# From Attention to Recall

Ayoub Ghriss

April 9, 2025

# Attention

In an autoregressive (causal) attention block after step $N$, given a query $q_t$ $(t = N + 1)$ we compute:

$$
\begin{aligned}
Att(q_t, K, V) &= \frac{1}{s_t} V^\top a_t \in \mathbb{R}^{d_v} \\
a_t &= \exp\left(\frac{Kq_t}{\sqrt{d}}\right), \\
s_t &= {a_t}^\top 1_n
\end{aligned}
$$

where $d$ is the dimension of the QK space and $d_v$ of the V space. The computation and memory cost is $O(Nd)$.
In non-causal (ViT) where each patch attends to all other patches, the computation cost is $O(N^2 d)$ and memory cost is $N(d + d_v)$.

Let's look at ViT attention:

$$Att(Q, K, V) = S^{-1}AV, \; A = \exp\left(\frac{QK^\top}{\sqrt{d}}\right), \; S = A1_n$$

The goal is to replace exp by an efficient activation $f$: $\exp(\langle q, k\rangle) \leftrightarrow \langle \psi_f(q), \psi_f(k)\rangle$ where $\psi_f : \mathbb{R}_d \to \mathbb{R}_q$.
Then

$$A = \psi_f(Q)\psi_f(K)^T = Q_\psi {K_\psi}^\top \qquad \text{row-wise } \psi$$

Then the attention operation becomes:

$$Att(Q, K, V) = \frac{Q_\psi {K_\psi}^\top}{Q_\psi {K_\psi}^\top 1_n} V = \frac{Q_\psi B_\psi}{Q_\psi C_\psi}$$

where $B_\psi = {K_\psi}^\top V \in \mathbb{R}^{d\times d_v}$ and $C_\psi = {K_\psi}^\top 1_n \in \mathbb{R}^d$. Time $O(Nqd_v)$ and space $(q(1 + d_v))$.

For autoregressive transformer, with

$$Att(Q, K, V) = \frac{Q_\psi B_\psi}{Q_\psi C_\psi}$$

it constant time for each new token. But memory-wise, it's only advantageous when $N > \frac{q(1+d_v)}{d+d_v} > 1$, for $d = d_v = q = 128$, $N > 64$ (not bad).

# Random Projection with Johnson-Lindenstrauss

Let's start with Linformers[1]:

- Goal: project a set of $n$ points $X = x_1, ..., x_n \in \mathbb{R}^{n \times d}$ to $\mathbb{R}^{k \times d}$ where $k << n$.
- JL Lemma states that there exists a random projection matrix $\mathbf{P} \in \mathbb{R}^{k \times n}$ (where $k = O(\log n/\epsilon^2)$) such that for any pair $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$, with high probability:

$$(1-\epsilon)\left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2 \leq \left\|\mathbf{P}\mathbf{x}_i - \mathbf{P}\mathbf{x}_j\right\|_2^2 \leq (1+\epsilon)\left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2$$

- The target dimension $k$ is independent of the original dimension $n$.

[1] Sinong Wang et al. "Linformer: Self-Attention with Linear Complexity". In: *CoRR* abs/2006.04768 (2020). eprint: 2006.04768.

# Distributional Johnson-Lindenstrauss (DJL)

Consider $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \subset \mathbb{R}^n$ and $\mathcal{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_M\} \subset \mathbb{R}^n$. Let $\mathbf{E} \in \mathbb{R}^{k\times n}$ and $\mathbf{F} \in \mathbb{R}^{k\times n}$ be two random projection matrices (typically i.i.d. subgaussian).

The DJL guarantees about the preservation of relationships **between** points from $\mathcal{X}$ and points from $\mathcal{Y}$ after projection. For instance, it might guarantee that for any pair $\mathbf{x}_i \in \mathcal{X}$ and $\mathbf{y}_j \in \mathcal{Y}$:

- **Distance Preservation:** $\|\mathbf{E}\mathbf{x}_i - \mathbf{F}\mathbf{y}_j\|_2^2$ is concentrated around $\|\mathbf{x}_i - \mathbf{y}_j\|_2^2$ (perhaps with some scaling factor depending on the normalization of $\mathbf{E}$ and $\mathbf{F}$).
- **Inner Product Preservation:** $\langle \mathbf{E}\mathbf{x}_i, \mathbf{F}\mathbf{y}_j \rangle$ ($\mathbf{x}_i^\top \mathbf{E}^\top \mathbf{F}\mathbf{y}_j$) is close to $\langle \mathbf{x}_i, \mathbf{y}_j \rangle = \mathbf{x}_i^\top \mathbf{y}_j$.
- A common form shows that $\mathbb{E}[\mathbf{x}_i^\top \mathbf{E}^\top \mathbf{F}\mathbf{y}_j]$ is proportional to $\mathbf{x}_i^\top \mathbf{y}_j$.

