HYBRID RANDOM FEATURES

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a new class of random feature methods for linearizing softmax and Gaussian kernels called *hybrid random features* (HRFs) that automatically adapt the quality of kernel estimation to provide most accurate approximation in the defined regions of interest. Special instantiations of HRFs lead to well-known methods such as trigonometric (Rahimi & Recht, 2007) or (recently introduced in the context of linear-attention Transformers) positive random features (Choromanski et al., 2021b). By generalizing Bochner's Theorem for softmax/Gaussian kernels and leveraging random features for compositional kernels, the HRF-mechanism provides strong theoretical guarantees - unbiased approximation and strictly smaller worst-case relative errors than its counterparts. We conduct exhaustive empirical evaluation of HRF ranging from pointwise kernel estimation experiments, through tests on data admitting clustering structure to benchmarking implicit-attention Transformers (also for downstream Robotics applications), demonstrating its quality in a wide spectrum of machine learning problems.

1 INTRODUCTION & RELATED WORK

Consider the *softmax* and *Gaussian kernel* functions $\mathrm{K} : \mathbb{R}^{d \times d} \rightarrow \mathbb{R}$ defined as follows:

$$\mathrm{SM}(\mathbf{x}, \mathbf{y}) \stackrel{\mathrm{def}}{=} \exp(\mathbf{x}^\top \mathbf{y}), \qquad \mathrm{K}_{\mathrm{gauss}} \stackrel{\mathrm{def}}{=} \exp(-\frac{\|\mathbf{x}-\mathbf{y}\|_2^2}{2}). \tag{1}$$

These two are prominent examples of functions used in the so-called *kernel methods* (Gretton et al., 2005; Zhang et al., 2018) and beyond, i.e. in softmax-sampling (Blanc & Rendle, 2018). Random features (RFs, Rahimi & Recht, 2007) yield a powerful mechanism for linearizing and consequently scaling up kernel methods with dot-product kernel decompositions disentangling $\mathbf{x}$ from $\mathbf{y}$ in the formulae for kernel value: $\mathrm{K}(\mathbf{x}, \mathbf{y}) \approx \phi(\mathbf{x})^\top \phi(\mathbf{y})$ via data-agnostic probabilistic (random feature) maps $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^m$. The tight relationship between softmax and Gaussian kernels given by the transformation: $\mathrm{K}_{\mathrm{gauss}}(\mathbf{x}, \mathbf{y}) = \exp(-\frac{\|\mathbf{x}\|_2^2}{2})\mathrm{SM}(\mathbf{x}, \mathbf{y})\exp(-\frac{\|\mathbf{y}\|_2^2}{2})$ provides a mapping of any random feature vector $\phi_{\mathrm{SM}}(\mathbf{u})$ for the softmax kernel to the corresponding one $\phi_{\mathrm{gauss}}(\mathbf{u}) = \exp(-\frac{\|\mathbf{u}\|_2^2}{2})\phi_{\mathrm{SM}}(\mathbf{u})$ for the Gaussian kernel, thus we will focus on the former kernel. The classic random feature map mechanism ϕ_m^{trig} for the softmax kernel, obtained from Bochner's Theorem applied to the Gaussian kernel (Rahimi & Recht, 2007), is of the form:

$$\phi_m^{\mathrm{trig}}(\mathbf{u}) = \frac{1}{\sqrt{m}} \exp(\frac{\|\mathbf{u}\|^2}{2}) \left(\sin(\omega_1^\top \mathbf{u}), ..., \sin(\omega_m^\top \mathbf{u}), \cos(\omega_1^\top \mathbf{u}), ..., \cos(\omega_m^\top \mathbf{u})\right)^\top, \tag{2}$$

where m stands for the number of random features and $\omega_1, ..., \omega_m \stackrel{\mathrm{iid}}{\sim} \mathcal{N}(0, \mathbf{I}_d)$.

The above is the most straightforward method for linearizing softmax and Gaussian kernels that is applied in several downstream use-cases, yet it was recently shown that in some of the new impactful applications of scalable kernel methods – such as implicit-attention Transformer architectures called Performers (Choromanski et al., 2021b) – the above technique does not work. Although the mechanism provides unbiased estimation, it struggles to accurately approximate close-to-zero kernel values, as characterized by the particularly large mean squared error (MSE) in that region. This is a crucial problem since most of the entries of the attention matrices in Transformers' models are very small and the approximators need to be particularly accurate there. Otherwise the renormalizers computed to make attention matrices row-stochastic (standard attention normalization procedure in Transformers) would be estimated imprecisely, potentially even by negative values.

The solution to this problem was presented in FAVOR+ mechanism (Choromanski et al., 2021b), where a new positive random feature map for unbiased softmax kernel estimation was applied:

$$\phi_m^{++}(\mathbf{u}) = \frac{1}{\sqrt{2m}} \exp(-\frac{\|\mathbf{u}\|^2}{2}) \left(\exp(\omega_1^\top \mathbf{u}), ..., \exp(\omega_m^\top \mathbf{u}), \exp(-\omega_1^\top \mathbf{u}), ..., \exp(-\omega_m^\top \mathbf{u})\right)^\top . \quad (3)$$

Even though, very accurate in estimating small softmax kernel values (which turns out to be crucial in making RFs work for Transformers training), this mechanism is characterized by larger MSE for large kernel values. In several applications of softmax kernels (in particular Transformers, where attention matrices typically admit sparse combinatorial structure with relatively few but critical large entries and several close-to-zero ones or softmax sampling) the algorithm needs to process simultaneously very small and large kernel values. The following natural questions arise:

Is it possible to get the best from both the mechanisms to obtain RFs-based estimators particularly accurate in both the antipodal regions: very small and large softmax kernel values ? Furthermore, can those estimators be designed to have low variance in more generally pre-defined regions ?

We give affirmative answers to both of the above questions by constructing a new class of random feature maps techniques called *hybrid random features* or HRFs. Theoretical methods used by us to develop HRFs: (a) provide a unifying perspective, where trigonometric random features from Bochner's Theorem and novel mechanisms proposed in (Choromanski et al., 2021b) are just special corollaries of the more general result from complex analysis, (b) integrate in the original way several other powerful probabilistic techniques such as Goemans-Willimson method (Goemans & Williamson, 2004) and random features for the compositional kernels (Daniely et al., 2017).

We provide detailed theoretical analysis of HRFs, showing in particular that they provide strictly more accurate worst-case softmax kernel estimation than previous algorithms and lead to computational gains. We also conduct their thorough empirical evaluation on tasks ranging from pointwise kernel estimation to downstream problems involving training Transformer-models or even end-to-end robotic controller-stacks including attention-based architectures.

Related Work: The literature on different random feature map mechanisms for Gaussian (and thus also softmax) kernel estimation is voluminous. Most focus has been put on reducing the variance of trigonometric random features from (Rahimi & Recht, 2007) via various Quasi Monte Carlo (QMC) methods, where directions and/or lengths of Gaussian vectors used to produce features are correlated, often through geometric conditions such as orthogonality (Choromanski et al., 2017; Rowland et al., 2018; Yu et al., 2016; Choromanski et al., 2019; Choromanski & Sindhwani, 2016). Our HRFs do not compete with those techniques (and can be in fact easily combined with them) since rather than focusing on improving sampling mechanism for a given approximation algorithm, they provide a completely new algorithm. The new application of random features for softmax kernel in Transformers proposed in (Choromanski et al., 2021b) led to the fruitful research on the extensions and limitations of these methods. In (Schlag et al., 2021) random features were replaced by sparse deterministic constructions (no longer approximating softmax kernel). In (Luo et al., 2021) it was observed that combining L_2-normalization of queries and keys for variance reduction of softmax kernel estimation with FFT-based implementations of relative position encoding and FAVOR+ mechanism from Performers helps in training. Trigonometric random features were applied for softmax sampling in (Rawat et al., 2019). Several other techniques such as Nyström method (Yang et al., 2012; Williams & Seeger, 2000; Rudi et al., 2015) were proposed to construct data-dependent feature representations. Even though, as we show in Sec. 2.3.1, certain instantiations of the HRF mechanism clearly benefit from some data analysis, our central goal remains an unbiased estimation of the softmax/Gaussian kernels which is no longer the case for those other techniques.

2 Hybrid Random Features

2.1 Preliminaries

We start with the following basic definitions and results.

