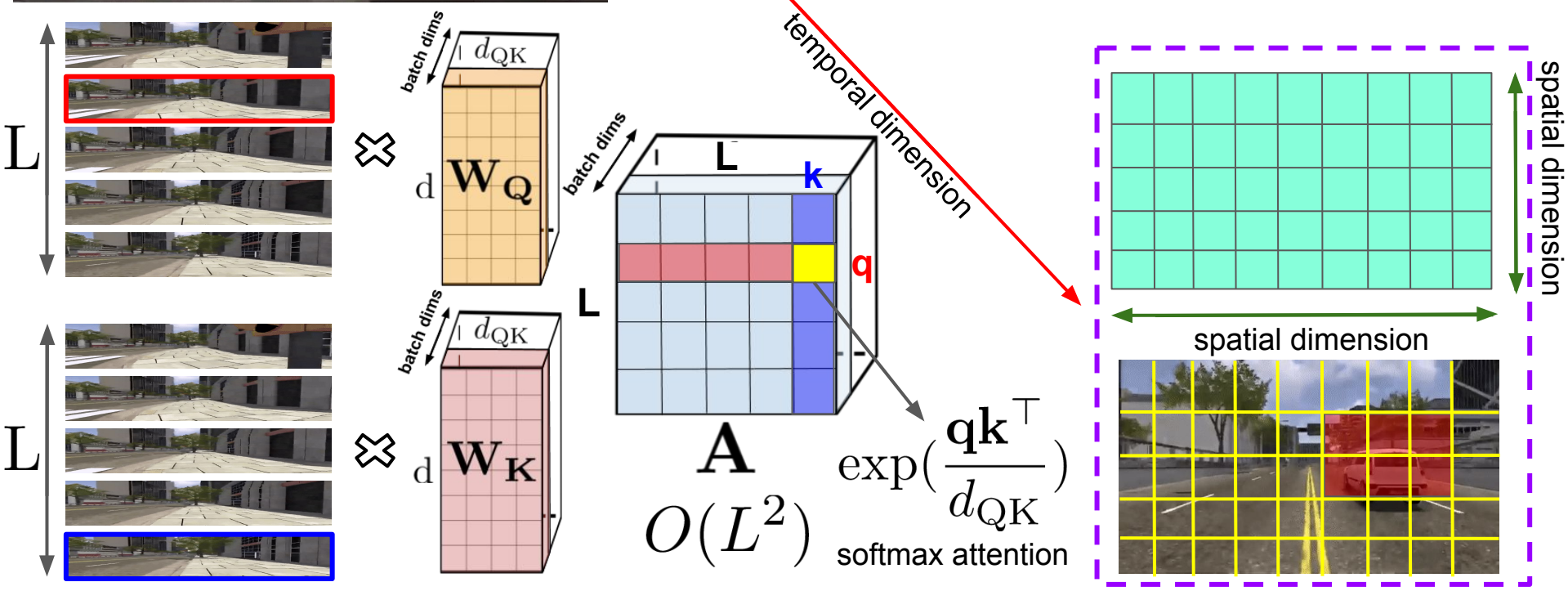
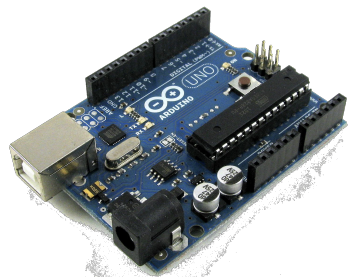
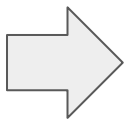
temporal dimension