**Theorem (Linformer Approximation Guarantee)**

*Let the original attention matrix be* $\mathbf{P} = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right) \in \mathbb{R}^{n \times n}$. *Let* $\mathbf{E}, \mathbf{F} \in \mathbb{R}^{k \times n}$ *be two random projection matrices satisfying the distributional Johnson-Lindenstrauss (JL) property. Define the projected Key matrix* $\tilde{\mathbf{K}} = \mathbf{E}\mathbf{K} \in \mathbb{R}^{k \times d}$. *Define the low-rank attention matrix* $\tilde{\mathbf{P}} = \mathrm{softmax}\left(\frac{\mathbf{Q}\tilde{\mathbf{K}}^\top}{\sqrt{d}}\right) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top\mathbf{E}^\top}{\sqrt{d}}\right) \in \mathbb{R}^{n \times k}$.
*Assume the largest singular value* $\sigma_1\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right) = O(n)$. *Then, for any* $0 < \epsilon, \delta < 1$, *with probability at least* $1 - \delta$, *we have:*

$$\left\|\mathbf{P} - \tilde{\mathbf{P}}\mathbf{F}\right\|_2 \leq \epsilon$$

*provided that the projection dimension* $k$ *satisfies:*

$$k = \Omega\left(\frac{d \log(d) n_r(\mathbf{P})}{\epsilon^2} \log(1/\delta)\right)$$

*where* $n_r(\mathbf{P}) = \frac{\|\mathbf{P}\|_F^2}{\|\mathbf{P}\|_2^2}$ *is* $\mathbf{P}$, $\|\cdot\|_2$ *denotes the spectral norm.*

The core idea of Performer is to approximate exp without explicitly computing the $n \times n$ attention matrix. The standard attention matrix is $\mathbf{A} = \mathrm{softmax}(\frac{\mathbf{QK}^\top}{\sqrt{d}})$. Its $(i,j)$-th entry is $A_{ij} = \frac{\exp(\mathbf{q}_i^\top \mathbf{k}_j/\sqrt{d})}{\sum_{l=1}^n \exp(\mathbf{q}_i^\top \mathbf{k}_l/\sqrt{d})}$.
Performer approximates this by finding random feature maps $\phi : \mathbb{R}^d \to \mathbb{R}^m$ such that the kernel $K(\mathbf{q},\mathbf{k}) = \exp(\mathbf{q}^\top \mathbf{k})$ (or related softmax kernel) can be approximated by an inner product in the feature space: $K(\mathbf{q},\mathbf{k}) \approx \phi(\mathbf{q})^\top \phi(\mathbf{k})$.

## Theorem (Unbiased Kernel Approximation via Random Features)

*Let* $K(\mathbf{x}, \mathbf{y}) = \mathbb{E}_{\omega \sim \mathcal{D}}[\zeta(\mathbf{x}, \omega)^\top \zeta(\mathbf{y}, \omega)]$ *be a kernel function expressed as an expectation over random variables* $\omega$ *drawn from a distribution* $\mathcal{D}$*.*
*Consider the random feature map* $\phi : \mathbb{R}^d \to \mathbb{R}^m$ *defined as:*

$$\phi(\mathbf{x}) = \frac{1}{\sqrt{m}} \begin{bmatrix} \zeta(\mathbf{x}, \omega_1) \\ \vdots \\ \zeta(\mathbf{x}, \omega_m) \end{bmatrix}^\top$$