Definition 2.1 (Kernel with a Random Feature Map Representation)**.** *We say that a kernel function* $\mathrm{K} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ *admits a random feature (RF) map representation if it can be written as:*

$$\mathrm{K}(\mathbf{x}, \mathbf{y}) = \mathbb{E}_{\omega \sim \Omega} \left[\sum_{i=1}^{l} \xi_i(\mathbf{x}, \omega) \xi_i(\mathbf{y}, \omega) \right], \quad (4)$$

for some $\xi_i : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$*, and where* ω *is sampled from some probabilistic distribution* $\Omega \in \mathcal{P}(\mathbb{R}^d)$*. The corresponding random feature map, for a given* $m \in \mathbb{N}$*, is defined as:*

$$\phi_m(\mathbf{u}) = \frac{1}{\sqrt{m}} \phi_m^1(\mathbf{u}) \star ... \star \phi_m^l(\mathbf{u}) \in \mathbb{R}^{ml}, \tag{5}$$

where $\phi_m^i = (\xi_i(\mathbf{u}, \omega_1), ..., \xi_i(\mathbf{u}, \omega_m))^\top$*,* $\star$ *stands for vertical concatenation, and* $\omega_1, ..., \omega_m \overset{\text{iid}}{\sim} \Omega$*.*

Random feature maps can be used to unbiasedly approximate corresponding kernels:

$$\widehat{\mathrm{K}}(\mathbf{x}, \mathbf{y}) = \phi_m(\mathbf{x})^\top \phi_m(\mathbf{y}). \tag{6}$$

Using the above notation, a trigonometric random feature ϕ_m^{trig} from Equation 2 can be encoded as applying: $l = 2$, $\xi_1(\mathbf{u}, \omega) = \sin(\omega^\top \mathbf{u})$, $\xi_2(\mathbf{u}, \omega) = \cos(\omega^\top \mathbf{u})$. Similarly, positive random features can be encoded as taking: $l = 2$ and $\xi_1(\mathbf{u}, \omega) = \frac{1}{\sqrt{2}} \exp(\omega^\top \mathbf{u})$, $\xi_2(\mathbf{u}, \omega) = \frac{1}{\sqrt{2}} \exp(-\omega^\top \mathbf{u})$.

The following result from (Choromanski et al., 2021b) shows that the mean squared error (MSE) of the trigonometric estimator is small for large softmax kernel values and large for small softmax kernel values, whereas an estimator applying positive random features behaves in the opposite way.

Lemma 2.2 (positive versus trigonometric RFs)**.** *Denote by* $\widehat{\mathrm{SM}}_m^{\text{trig}}(\mathbf{x}, \mathbf{y})$ *an estimator of* $\mathrm{SM}(\mathbf{x}, \mathbf{y})$ *for* $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ *using trigonometric RFs and* $\omega_1, ..., \omega_m \overset{\text{iid}}{\sim} \mathcal{N}(0, \mathbf{I}_d)$*. Denote by* $\widehat{\mathrm{SM}}_m^{++}(\mathbf{x}, \mathbf{y})$ *its analogue using positive RFs. Take:* $\Delta = \mathbf{x} - \mathbf{y}$*,* $\mathbf{z} = \mathbf{x} + \mathbf{y}$*,* $f_1(u) = (2m)^{-1} \exp(u^2) \mathrm{SM}^{-2}(\mathbf{x}, \mathbf{y})$*,* $f_2(u) = (2m)^{-1} \exp(u^2) \mathrm{SM}^2(\mathbf{x}, \mathbf{y})$*,* $f_3(u) = (1 - \exp(-u^2))^2$*. The MSEs of these estimators are:*

$$\mathrm{MSE}(\widehat{\mathrm{SM}}_m^{\text{trig}}(\mathbf{x}, \mathbf{y})) = f_1(\|\mathbf{z}\|_2) f_3(\|\Delta\|_2), \ \mathrm{MSE}(\widehat{\mathrm{SM}}_m^{++}(\mathbf{x}, \mathbf{y})) = f_2(\|\mathbf{z}\|_2) f_3(\|\mathbf{z}\|_2). \tag{7}$$

2.2 Towards Hybrid Random Features: Complex Exponential Estimators

The following lemma is a gateway to construct unbiased estimators of the softmax kernel.

Lemma 2.3. *Let* $\omega \sim \mathcal{N}(0, \mathbf{I}_d)$*. Then for every* $\mathbf{z} = (z_1, ..., z_d)^\top \in \mathbb{C}^d$ *the following holds:*

$$\mathbb{E}\left[\exp(\omega^\top \mathbf{z})\right] = \exp\left(\frac{\sum_{i=1}^d z_i^2}{2}\right). \tag{8}$$

Both $\widehat{\mathrm{SM}}^{\text{trig}}$ and $\widehat{\mathrm{SM}}^{++}$ can be easily derived from the above formula by taking $\mathbf{z} = \mathbf{i}(\mathbf{x} - \mathbf{y})$ and $\mathbf{z} = \mathbf{x} + \mathbf{y}$ respectively. Lemma 2.3 is however much more powerful, as we show next. For a complex vector $\mathbf{z} = (z_1, ..., z_d) \in \mathbb{C}^d$, we denote: $\mathbf{z}^2 = \sum_{i=1}^d z_i^2$. Consider $\mathbf{z} = \mathbf{A}\mathbf{x} + (\mathbf{A}^\top)^{-\mathbf{1}} \mathbf{y}$ for an invertible (in $\mathbb{C}^{d \times d}$) matrix $\mathbf{A} \in \mathbb{C}^{d \times d}$. Using Lemma 2.3, we get:

$$\exp\left(\frac{(\mathbf{A}\mathbf{x})^2}{2}\right) \exp\left(\frac{((\mathbf{A}^\top)^{-1}\mathbf{y})^2}{2}\right) \mathrm{SM}(\mathbf{x}, \mathbf{y}) = \mathbb{E}[\exp(\omega^\top (\mathbf{A}\mathbf{x} + (\mathbf{A}^\top)^{-1}\mathbf{y})] \tag{9}$$

Thus for $\Psi_{\mathbf{M}}^m(\mathbf{u}) \stackrel{\text{def}}{=} \frac{1}{\sqrt{m}} \exp(-\frac{(\mathbf{M}\mathbf{u})^2}{2})(\exp(\omega_1^\top \mathbf{M}\mathbf{u}), ..., \exp(\omega_m^\top \mathbf{M}\mathbf{u}))^\top$ and $\omega_i \sim \mathcal{N}(0, \mathbf{I}_d)$:

$$\mathrm{SM}(\mathbf{x}, \mathbf{y}) = \mathbb{E}[\Psi_{\mathbf{A}}^m(\mathbf{x})^\top \Psi_{(\mathbf{A}^\top)^{-1}}^m(\mathbf{y})]. \tag{10}$$

We obtain the new random feature map mechanism providing unbiased estimation of the softmax kernel. In general it is *asymmetric* since it applies different parameterization for $\mathbf{x}$ and $\mathbf{y}$: one using $\mathbf{A}$, the other one $(\mathbf{A}^\top)^{-1}$. Asymmetric random features is a simple generalization of the model using the same ϕ for both $\mathbf{x}$ and $\mathbf{y}$, presented earlier. The resulting estimator $\widehat{\mathrm{SM}}_m^{\text{cexp}}(\mathbf{x}, \mathbf{y})$, that we call *complex exponential*, is defined as:

$$\widehat{\mathrm{SM}}_m^{\text{cexp}}(\mathbf{x}, \mathbf{y}) \stackrel{\text{def}}{=} \Psi_{\mathbf{A}}^m(\mathbf{x})^\top \Psi_{(\mathbf{A}^\top)^{-1}}^m(\mathbf{y}) \tag{11}$$

For $\mathbf{A} = \mathbf{i}\mathbf{I}_d$, estimator $\widehat{\mathrm{SM}}_m^{\text{cexp}}(\mathbf{x}, \mathbf{y})$ becomes $\widehat{\mathrm{SM}}_m^{\text{trig}}(\mathbf{x}, \mathbf{y})$ and for $\mathbf{A} = \mathbf{I}_d$, it becomes $\widehat{\mathrm{SM}}_m^{++}(\mathbf{x}, \mathbf{y})$. Note also that an estimator based on this mechanism has variance equal to zero if:

$$\mathbf{x} = -(\mathbf{A}^{-1})(\mathbf{A}^\top)^{-1}\mathbf{y} \tag{12}$$

In fact we can relax that condition. It is easy to check that for the variance to be zero, we only need:

$$\mathrm{Re}(\mathbf{A})\mathbf{x} + \mathrm{Re}((\mathbf{A}^T)^{-1})\mathbf{y} = 0 \text{ and } \mathrm{Im}(\mathbf{A})\mathbf{x} + \mathrm{Im}((\mathbf{A}^T)^{-1})\mathbf{y} = 0 \tag{13}$$