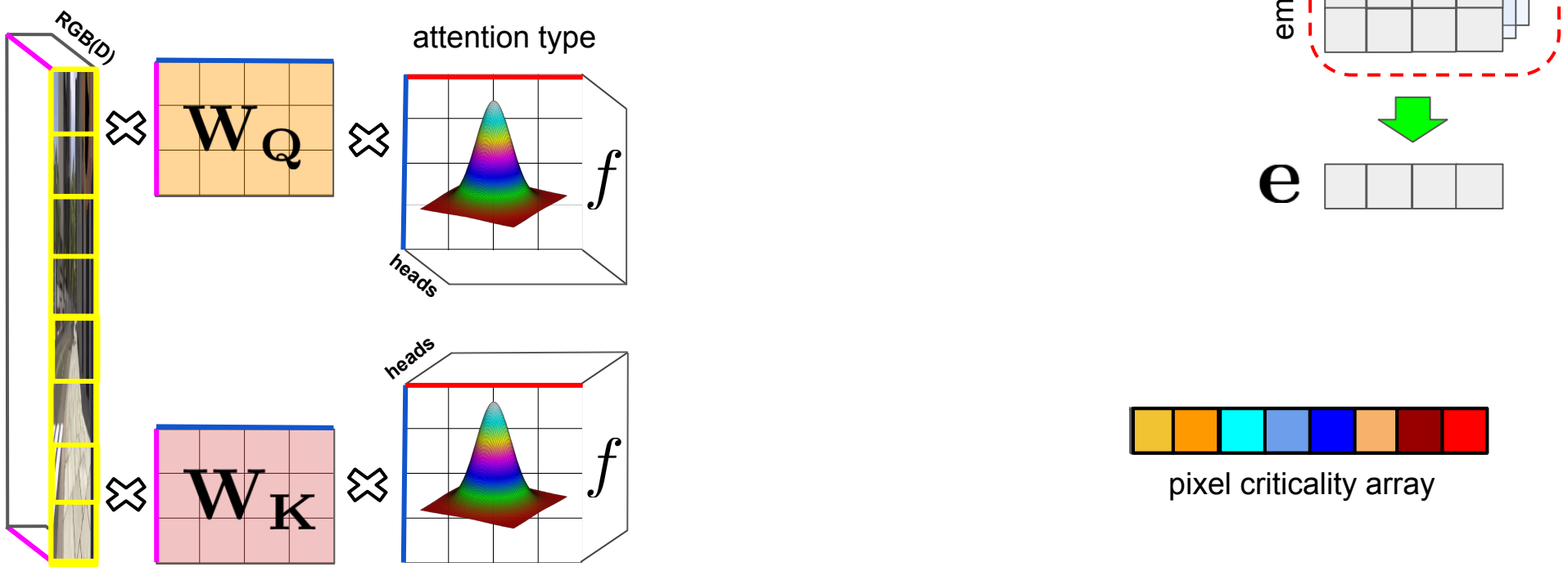
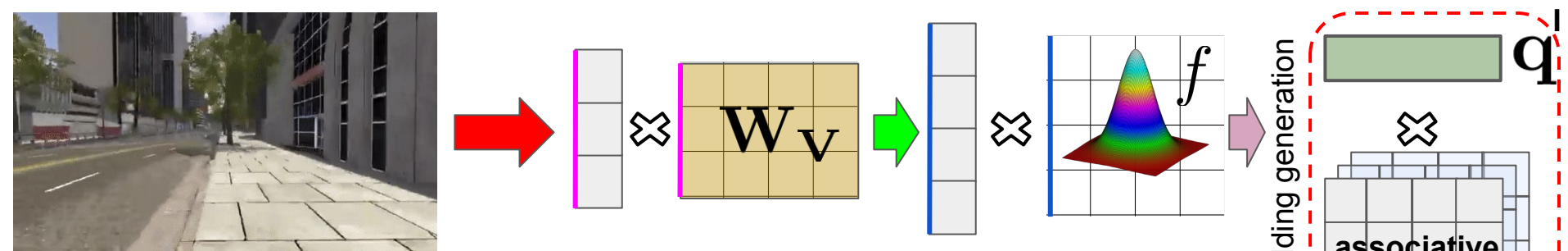
spatial dimension