*where* $\omega_1, \ldots, \omega_m$ *are drawn i.i.d. from* $\mathcal{D}$*. Then* $\phi(\mathbf{x})^\top \phi(\mathbf{y})$ *is an unbiased estimator of the kernel* $K(\mathbf{x}, \mathbf{y})$*:*

$$\mathbb{E}_{\omega_1, \ldots, \omega_m}[\phi(\mathbf{x})^\top \phi(\mathbf{y})] = K(\mathbf{x}, \mathbf{y})$$

Specifically, for the Gaussian kernel $K_G(\mathbf{x}, \mathbf{y}) = \exp(\mathbf{x}^\top \mathbf{y})$, one can use trigonometric features: $\zeta(\mathbf{x}, \omega) = [\cos(\omega^\top \mathbf{x}), \sin(\omega^\top \mathbf{x})]$ where $\omega \sim \mathcal{N}(0, \mathbf{I}_d)$.

Furthermore, the softmax kernel[2] $\exp(\mathbf{q}^\top \mathbf{k}/\sqrt{d})$ can be approximated by redefining $\mathbf{q}' = \mathbf{q}/\sqrt[4]{d}$ and $\mathbf{k}' = \mathbf{k}/\sqrt[4]{d}$, and using features for $K(\mathbf{q}', \mathbf{k}') = \exp(\mathbf{q}'^\top \mathbf{k}')$. The paper proposes positive random features (e.g., using $\exp(-\|\mathbf{x}\|^2/2)$ multipliers) to ensure non-negativity suitable for the softmax function:

$$K_{softmax}(\mathbf{q}, \mathbf{k}) \approx \phi_{pos}(\mathbf{q})^\top \phi_{pos}(\mathbf{k})$$

where $\phi_{pos}$ uses features like $\frac{\exp(-\|\mathbf{x}\|^2/2)}{\sqrt{m}}[\exp(\omega_1^\top \mathbf{x}), \ldots, \exp(\omega_m^\top \mathbf{x})]^\top$ (or trigonometric variants).

[2] Krzysztof Marcin Choromanski et al. "Rethinking Attention with Performers". In: *International Conference on Learning Representations*. 2021.

## Corollary (Approximation Error Bound in Performer)

*Let* $\mathbf{A} = \mathrm{softmax}(\frac{\mathbf{QK}^\top}{\sqrt{d}})$ *and* $\hat{\mathbf{A}}$ *be the attention matrix computed using the FAVOR+ mechanism with m positive random features* $\phi$*:*

$$\hat{\mathbf{A}} = \hat{\mathbf{D}}^{-1}\hat{\mathbf{A}}' \quad \textit{where} \quad \hat{\mathbf{A}}' = \phi(\mathbf{Q})\phi(\mathbf{K})^\top \quad \textit{and} \quad \hat{\mathbf{D}} = \mathrm{diag}(\hat{\mathbf{A}}'\mathbf{1}_n)$$

*(*$\phi(\mathbf{Q})$ *applies* $\phi$ *row-wise). Then, under suitable assumptions (e.g., on the norms of* $\mathbf{Q}, \mathbf{K}$*), the approximation error relative to the true attention output* $\mathbf{AV}$ *vs the approximate output* $\hat{\mathbf{A}}\mathbf{V}$ *can be bounded. With high probability (over the choice of random features* $\omega_i$*), the error* $\|\mathbf{A} - \hat{\mathbf{A}}\|$ *(or related output error) decreases roughly as* $O(1/\sqrt{m})$*.*
*Specifically, the paper provides bounds like:*

$$\mathbb{E}\left\|\mathbf{A} - \hat{\mathbf{A}}\right\|_F^2 \leq O\left(\frac{n^2 \cdot poly(d)}{m}\right)$$

## Corollary (Approximation Error Bound in Performer)

*Let* $\mathbf{A} = \mathrm{softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}})$ *and* $\hat{\mathbf{A}}$ *be the attention matrix computed using the FAVOR+ mechanism with* $m$ *positive random features* $\phi$*:*

$$\hat{\mathbf{A}} = \hat{\mathbf{D}}^{-1}\hat{\mathbf{A}}' \quad \textit{where} \quad \hat{\mathbf{A}}' = \phi(\mathbf{Q})\phi(\mathbf{K})^\top \quad \textit{and} \quad \hat{\mathbf{D}} = \mathrm{diag}(\hat{\mathbf{A}}'\mathbf{1}_n)$$