2.3 THE ALGORITHM

We are ready to present the mechanism of *Hybrid Random Features* (HRFs). Denote by $\mathcal{E} = (\widehat{\mathrm{SM}}^k(\mathbf{x},\mathbf{y}))_{k=1}^{p+1}$ a list of estimators of $\mathrm{SM}(\mathbf{x},\mathbf{y})$ (the so-called *base estimators*) and by $\Lambda = (\widehat{\lambda}^k(\mathbf{x},\mathbf{y}))_{k=1}^{p}$ a list of estimators of: $\{\lambda^k(\mathbf{x},\mathbf{y})\}_{k=1}^{p}$ for some functions $\lambda^k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, constructed independently from $\mathcal{E}$. Take the following estimator of $\mathrm{SM}(\mathbf{x},\mathbf{y})$:

$$\widehat{\mathrm{SM}}^{\mathcal{E},\Lambda}(\mathbf{x},\mathbf{y}) = \sum_{k=1}^{p} \widehat{\lambda}^k(\mathbf{x},\mathbf{y})\widehat{\mathrm{SM}}^k(\mathbf{x},\mathbf{y}) + \left(1 - \sum_{k=1}^{p} \widehat{\lambda}^k(\mathbf{x},\mathbf{y})\right)\widehat{\mathrm{SM}}^{p+1}(\mathbf{x},\mathbf{y}) \tag{14}$$

We call $\widehat{\mathrm{SM}}^{\mathcal{E},\Lambda}(\mathbf{x},\mathbf{y})$ a hybrid random feature (HRF) estimator of $\mathrm{SM}(\mathbf{x},\mathbf{y})$ parameterized by $\mathcal{E},\Lambda$. Note that if elements of $\mathcal{E}$ are unbiased estimators of $\mathrm{SM}(\mathbf{x},\mathbf{y})$, then trivially $\widehat{\mathrm{SM}}^{\mathcal{E},\Lambda}(\mathbf{x},\mathbf{y})$ is also an unbiased estimator of $\mathrm{SM}(\mathbf{x},\mathbf{y})$. Assume that each $\widehat{\mathrm{SM}}^k(\mathbf{x},\mathbf{y})$ is of the form: $\widehat{\mathrm{SM}}^k(\mathbf{x},\mathbf{y}) = (\alpha_m^k(\mathbf{x}))^\top \beta_m^k(\mathbf{y})$ for $\phi_m^k(\mathbf{u}) = \frac{1}{\sqrt{m}}\phi_m^{1,k}(\mathbf{u}) \star ... \star \phi_m^{t_k,k}(\mathbf{u})$, $t_k > 0$ and $\phi_m^{1,k}, ..., \phi_m^{t_k,k} : \mathbb{R}^d \to \mathbb{R}^m$, where $\phi \in \{\alpha,\beta\}$. Assume also that $\lambda^k(\mathbf{x},\mathbf{y})$ can be written as:

$$\lambda^k(\mathbf{x},\mathbf{y}) = a_k + \mathbb{E}_{\tau\sim\Omega}[\sum_{i=1}^{l_k} \xi_k^i(\mathbf{x},\tau)\eta_k^i(\mathbf{y},\tau)] \tag{15}$$

for some scalars $a_k \in \mathbb{R}$, distribution $\Omega \in \mathcal{P}(\mathbb{R}^d)$, mappings $\xi_k^i, \eta_k^i : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ and that the corresponding estimator $\widehat{\lambda}^k(\mathbf{x},\mathbf{y}) = \widehat{\lambda}_n^k(\mathbf{x},\mathbf{y})$ of $\lambda^k(\mathbf{x},\mathbf{y})$ is of the form:

$$\widehat{\lambda}_n^k(\mathbf{x},\mathbf{y}) = a_k + (\gamma_n^k(\mathbf{x}))^\top \mu_n^k(\mathbf{y}) \tag{16}$$

for $\rho_n^k(\mathbf{u}) = \frac{1}{\sqrt{n}}\rho_n^{1,k}(\mathbf{u}) \star ... \star \rho_n^{l_k,k}(\mathbf{u})$, $\rho_n^{i,k}(\mathbf{u}) = (f_k^i(\mathbf{u},\tau_1), ..., f_k^i(\mathbf{u},\tau_n))^\top$, $\tau_1, ..., \tau_n \sim \Omega$, and where $(\rho, f) \in \{(\gamma,\xi),(\mu,\eta)\}$. It turns out that under these assumptions, $\widehat{\mathrm{SM}}_{m,n}^{\mathrm{hyb}}(\mathbf{x},\mathbf{y})$ can be rewritten as a dot-product of two (hybrid) random feature vectors, thus provides softmax kernel linearization, as the next lemma shows. We denote by $\mathbf{i}$ a complex number such that $\mathbf{i}^2 = -1$.

Lemma 2.4. *The HRF estimator* $\widehat{\mathrm{SM}}_{m,n}^{\mathrm{hyb}}(\mathbf{x},\mathbf{y})$ *satisfies:* $\widehat{\mathrm{SM}}_{m,n}^{\mathrm{hyb}}(\mathbf{x},\mathbf{y}) = \Psi_\alpha^\gamma(\mathbf{x})^\top \Psi_\beta^\mu(\mathbf{y})$*, where* Ψ_ϕ^ρ *for* $(\phi,\rho) \in \{(\alpha,\gamma),(\beta,\mu)\}$ *is given as:* $\Psi_\phi^\rho(\mathbf{z}) = \Psi_\phi^1(\mathbf{z}) \star \Psi_\phi^{\rho,1}(\mathbf{z}) \star \Psi_\phi^2(\mathbf{z}) \star \Psi_\phi^{\rho,2}(\mathbf{z})$ *and:*

$$\begin{aligned}
\Psi_\phi^1(\mathbf{z}) &= \prod_{k=1,...,p}^{\star} \sqrt{\frac{a_k}{m}}\phi_m^{1,k}(\mathbf{z}) \star ... \star \phi_m^{t_k,k}(\mathbf{z}) \\
\Psi_\phi^{\rho,1}(\mathbf{z}) &= \frac{1}{\sqrt{mn}} \prod_{k=1,...,p}^{\star} \prod_{i,j\in\{1,...,l_k\}\times\{1,...,t_k\}}^{\star} \rho_n^{i,k}(\mathbf{z}) \otimes \phi_m^{j,k}(\mathbf{z}) \\
\Psi_\phi^2(\mathbf{z}) &= \sqrt{\frac{1-\sum_{k=1}^p a_k}{m}}\phi_m^{1,p+1}(\mathbf{z}) \star ... \star \phi_m^{t_{p+1},p+1}(\mathbf{z}) \\
\Psi_\phi^{\rho,2}(\mathbf{z}) &= \frac{\mathbf{i}}{\sqrt{mn}} \prod_{k=1,...,p}^{\star} \prod_{i,j\in\{1,...,l_k\}\times\{1,...,t_{p+1}\}}^{\star} \rho_n^{i,k}(\mathbf{z}) \otimes \phi_m^{j,p+1}(\mathbf{z})
\end{aligned} \tag{17}$$

Here $\mathbf{u} \otimes \mathbf{v}$ *is the vectorized outer-product of* $\mathbf{u}$ *and* $\mathbf{v}$ *and* $\prod^\star$ *denotes vectors-concatenation.*

2.3.1 ADAPTING HRF ESTIMATORS: HOW TO INSTANTIATE HYBRID RANDOM FEATURES

We will use complex exponential estimators from Section 2.2 as base estimators in different insantiations of HRF estimators (that by definition admit random feature map decomposition). The key to the linearization of the HRF estimators is then Equation 15 which implies linearization of the shifted versions of λ-coefficients (Equation 16) and consequently - desired random feature map decomposition via the mechanism of compositional kernels (details in the Appendix). The questions remains how to construct λ-coefficients to reduce variance of the estimation in the desired regions of interest.