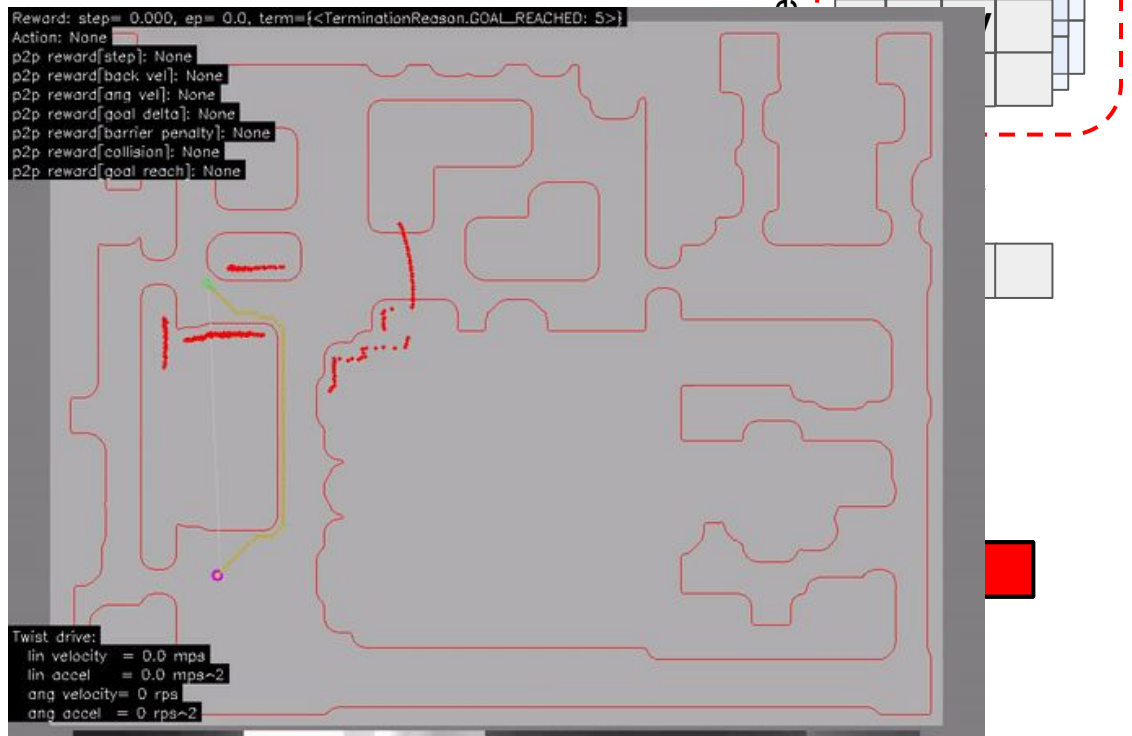
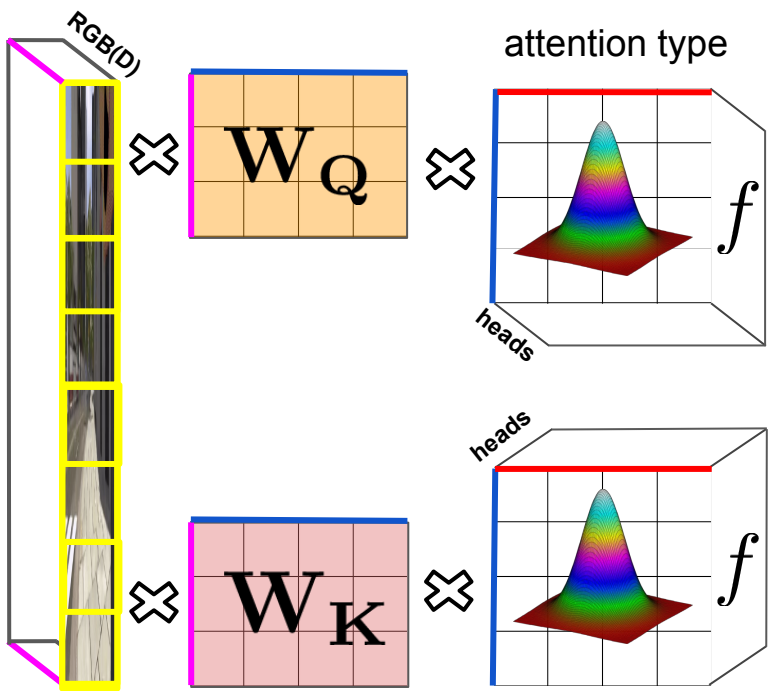
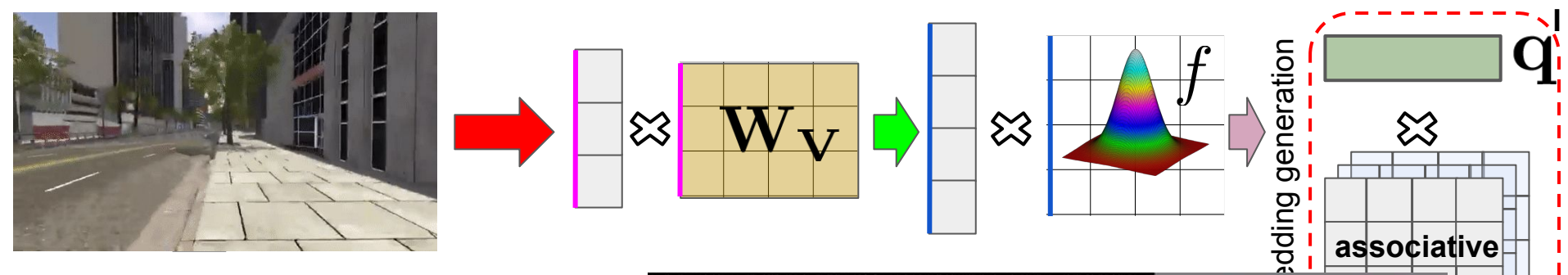