*(*$\phi(\mathbf{Q})$ *applies* $\phi$ *row-wise). Then, under suitable assumptions (e.g., on the norms of* $\mathbf{Q}, \mathbf{K}$*), the approximation error relative to the true attention output* $\mathbf{AV}$ *vs the approximate output* $\hat{\mathbf{A}}\mathbf{V}$ *can be bounded. With high probability (over the choice of random features* $\omega_i$*), the error* $\|\mathbf{A} - \hat{\mathbf{A}}\|$ *(or related output error) decreases roughly as* $O(1/\sqrt{m})$*.*
*Specifically, the paper provides bounds like:*

$$\mathbb{E}\left\|\mathbf{A} - \hat{\mathbf{A}}\right\|_F^2 \leq O\left(\frac{n^2 \cdot poly(d)}{m}\right)$$

The exact form depends on specific assumptions and normalizations). The approximation accuracy improves as the number of random features $m$ increases for a fixed $n$.

A key contribution of FAVOR+ is the use of **orthogonal random features**. Using random features that are (approximately) orthogonal can significantly reduce the variance of the estimator $\phi(\mathbf{x})^\top \phi(\mathbf{y})$ for the same number of features $m$.

**Theorem (Variance Reduction with Orthogonal Features (Conceptual))**

*Let $\phi(\mathbf{x})$ and $\tilde{\phi}(\mathbf{x})$ be two random feature constructions of dimension $m$, both providing unbiased estimates of a kernel $K(\mathbf{x}, \mathbf{y})$. If $\tilde{\phi}$ uses random projection vectors $\omega_1, \ldots, \omega_m$ that are sampled to be (stochastically) orthogonal or near-orthogonal, while $\phi$ uses i.i.d. samples, then the variance of the estimator based on $\tilde{\phi}$ can be significantly lower:*

$$Var[\tilde{\phi}(\mathbf{x})^\top \tilde{\phi}(\mathbf{y})] < Var[\phi(\mathbf{x})^\top \phi(\mathbf{y})]$$

*This leads to a more accurate approximation of the attention matrix $\mathbf{A}$ for a fixed projection dimension $m$, or allows achieving the same accuracy with a smaller $m$.*

Performers: provide the foundation for replacing the $O(n^2 d)$ computation of the standard attention mechanism with an $O(nmd)$ computation using the FAVOR+ mechanism, achieving linear complexity in sequence length $n$ (assuming $m \ll n$). The approximation error is controllable by adjusting $m$.

### Definition (cosFormer Attention Mechanism)

Let $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{N \times d}$. The output for the $i$-th query vector $\mathbf{q}_i$ is defined as:

$$\mathrm{Attn}(\mathbf{q}_i, \mathbf{K}, \mathbf{V}) = \frac{\sum_{j=1}^{N} \cos\left(\omega(i,j)\right) \cdot \mathrm{ReLU}(\mathbf{q}_i)\mathrm{ReLU}(\mathbf{k}_j) \odot \mathbf{v}_j}{\sum_{j=1}^{N} \cos\left(\omega(i,j)\right) \cdot \mathrm{ReLU}(\mathbf{k}_j)}$$

where $\omega(i-j) = \frac{\pi}{2M}(i-j)$ where $M$ is a hyperparameter.

$$\mathrm{Output}_i = \sum_{j=1}^{N} \cos(\omega(i-j)) \cdot w_j \cdot v_j$$

## Definition (cosFormer Attention Mechanism)

Let $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{N\times d}$. The output for the $i$-th query vector $\mathbf{q}_i$ is defined as:

$$\text{Attn}(\mathbf{q}_i, \mathbf{K}, \mathbf{V}) = \frac{\sum_{j=1}^{N} \cos\left(\omega(i,j)\right) \cdot \text{ReLU}(\mathbf{q}_i)\text{ReLU}(\mathbf{k}_j) \odot \mathbf{v}_j}{\sum_{j=1}^{N} \cos\left(\omega(i,j)\right) \cdot \text{ReLU}(\mathbf{k}_j)}$$

where $\omega(i-j) = \frac{\pi}{2M}(i-j)$ where $M$ is a hyperparameter.

$$\text{Output}_i = \sum_{j=1}^{N} \cos(\omega(i-j)) \cdot w_j \cdot v_j$$

Any issue with this approach?