Accurate approximation of small and large softmax kernel values simultaneously: As we explained earlier, in several applications of softmax estimators a desired property is to provide accurate estimation in two antipodal regions - small and large softmax kernel values. This can be done in particular by choosing: $p = 1$, $\widehat{\mathrm{SM}}^{1} = \widehat{\mathrm{SM}}^{++}$, $\widehat{\mathrm{SM}}^{2} = \widehat{\mathrm{SM}}^{\mathrm{trig}}$ and $\lambda(\mathbf{x}, \mathbf{y}) = \frac{\theta_{\mathbf{x},\mathbf{y}}}{\pi}$, where $\theta_{\mathbf{x},\mathbf{y}}$ stands for an angle between $\mathbf{x}$ and $\mathbf{y}$. The key observation is that $\lambda(\mathbf{x}, \mathbf{y})$ can be unbiasedly approximated via the mechanism of sgn random features (following from Goemans-Willimson algorithm (Goemans & Williamson, 2004)) as follows (details in the Appendix, Sec. D):

$$\widehat{\lambda}(\mathbf{x}, \mathbf{y}) = \frac{1}{2} + \rho_n(\mathbf{x})^\top \rho_n(\mathbf{y}), \tag{18}$$

where: $\rho_n(\mathbf{z}) \stackrel{\mathrm{def}}{=} \frac{\mathbf{i}}{\sqrt{2n}}(\mathrm{sgn}(\tau_1^\top \mathbf{z}), ..., \mathrm{sgn}(\tau_n^\top \mathbf{z}))^\top$ for $\tau_1, ..., \tau_n \sim \mathcal{N}(0, \mathbf{I}_d)$ and $\mathbf{i}^2 = -1$. We call the resulting estimator of the softmax kernel *angular hybrid*. As we show in Sec. 3, this estimator is indeed particularly accurate in approximating both small and large softmax kernel values. For instance, for the prominent case when input vectors are of fixed length, its variance is zero for both: the smallest ($\theta_{\mathbf{x},\mathbf{y}} = \pi$) and largest ($\theta_{\mathbf{x},\mathbf{y}} = 0$) softmax kernel values (see: Fig. 1).

Gaussian Lambda Coefficients: Another tempting option is to instantiate λ-coefficients with approximate Gaussian kernel values (estimated either via positive or trigonometric random features). In that setting, functions $\widehat{\lambda}^k(\mathbf{x}, \mathbf{y})$ are defined as: $\widehat{\lambda}^k(\mathbf{x}, \mathbf{y}) = \exp(-\frac{\|\mathbf{x}+\mathbf{M}_k\mathbf{y}\|_2^2}{2c_k^2})$, for some $\mathbf{M}_1, ..., \mathbf{M}_p \in \mathbb{R}^{d \times d}$ and $c_1, ..., c_p \in \mathbb{R}$. Since this time λ-coefficients are the values of the Gaussian kernel between vectors: $\frac{\mathbf{x}}{c_k}$ and $-\frac{\mathbf{M}_k\mathbf{y}}{c_k}$, we can take $a_k = 0$ and define γ^k,μ^k as:

$$\gamma^k(\mathbf{x}) = \exp(-\frac{\|\mathbf{x}\|_2^2}{2c_k^2})\phi^{\mathrm{SM}}(\frac{\mathbf{x}}{c_k}) \text{ and } \mu^k(\mathbf{y}) = \exp(-\frac{\|\mathbf{M}_k\mathbf{y}\|_2^2}{2c_k^2})\phi^{\mathrm{SM}}(-\frac{\mathbf{M}_k\mathbf{y}}{c_k}), \tag{19}$$

where ϕ^{SM} is the random feature map corresponding to a particular estimator of the softmax kernel. Note that the resulting hybrid estimator, that we call *Gaussian hybrid* has nice pseudo-combinatorial properties. The coefficients $\lambda^k \in (0, 1]$ fire if $\mathbf{x} \approx -\mathbf{M}_k\mathbf{y}$ (the firing window can be accurately controlled via hyperparameters c_k). Thus matrices $\mathbf{M}_k$ can be coupled with the base estimators particularly accurate in the regions, where $\mathbf{x} \approx -\mathbf{M}_k\mathbf{y}$. The mechanism can be thus trivially applied to the setting of accurate approximation of both small and large softmax kernel values for inputs of fixed length, yet it turns out to be characterized by the larger variance than its angular hybrid counterpart and cannot provide variance vanishing property for both: $\theta_{\mathbf{x},\mathbf{y}} = 0$ and $\theta_{\mathbf{x},\mathbf{y}} = \pi$.

Adaptation to data admitting clustering structure: Assume now that the inputs $\mathbf{x}$ come from a distribution $\mathcal{P}(X)$ admitting certain clustering structure with clusters: $X_1, ..., X_u \subseteq \mathbb{R}^d$ of centers: $\mathbf{x}_1, ..., \mathbf{x}_u \in \mathbb{R}^d$ and that the analogous fact is true for $\mathbf{y}$ with the corresponding clusters: $Y_1, ..., Y_w \subseteq \mathbb{R}^d$ and centers: $\mathbf{y}_1, ..., \mathbf{y}_w \in \mathbb{R}^d$. For a fixed pair of clusters' centers: $(\mathbf{x}_i, \mathbf{y}_j)$, one can associate a complex exponential estimator $\widehat{\mathrm{SM}}^{\mathrm{cexp}}_{(\mathbf{x}_i,\mathbf{y}_j)}$ with the corresponding matrix $\mathbf{A} \in \mathbb{C}^{d \times d}$ (see: Sec. 2.2) that minimizes the following loss, where $|\mathbf{z}|$ for $\mathbf{z} \in \mathbb{C}^d$ is defined as $|\mathbf{z}| = \sqrt{\sum_{i=1}^d |z_i|^2}$:

$$l(\mathbf{A}) = |\mathbf{A}\mathbf{x}_i + (\mathbf{A}^\top)^{-1}\mathbf{y}_j|^2 \tag{20}$$

Note that, by Equation 12, if one can find $\mathbf{A}$ such that $l(\mathbf{A}) = 0$, then $\widehat{\mathrm{SM}}^{\mathrm{cexp}}_{(\mathbf{x}_i,\mathbf{y}_j)}$ makes no error in approximating softmax kernel value on the centers: $\mathbf{x}_i, \mathbf{y}_j$. Furthermore, if sampled points are close to the centers and $l(\mathbf{A})$ is small, the corresponding variance of the estimator will also be small. Thus one can think about $\widehat{\mathrm{SM}}^{\mathrm{cexp}}_{(\mathbf{x}_i,\mathbf{y}_j)}$ as an estimator adapted to a pair of clusters $(\mathbf{x}_i, \mathbf{y}_j)$. We can add additional constraints regarding $\mathbf{A}$, i.e. $\mathbf{A} \in \mathbb{R}^{d \times d}$ or $\mathbf{A}$ being diagonal (or both) for closed-form formulae of the optimal $\mathbf{A}$, yet not necessarily zeroing out $l(\mathbf{A})$ (details in the Appendix, Sec. H). Complex exponential estimators defined in this way can be combined together, becoming base estimators in the HRF estimator. The corresponding λ-coefficients can be chosen to be Gaussian as described in the previous paragraph (with optimized matrices $\mathbf{M}_k$ so that a coefficient fires for the corresponding pair of clusters' centers), but there is another much simpler choice of λs. We index different base estimators by pairs of clusters' centers $(\mathbf{x}_i, \mathbf{y}_j)$. One can simply take: $\gamma^{(\mathbf{x}_i,\mathbf{y}_j)}(\mathbf{x}) = 1$ if $\mathbf{x}$ belongs to the cluster of center $\mathbf{x}_i$ and $\gamma^{(\mathbf{x}_i,\mathbf{y}_j)}(\mathbf{x}) = 0$ otherwise. Vectors $\eta^{(\mathbf{x}_i,\mathbf{y}_j)}(\mathbf{y})$ (that also