Why did Performers Team start working on these new applications ?

1. **Not a science-fiction:** we have all the tools.
2. **Impact:** if successful, it will revolutionize the way we do machine learning.
3. **The pre-revolution has already started:**

Performers & Reformers: Reformer's reversible layers + Performer's attention = CausalFavor

Performers & Conformers (External): “Performer in Conformer (PIC)” only needs **10 million** params. vs. **116.4 million** param. regular Conformer on *LibriSpeech* corpus. **PIC** (w/ **much smaller model size**) also beats dynamic/lightweight convolution + attention by **20%** in word error.

Thank you for the Attention !

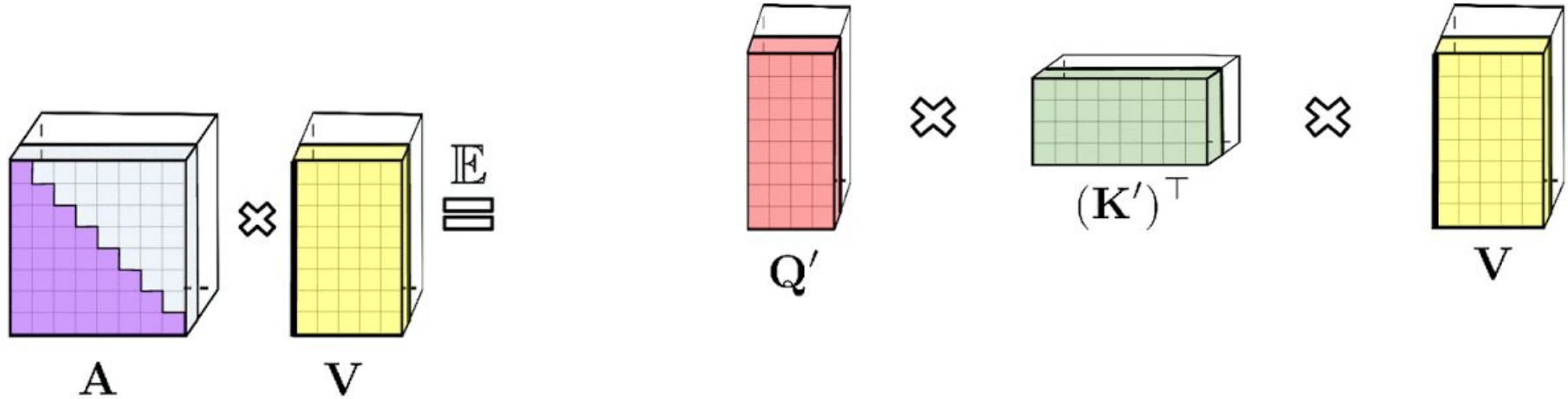


Fig. Linearized softmax causal attention as a prefix-sum computation engine.