**Theorem (Linear Complexity of cosFormer Attention)**

*Assume the positional function depends only on the relative position, $\omega(i,j) = \omega(i-j)$. The cosFormer attention output (specifically the numerator sum $\sum_{j=1}^{N} \cos(\omega(i-j)) \cdot w_j v_j$ for each query $i$) can be computed for all $N$ queries simultaneously in $\mathcal{O}(Nd)$ time complexity (assuming $w_j, v_j$ are derived from $\mathbf{K}, \mathbf{V}$ in linear time).*

# Proof sketch

Using the cosine angle subtraction formula $\cos(a-b) = \cos(a)\cos(b) + \sin(a)\sin(b)$, we can rewrite the sum:

$$\begin{aligned}\text{Output}_i &= \sum_{j=1}^{N} \cos(\omega(i) - \omega(j)) \cdot (w_j v_j) \quad \text{(assuming } \omega(i-j) \text{ can be written this way)} \\ &= \sum_{j=1}^{N} [\cos(\omega(i))\cos(\omega(j)) + \sin(\omega(i))\sin(\omega(j))] \cdot (w_j v_j) \ldots\end{aligned}$$

Let:

$$\mathbf{S}_C = \sum_{j=1}^{N} \cos(\omega(j)) \cdot (w_j v_j) \quad \in \mathbb{R}^d$$

$$\mathbf{S}_S = \sum_{j=1}^{N} \sin(\omega(j)) \cdot (w_j v_j) \quad \in \mathbb{R}^d$$

These two sums, $\mathbf{S}_C$ and $\mathbf{S}_S$, depend only on the keys and values and can be computed once for all queries in $\mathcal{O}(Nd)$ time.
The output for the $i$-th query is then:

$$\text{Output}_i = \cos(\omega(i))\mathbf{S}_C + \sin(\omega(i))\mathbf{S}_S$$

This final step takes $\mathcal{O}(d)$ time per query $i$. Computing this for all $N$ queries takes $\mathcal{O}(Nd)$ time.
The denominator $Z_i = \sum_{j=1}^{N} \cos(\omega(i-j)) w_j$ can be computed similarly in $\mathcal{O}(N)$ time (if $w_j$ is scalar) or $\mathcal{O}(Nd)$ (if vector re-weighting).
Therefore, the overall complexity is dominated by the computation of the sums $\mathbf{S}_C, \mathbf{S}_S$ (and similar sums for the denominator), resulting in $\mathcal{O}(Nd)$ complexity.

- **Connection to RoPE:** The use of cosine functions with relative positions $\omega(i-j)$ is conceptually linked to Rotary Position Embeddings (RoPE), which also uses sinusoidal functions to inject relative positional information effectively.
- **ReLU Re-weighting:** The $\mathrm{ReLU}(\mathbf{k}_j)$ term is motivated as a simple, non-linear gating mechanism to focus attention on relevant key-value pairs, loosely analogous to the data-dependent normalization performed by the softmax denominator.

**Implication:** Theorem 6 demonstrates that the cosFormer attention mechanism, by definition, avoids the quadratic complexity bottleneck of standard attention and achieves linear time complexity $\mathcal{O}(Nd)$ with respect to sequence length $N$. The paper then empirically shows that this formulation performs competitively with standard attention.

# Architecture

- Single layer of neurons (often binary: +1/-1 or 0/1). Let's assume bipolar (+1/-1) neurons.
- Fully connected: Every neuron is connected to every other neuron.
- **Symmetric Weights:** The weight from neuron $i$ to $j$ is the same as from $j$ to $i$ ($w_{ij} = w_{ji}$). This is crucial for stability.
- **No Self-Connections:** Neurons do not connect to themselves ($w_{ii} = 0$).
- The units of a neural net imitate the neurons.

# Storing Patterns (Learning)

Hopfield networks store patterns using a Hebbian-like rule (often the outer product rule). To store $P$ patterns $\xi^1, \xi^2, ..., \xi^P$, where each pattern $\xi^\mu$ is a vector of $N$ bipolar values (+1/-1): $\xi^\mu = (\xi_1^\mu, \xi_2^\mu, ..., \xi_N^\mu)$ (N output dimension of a layer)
The weight between neuron $i$ and $j$ is calculated as:

$$w_{ij} = \frac{1}{N} \sum_{\mu=1}^{P} \xi_i^\mu \xi_j^\mu \quad (\text{for } i \neq j)$$

And $w_{ii} = 0$.