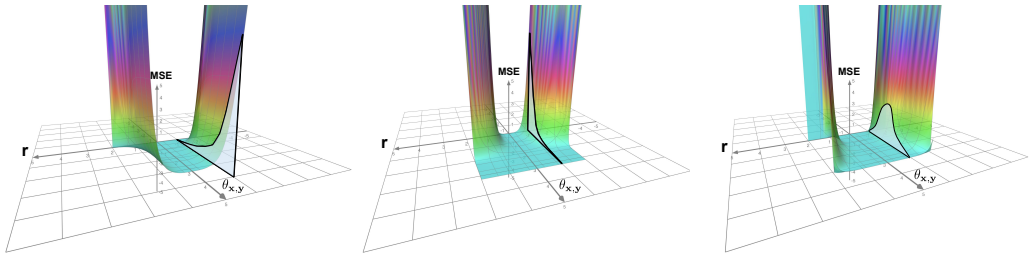

Figure 1: MSEs for three softmax kernel estimators (from left to right): $\widehat{\mathrm{SM}}^{\mathrm{trig}}$, $\widehat{\mathrm{SM}}^{++}$ and angular hybrid for $m = 10, n = 1$ and input of lenghts $r = 5$. MSEs are given as functions of: an angle $\theta_{\mathbf{x},\mathbf{y}} \in [0, \pi]$ between $\mathbf{x}$ and $\mathbf{y}$ and r (symmetrized along 0 for length axis) . For each plot, we marked in grey its slice for a fixed r to illustrate that only for the angular hybrid estimator, the MSE goes to zero for both: $\theta_{\mathbf{x},\mathbf{y}} \to 0$ and $\theta_{\mathbf{x},\mathbf{y}} \to \pi$.

become scalars in this scenario) are defined analogously. Thus effectively the hybrid estimator activates precisely this random feature mechanism that is optimal for the particular pair of clusters. This scheme is feasible if cluster-membership calculation is computationally acceptable, otherwise aforementioned Gaussian coefficients can be applied.

We conclude with an observation that in principle (to reduce the number of base estimators if needed) one can also construct HRF estimators that take base estimators defined in the above way only for the most massive pairs of clusters and add an additional default base estimators (for instance $\widehat{\mathrm{SM}}^{++}$).

3 Theoretical guarantees

We provide here theoretical guarantees regarding HRF estimators. We focus on the bipolar setting with two base estimators: $\widehat{\mathrm{SM}}^{\mathrm{trig}}$ and $\widehat{\mathrm{SM}}^{++}$. The following is true:

Theorem 3.1 (MSE of the bipolar hybrid estimator)**.** *Take the bipolar hybrid estimator $\widehat{\mathrm{SM}}^{\mathrm{hyb}}_{m,n}(\mathbf{x},\mathbf{y})$, where $\widehat{\mathrm{SM}}^{\mathrm{trig}}_{m}(\mathbf{x},\mathbf{y})$ and $\widehat{\mathrm{SM}}^{++}_{m}(\mathbf{x},\mathbf{y})$ are chosen independently i.e. their random projections are chosen independently (note that we always assume that $\widehat{\lambda}_n(\mathbf{x},\mathbf{y})$ is constructed independently from $\widehat{\mathrm{SM}}^{\mathrm{trig}}_{m}(\mathbf{x},\mathbf{y})$ and $\widehat{\mathrm{SM}}^{++}_{m}(\mathbf{x},\mathbf{y})$). Then the following holds:*

$$\mathrm{MSE}(\widehat{\mathrm{SM}}^{\mathrm{hyb}}_{m,n}(\mathbf{x},\mathbf{y})) = \mathbb{E}[\widehat{\lambda}^2_n(\mathbf{x},\mathbf{y})]\mathrm{MSE}(\widehat{\mathrm{SM}}^{++}_{m}(\mathbf{x},\mathbf{y})) + \mathbb{E}[(1-\widehat{\lambda}_n(\mathbf{x},\mathbf{y}))^2]\mathrm{MSE}(\widehat{\mathrm{SM}}^{\mathrm{trig}}_{m}(\mathbf{x},\mathbf{y})) \tag{21}$$

*Furthermore, if $\widehat{\mathrm{SM}}^{\mathrm{trig}}_{m}(\mathbf{x},\mathbf{y})$ and $\widehat{\mathrm{SM}}^{++}_{m}(\mathbf{x},\mathbf{y})$ apply the **exact** same sets of random projections, the mean squared error of the hybrid estimator is further reduced, namely we have:*

$$\begin{aligned}\mathrm{MSE}(\widehat{\mathrm{SM}}^{\mathrm{hyb}}_{m,n}(\mathbf{x},\mathbf{y})) = \mathbb{E}[\widehat{\lambda}^2_n(\mathbf{x},\mathbf{y})]\mathrm{MSE}(\widehat{\mathrm{SM}}^{++}_{m}(\mathbf{x},\mathbf{y})) + \mathbb{E}[(1-\widehat{\lambda}_n(\mathbf{x},\mathbf{y}))^2]\mathrm{MSE}(\widehat{\mathrm{SM}}^{\mathrm{trig}}_{m}(\mathbf{x},\mathbf{y}))\\ -\frac{2}{m}\mathrm{SM}^2(\mathbf{x},\mathbf{y})(1-\cos(\|\mathbf{x}\|_2^2-\|\mathbf{y}\|_2^2))\mathbb{E}[\widehat{\lambda}_n(\mathbf{x},\mathbf{y})(1-\widehat{\lambda}_n(\mathbf{x},\mathbf{y}))]\end{aligned} \tag{22}$$

The exact formula on $\mathrm{MSE}(\widehat{\mathrm{SM}}^{++}_{m}(\mathbf{x},\mathbf{y}))$ and $\mathrm{MSE}(\widehat{\mathrm{SM}}^{\mathrm{trig}}_{m}(\mathbf{x},\mathbf{y}))$ is given in Lemma 2.2.

We can apply this result directly to the angular hybrid estimator, since:

Lemma 3.2. *For the angular hybrid estimator the following holds:*

$$\mathbb{E}[\widehat{\lambda}^2_n(\mathbf{x},\mathbf{y})] = \frac{\theta_{\mathbf{x},\mathbf{y}}}{\pi}\left(\frac{\theta_{\mathbf{x},\mathbf{y}}}{\pi} - \frac{\theta_{\mathbf{x},\mathbf{y}}}{n\pi} + \frac{1}{n}\right), \mathbb{E}[\widehat{\lambda}_n(\mathbf{x},\mathbf{y})] = \frac{\theta_{\mathbf{x},\mathbf{y}}}{\pi}. \tag{23}$$

We see that, as mentioned before, the variance of the angular hybrid estimator is zero for both: $\theta_{\mathbf{x},\mathbf{y}} = 0$ and $\theta_{\mathbf{x},\mathbf{y}} = \pi$ if inputs $\mathbf{x}, \mathbf{y}$ have the same length. We need one more definition.

Definition 3.3. *Assume that the inputs to the estimators are taken from some given bounded set $\mathcal{C} \subseteq \mathbb{R}^d$. For a given estimator $\widehat{\mathrm{SM}}$ on feature vectors $\mathbf{x}, \mathbf{y} \in \mathcal{C}$, we define its max-relative-error with respect to $\mathcal{C}$ as: $\epsilon_{\mathcal{C}}(\widehat{\mathrm{SM}}) = \max_{\mathbf{x},\mathbf{y}\in\mathcal{C}} \epsilon_{\mathbf{x},\mathbf{y}}(\widehat{\mathrm{SM}})$, where $\epsilon_{\mathbf{x},\mathbf{y}}(\widehat{\mathrm{SM}}) = \frac{\sqrt{\mathrm{MSE}(\widehat{\mathrm{SM}}(\mathbf{x},\mathbf{y}))}}{\mathrm{SM}(\mathbf{x},\mathbf{y})}$.*

This definition captures the critical observation that in several applications of the softmax kernel estimation, e.g. efficient softmax sampling or linear-attention Transformers (Choromanski et al., 2021b), small relative errors are a much more meaningful measure of the quality of the method than small absolute errors. It also enables us to find hidden symmetries between different estimators:

Lemma 3.4. *Denote by $S(r)$ a sphere centered at 0 and of radius r. Define $\epsilon_{\theta,r}(\widehat{\mathrm{SM}}) \stackrel{\mathrm{def}}{=} \epsilon_{\mathbf{x},\mathbf{y}}(\widehat{\mathrm{SM}})$ for $\mathbf{x},\mathbf{y} \in S(r)$ and such that $\theta = \theta_{\mathbf{x},\mathbf{y}}$ (note that the mean squared errors of the considered estimators depend only on the angle $\theta_{\mathbf{x},\mathbf{y}}$ for $\mathbf{x},\mathbf{y}$ chosen from a fixed sphere). Then the following holds:*

$$\epsilon_{\theta,r}(\widehat{\mathrm{SM}}_m^{\mathrm{trig}}) = \epsilon_{\pi-\theta,r}(\widehat{\mathrm{SM}}_m^{++}) = \frac{1}{\sqrt{2m}} \exp(2r^2 \sin^2(\frac{\theta}{2})) \left(1 - \exp(-4r^2 \sin^2(\frac{\theta}{2}))\right) \tag{24}$$

and consequently for $W(r) = \exp(2r^2)\left(1 - \exp(-4r^2)\right)$:

$$\epsilon_{S(r)}(\widehat{\mathrm{SM}}_m^{\mathrm{trig}}) = \epsilon_{S(r)}(\widehat{\mathrm{SM}}_m^{++}) = \lim_{\theta\to\pi} \epsilon_{\theta,r}(\widehat{\mathrm{SM}}_m^{\mathrm{trig}}) = \lim_{\theta\to 0} \epsilon_{\theta,r}(\widehat{\mathrm{SM}}_m^{++}) = \sqrt{\frac{1}{2m}} W(r) \tag{25}$$

Our main result shows that HRF estimators can be applied to reduce the max-relative-error of the previously applied estimators of the softmax kernel. In the next theorem we show that the max-relative-error of the angular hybrid estimator scales as: $\frac{1}{r}\exp(2r^2)$ in the length r of its inputs as opposed to $\exp(2r^2)$ as it is the case for $\widehat{\mathrm{SM}}^{\mathrm{trig}}$ and $\widehat{\mathrm{SM}}^{++}$ (see: Lemma 3.4). Furthermore, the max-relative-error scales as $\sqrt{\theta}$ and $\sqrt{\pi - \theta}$ as $\theta \to 0$ and $\theta \to \pi$ respectively, in particular goes to 0 in both critical cases. This is not true for $\widehat{\mathrm{SM}}^{\mathrm{trig}}$ nor for $\widehat{\mathrm{SM}}^{++}$.

Theorem 3.5. *The max-relative-error of the angular hybrid estimator for the inputs $\mathbf{x},\mathbf{y}$ on the sphere $S(r)$ of radius $r \geq 1$ satisfies for $W(r) = \exp(2r^2)\left(1 - \exp(-4r^2)\right)$:*

$$\epsilon_{S(r)}(\widehat{\mathrm{SM}}_{m,n}^{\mathrm{anghyb}}) \leq \frac{1}{r}\sqrt{\frac{1}{2m}} W(r) \sqrt{\frac{1}{\pi} - \frac{1}{n\pi} + \frac{1}{n\sqrt{\pi}}} \tag{26}$$

Furthermore: $\lim_{\theta\to 0} \frac{\epsilon_{\theta,r}(\widehat{\mathrm{SM}}_{m,n}^{\mathrm{anghyb}})}{\sqrt{\theta}} = \lim_{\theta\to\pi} \frac{\epsilon_{\theta,r}(\widehat{\mathrm{SM}}_{m,n}^{\mathrm{anghyb}})}{\sqrt{\theta-\pi}} = \sqrt{\frac{1}{2\pi mn}} W(r)$.

Additional implications of Theorem 3.5, using the fact that $\Theta(mn)$-dimensional HRFs can be constructed in time $O(nd + md + mn)$ (regular RFs need $O(mnd)$), are in the Appendix (Sec. D.4).

4 Experiments

In this section we conduct exhaustive evaluation of the mechanism of HRFs on several tasks.

4.1 Pointwise Softmax Kernel Estimation

We start with ablation studies regarding empirical relative errors of different softmax kernel estimators across different inputs' lengths r and angles $\theta_{\mathbf{x},\mathbf{y}}$. The results are presented in Fig. 2. Complete ablations are presented in the Appendix (Sec. G). The angular hybrid estimator most accurately approximates softmax kernel and has smaller max-relative-error than other estimators.

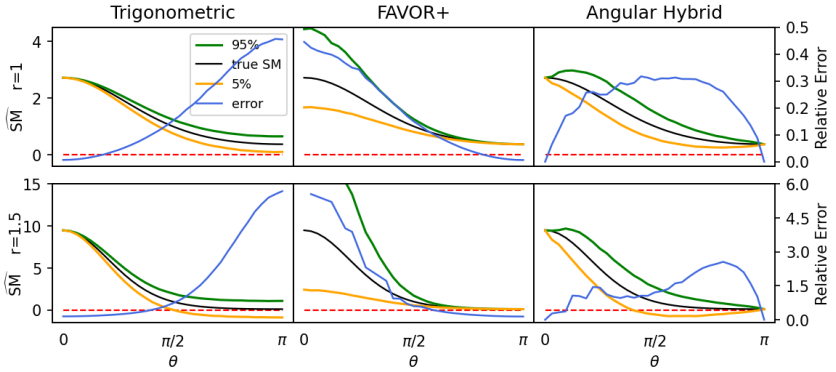

Figure 2: Pointwise estimation of $\mathrm{SM}(\mathbf{x},\mathbf{y})$ for the same-length 64-dim inputs ($r = 1.0$ and $r = 1.5$) and various angles $\theta_{\mathbf{x},\mathbf{y}}$. We used $s = 10000$ estimated softmax values in each subplot. The true softmax value and the 5th and 95th quantile estimated values are shown by the left y-axis, and the empirical relative errors are shown by the right y-axis. Trigonometric estimator and FAVOR+ applied 128 random features. To make fair comparison, for the hybrid variant the configuration leading to the similar number of FLOPS operations per random feature map creation was applied.

4.2 DATA ADMITTING CLUSTERING STRUCTURE

Next we test HRF estimators customized to data with clustering structure, as described in Sec. 2.3.1. Inputs $\mathbf{x}$ are taken from two 50-dimensional 1000-element Gaussian clusters and the inputs $\mathbf{y}$ from two other 50-dimensional 1000-element Gaussian clusters. We use empirical MSE metric and constrain matrix $\mathbf{A}$ to be real diagonal for a compact closed-form formulae (see: Appendix, Sec. H). We created four different clusters' configurations corresponding to different values of $s = l(\mathbf{A})$ (see: Eq. 20) for all pairs of clusters and with different concentrations around centers controlled by the standard deviation σ (small/large values of s and σ). As we show in Fig. 3, for all variants HRF estimators adapted to clustered data consistently provide notable accuracy gains, even for larger values of s and less concentrated clusters. Additional experimental details are given in Appendix H.

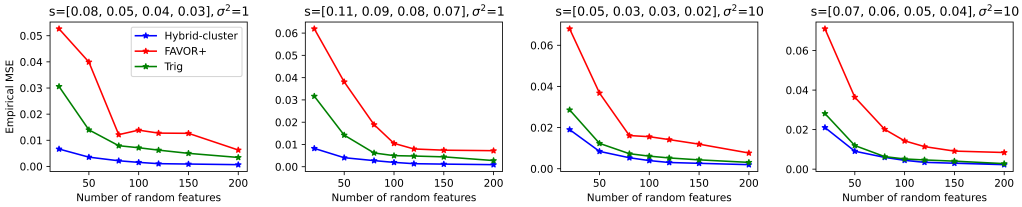

Figure 3: Comparing different estimators for data with clustering structure. The empirical MSE is obtained by averaging over 20 trials. For the HRFs, we apply deterministic λ-coefficients. The fact that the empirical error for FAVOR+ is not perfectly monotonic in m was first observed in (Luo et al., 2021) (see: Fig. 1: (b)).

4.3 LANGUAGE MODELING

In this section, we apply HRFs to the language modeling task, training a 2-layer LSTM with hidden size $h = 200$ and applying RFs for softmax sampling in training as described in (Rawat et al., 2019). Additional experimental details are given in the Appendix (Sec. I).

Statistical Metrics. We compute $1d$-Wasserstein distance and the Kolmogorov-Smirnov (KS) metric (Ramdas et al., 2017) between a true softmax distribution induced by a query on all the classes and the approximate one given by different RF-mechanisms. We also compute the fraction of all probability estimates with undesired negative values. The results on the PennTree Bank Marcus et al. (1993) are presented in Fig. 4. In the Appendix (Sec. I) we present additional results for the $\mathrm{WikiText2}$ dataset. We see that HRFs lead to most accurate softmax distribution estimation.

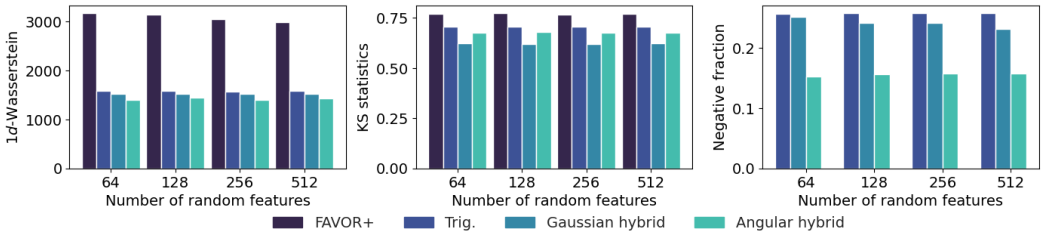

Figure 4: Statistical metrics measuring softmax matrix approximation quality on PennTree Bank dataset. For FAVOR+ and the trigonometric mechanism, the number of random features are $64, 128, 256, 512$. To make fair comparison, for the hybrid variants, the configurations leading to the similar number of FLOPS operations per random feature map creation were applied.

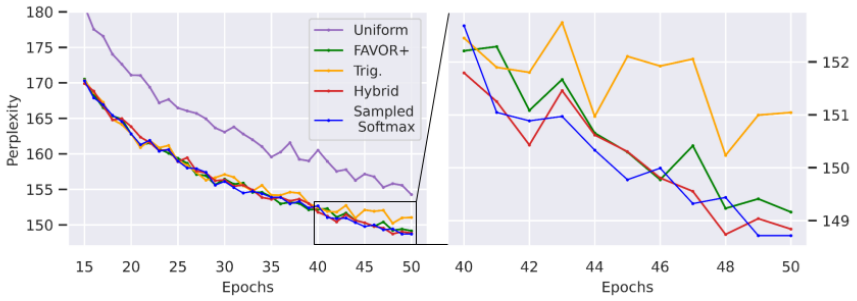

Figure 5: Language Modeling with softmax sampling: training results on PennTree Bank dataset (40-60 epoch window zoomed in on the right subfigure). FAVOR+/trigonometric mechanisms used 1024 random features. To make fair comparison, for HRFs, the configurations leading to the similar number of FLOPS operations per random feature map creation were applied. HRFs turn out to be the most accurate random feature method.

Validation Results. We sampled $m = 40$ negative classes to approximate the full softmax loss during training, and calculated the validation loss using full softmax (see: Fig. 5). The uniform method sampled negative classes uniformly at random, the sampled softmax method sampled classes with the full softmax distribution as the sampling distribution. HRFs and positive random features outperform SOTA RF-mechanism based on trigonometric random features (Rawat et al., 2019).

4.4 Training Speech Models with HRF-Conformers-Performers

We also tested HRFs on speech models with LibriSpeech ASR corpus (Panayotov et al., 2015). We put implicit Performer's attention into 17-layer Conformer-Transducer encoder (Gulati et al., 2020) and compared softmax kernel estimators in terms of word error rate (WER) metric, commonly used to evaluate speech models. The results are presented in Table 1. HRFs produce smallest WER models. Additional details regarding speech experiments are given in the Appendix (Sec. J).

Table 1: Comparison of WERs of Conformer-Transducer applying different RF-mechanisms for the implicit attention. Numbers next to method names define the values of m or (m, n). We note that differencies $\geq 0.2\%$ are considered to be statistically significant for that task.

	HRF(16,8)	HRF(8, 8)	FAVOR+ 432	FAVOR+ 256	Trig 432	Trig 256
WER	2.03%	2.05%	2.65%	2.77%	3.12%	3.3%

4.5 Downstream Robotics Experiments

In Robotics, we leverage the accuracy of HRFs to obtain inference improvements, critical for on-robot deployment. To abstract from a particular hardware characteristic, we measure it in the number of FLOPS. We conduct two experiments targeting: (a) quadruped locomotion and (b) robotic-arm manipulation. In the former, HRFs are applied as a replacement of the default mechanism using positive random features in the class of implicit-attention architectures for vision processing called *Implicit Attention Policies* (or: IAPs) (Choromanski et al., 2021a). In the latter, HRFs become a part of the regular Vision-Performer stack used to process high-resolution (500 x 500 x 3) input. The results are presented in Fig. 6. We conclude that HRFs provide drastic compute improvements with no loss of quality. All details regarding both experimental setups are given in the Appendix (Sec. K). Videos of the HRF-trained robotic policies are included in the supplementary material.

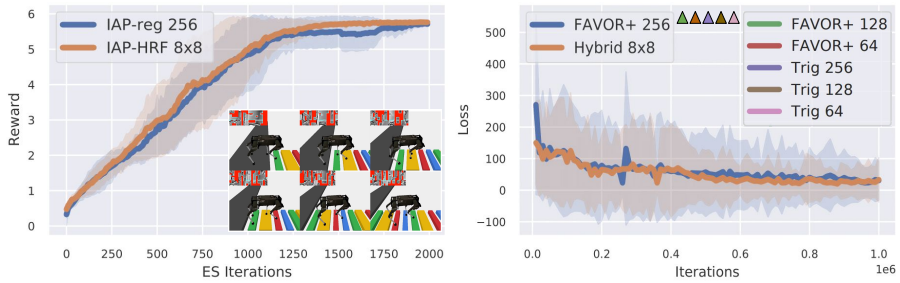

Figure 6: **Left**: Step-stone locomotion task. Comparison of the training curves for: the IAP using angular hybrid estimator of $m = n = 8$ and IAP applying regular FAVOR+ mechanism from (Choromanski et al., 2021b) with $m = 256$. Both final policies are of similar quality, yet HRF-method requires **3x+** fewer FLOPS to run its trained policy. The visualization of the HRF policy in action and its attention (with filtered out pixels marked in red) is in the bottom right corner. **Right:** Similar setting (and conclusions) but for the robotic-arm manipulation task. The additional five regular RF-configurations did not train by producing Nan loss.

5 Conclusion

We presented a new class of random feature techniques, called *hybrid random features* for softmax/Gaussian kernel estimation that more accurately approximate these kernels than previous algorithms. We also demonstrated their robustness in a wide range of applications from softmax sampling to Transformers/attention training, also for downstream Robotics tasks.

Reproducibility Statement: Section 4 and the Appendix include all experimental details (e.g. hyperparameter setup) needed to reproduce all the results presented in the paper. The part of the code that we could make publicly available can be found in the following anonymous github: `https://anonymous.4open.science/r/HRF_ICLR2022-B69C/`.

Ethics Statement: HRFs can be used in principle to train massive Transformer models with large number of parameters and proportional compute resources. Thus they should be used responsibly, in particular given CO_2 emission challenges that scientific community tries to address now.

REFERENCES

Unitree Robotics. URL http://www.unitree.cc/.

Guy Blanc and Steffen Rendle. Adaptive sampled softmax with kernel based sampling. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 589–598. PMLR, 2018. URL http://proceedings.mlr.press/v80/blanc18a.html.

Y. Cho and L. K. Saul. Analysis and extension of arc-cosine kernels for large margin classification. Technical Report CS2012-0972, Department of Computer Science and Engineering, University of California, San Diego., 2012.

Krzysztof Choromanski and Vikas Sindhwani. Recycling randomness with structure for sublinear time kernel expansions. In Maria-Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 2502–2510. JMLR.org, 2016. URL http://proceedings.mlr.press/v48/choromanski16.html.

Krzysztof Choromanski, Mark Rowland, Wenyu Chen, and Adrian Weller. Unifying orthogonal Monte Carlo methods. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1203–1212. PMLR, 2019. URL http://proceedings.mlr.press/v97/choromanski19a.html.

Krzysztof Choromanski, Deepali Jain, Jack Parker-Holder, Xingyou Song, Valerii Likhosherstov, Anirban Santara, Aldo Pacchiano, Yunhao Tang, and Adrian Weller. Unlocking pixels for reinforcement learning via implicit attention. *CoRR*, abs/2102.04353, 2021a. URL https://arxiv.org/abs/2102.04353.

Krzysztof Marcin Choromanski, Mark Rowland, and Adrian Weller. The unreasonable effectiveness of structured random orthogonal embeddings. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pp. 219–228, 2017.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamás Sarlós, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J. Colwell, and Adrian Weller. Rethinking attention with performers. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021b. URL https://openreview.net/forum?id=Ua6zuk0WRH.

Amit Daniely, Roy Frostig, Vineet Gupta, and Yoram Singer. Random features for compositional kernels. *CoRR*, abs/1703.07872, 2017. URL http://arxiv.org/abs/1703.07872.