- This sums the Hebbian products ($\xi_i^\mu \xi_j^\mu$) over all patterns to be stored.
- If two neurons have the same state (+1/+1 or -1/-1) in many patterns, their connection weight will be positive (excitatory).
- If they have opposite states (+1/-1 or -1/+1) in many patterns, their weight will be negative (inhibitory).

# Retrieving Patterns (Recall / Dynamics)

Retrieval starts by setting the network state to an initial (potentially noisy or incomplete) pattern.

Neurons then update their states iteratively until the network converges to a stable state.

**Update Rule (Asynchronous):** Pick a neuron $i$ at random. Update its state $s_i$ based on the weighted sum of inputs from other neurons:

1. Calculate the input sum: $h_i = \sum_{j \neq i} w_{ij} s_j$
2. Apply the threshold function: $s_i(t+1) = \text{sgn}(h_i) = \begin{cases} +1 & \text{if } h_i \geq 0 \\ -1 & \text{if } h_i < 0 \end{cases}$ (Or keep $s_i(t)$ if $h_i = 0$ in some definitions)

Repeat until no neuron changes its state.

**Goal:** The network should converge to the stored pattern closest (in Hamming distance) to the initial state. These stored patterns act as **attractors**.

We can draw an analogy between the "kernelized" attention and Hebbian learning (recall):

- Memorization : $B \leftarrow B + \psi(K)V$, $C \leftarrow C + \psi(K)$
- Recall: $\frac{QB}{C}$

- All these methods restrict to feature maps in $\mathbb{R}^+$. Other than Performers, the other methods do not justify their choice
- Can we do better than $\mathrm{softmax}$ in finite dimension?
- Can we find a non-positive feature map $\psi_f$ that keeps $f$ non-negative?

We claim that any activation of the form $(x, y) \mapsto f(< x, y >)$ where $f$ is polynomial of degree $m$ requires $q = O(d^m)$. That is, if $f$ is analytical then $\psi_f$ maps to an infinite dimensional space.
$f_p(x, y) = (< x_p >)^p$ can be generated using the feature map $\psi$ that maps $x$ to the terms of $(\sum_i x_i)^p$.

We claim that if $q$ is finite and verifies $\langle\psi(x), \psi(y)\rangle > 0, \forall x, y$
then there is an orthogonal matrix $O \in \mathbb{R}^{q\times q}$ such that $x \mapsto O\psi(x)$ has its image in the positive orthant.

- $\langle\psi(x), \psi(y)\rangle > 0, \forall x, y$: image of $\psi$ is included in a non-obtuse cone, we can convexity of it to work with a larger closed convex cone $K$.
- How do we prove that $K$ can be rotated into the positive orthant: I think so

Take $K$ non-obtuse closed convex cone in $\mathbb{R}_2$, let $u$ be an extreme ray of $C$:
Take $v = u^\perp$ and project $C$ on $v$. We should now prove that $P(C)$ is also obtuse cone of $R^1$, if $x^\perp$ in $C^\perp$, then it's a projection of an element in $C$ and due to the linearity of the projection, the properties apply.
Now we need to show it's obtuse: $\langle x^\perp, y^\perp \rangle > 0$

$$\langle x, y \rangle = \langle x_u, y_u \rangle + \langle x_v, y_v \rangle$$

$\langle x^\perp, y^\perp \rangle < 0$ if and only if $\langle x, y \rangle \leq \langle x_u, y_u \rangle$
Then $x = x_u + x_v$ and $y = y_u + y_v$ where $x_v$ and $y_v$ are opposite.
We have $u$ minimizes $< u, z >$ for some $z = z_u + z_v$
take $w = \alpha x + (1 - \alpha) y$ normalized.
Then $< z, w > = < w_u, z_u > + < w_v, z_v >$.
however, $u$ is extremal, that is:
Update: doesn't work. Take second order cone, it's still non-obtuse,but not inside the positive orthant. Generally, $\psi$ has to map to self-dual cone, but the positive orthant is just one of them.