Michel X. Goemans and David P. Williamson. Approximation algorithms for MAX-3-CUT and other problems via complex semidefinite programming. *J. Comput. Syst. Sci.*, 68(2):442–470, 2004. doi: 10.1016/j.jcss.2003.07.012. URL https://doi.org/10.1016/j.jcss.2003.07.012.

Arthur Gretton, Ralf Herbrich, Alexander J. Smola, Olivier Bousquet, and Bernhard Schölkopf. Kernel methods for measuring independence. *J. Mach. Learn. Res.*, 6:2075–2129, 2005. URL http://jmlr.org/papers/v6/gretton05a.html.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. Conformer: Convolution-augmented transformer for speech recognition. In Helen Meng, Bo Xu, and Thomas Fang Zheng (eds.), *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pp. 5036–5040. ISCA, 2020. doi: 10.21437/Interspeech.2020-3015. URL https://doi.org/10.21437/Interspeech.2020-3015.

Hakan Inan, Khashayar Khosravi, and Richard Socher. Tying word vectors and word classifiers: A loss framework for language modeling. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

Shengjie Luo, Shanda Li, Tianle Cai, Di He, Dinglan Peng, Shuxin Zheng, Guolin Ke, Liwei Wang, and Tie-Yan Liu. Stable, fast and accurate: Kernelized attention with relative positional encoding, 2021.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993. URL https://aclanthology.org/J93-2004.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer Sentinel Mixture Models. *5th International Conference on Learning Representations*, abs/1609.07843, 2017.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, pp. 5206–5210. IEEE, 2015. doi: 10.1109/ICASSP.2015.7178964. URL https://doi.org/10.1109/ICASSP.2015.7178964.

Ofir Press and Lior Wolf. Using the Output Embedding to Improve Language Models, 2017.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In John C. Platt, Daphne Koller, Yoram Singer, and Sam T. Roweis (eds.), *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pp. 1177–1184. Curran Associates, Inc., 2007. URL http://papers.nips.cc/paper/3182-random-features-for-large-scale-kernel-machines.

Aaditya Ramdas, Nicolás García Trillos, and Marco Cuturi. On Wasserstein two-sample testing and related families of nonparametric tests. *Entropy*, 19(2):47, 2017. doi: 10.3390/e19020047. URL https://doi.org/10.3390/e19020047.

Ankit Singh Rawat, Jiecao Chen, Felix X. Yu, Ananda Theertha Suresh, and Sanjiv Kumar. Sampled softmax with random Fourier features. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 13834–13844, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/e43739bba7cdb577e9e3e4e42447f5a5-Abstract.html.

Mark Rowland, Krzysztof Choromanski, François Chalus, Aldo Pacchiano, Tamás Sarlós, Richard E. Turner, and Adrian Weller. Geometrically coupled Monte Carlo sampling. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 195–205, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/b3e3e393c77e35a4a3f3cbd1e429b5dc-Abstract.html.

Alessandro Rudi, Raffaello Camoriano, and Lorenzo Rosasco. Less is more: Nyström computational regularization. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama,

and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp. 1657–1665, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/03e0704b5690a2dee1861dc3ad3316c9-Abstract.html.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9355–9366. PMLR, 2021. URL http://proceedings.mlr.press/v139/schlag21a.html.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018. URL http://arxiv.org/abs/1807.03748.

Christopher K. I. Williams and Matthias W. Seeger. Using the Nyström method to speed up kernel machines. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp (eds.), *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*, pp. 682–688. MIT Press, 2000. URL https://proceedings.neurips.cc/paper/2000/hash/19de10adbaa1b2ee13f77f679fa1483a-Abstract.html.

Tianbao Yang, Yu-Feng Li, Mehrdad Mahdavi, Rong Jin, and Zhi-Hua Zhou. Nyström method vs random Fourier features: A theoretical and empirical comparison. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pp. 485–493, 2012. URL https://proceedings.neurips.cc/paper/2012/hash/621bf66ddb7c962aa0d22ac97d69b793-Abstract.html.

Felix X. Yu, Ananda Theertha Suresh, Krzysztof Marcin Choromanski, Daniel N. Holtmann-Rice, and Sanjiv Kumar. Orthogonal random features. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 1975–1983, 2016. URL http://papers.nips.cc/paper/6246-orthogonal-random-features.

Qinyi Zhang, Sarah Filippi, Arthur Gretton, and Dino Sejdinovic. Large-scale kernel methods for independence testing. *Stat. Comput.*, 28(1):113–130, 2018. doi: 10.1007/s11222-016-9721-7. URL https://doi.org/10.1007/s11222-016-9721-7.

APPENDIX: HYBRID RANDOM FEATURES

A PROOF OF LEMMA 2.3

Proof. From the independence of $\omega_1, ..., \omega_d$, we get:

$$\mathbb{E}[\exp(\omega^\top \mathbf{z})] = \prod_{i=1}^{d} \mathbb{E}[\exp(\omega_i z_i)] \tag{27}$$

Thus it suffices to show that for $g \sim \mathcal{N}(0,1)$ and any $z \in \mathbb{C}$ the following holds:

$$\mathbb{E}[\exp(gz)] = \exp(\frac{z^2}{2}) \tag{28}$$

Define $f(z) = \mathbb{E}[\exp(gz)]$. Note that $f(z) = \exp(\frac{z^2}{2})$ for $z = \mathbf{i}x$, where $\mathbf{i}^2 = -1$ and $x \in \mathbb{R}$ which follows from the formula of the characteristic function of the Gaussian distribution. Similarly, $f(z) = \exp(\frac{z^2}{2})$ for $z \in \mathbb{R}$ which follows from the formula of the moment generating function of the Gaussian distribution. We now use the fact from complex analysis that if two analytic functions $\mathbb{C} \rightarrow \mathbb{C}$ are identical on uncountably many points then they are equal on $\mathbb{C}$ to complete the proof (we leave to the reader checking that both $f(z)$ and $g(z) \stackrel{\text{def}}{=} \exp(\frac{z^2}{2})$ are analytic). □

B PROOF OF LEMMA 2.4

Proof. The following is true:

$$\begin{aligned}
\widehat{\mathrm{SM}}^{\mathrm{hyb}}_{m,n}(\mathbf{x},\mathbf{y}) = \sum_{k=1}^{p} \frac{a_k}{m} (\alpha_m^{1,k}(\mathbf{x}) \star ... \star \alpha_m^{t_k,k}(\mathbf{x}))^\top (\beta_m^{1,k}(\mathbf{y}) \star ... \star \beta_m^{t_k,k}(\mathbf{y})) + \\
\frac{1}{mn} \sum_{k=1}^{p} \sum_{i=1}^{l_k} \sum_{j=1}^{t_k} (\gamma_n^{i,k}(\mathbf{x}) \otimes \alpha_m^{j,k}(\mathbf{x}))^\top (\mu_n^{i,k}(\mathbf{y}) \otimes \beta_m^{j,k}(\mathbf{y})) + \\
\frac{1 - \sum_{k=1}^{p} a_k}{m} (\alpha_m^{1,p+1}(\mathbf{x}) \star ... \star \alpha_m^{t_{p+1},p+1}(\mathbf{x}))^\top (\beta_m^{1,p+1}(\mathbf{y}) \star ... \star \beta_m^{t_{p+1},p+1}(\mathbf{y})) - \\
\frac{1}{mn} \sum_{k=1}^{p} \sum_{i=1}^{l_k} \sum_{j=1}^{t_{p+1}} (\gamma_n^{i,k}(\mathbf{x}) \otimes \alpha_m^{j,p+1}(\mathbf{x}))^\top (\mu_n^{i,k}(\mathbf{y}) \otimes \beta_m^{j,p+1}(\mathbf{y})).
\end{aligned} \tag{29}$$

The statement of the lemma follows directly from that. Note that the key trick is an observation that a product of two functions of $\mathbf{x}, \mathbf{y}$ that can be linearized on expectation (i.e. rewritten with $\mathbf{x}$ and $\mathbf{y}$ disentangled) is itself a function that can be linearized on expectation. In the setting where the former are approximated by random features, the latter is approximated by their cartesian product (the random feature mechanism for the compositional kernels from (Daniely et al., 2017)). □

C BIPOLAR HRF ESTIMATORS: PROOF OF THEOREM 3.1

Proof. The following holds:

$$\begin{aligned}
\mathrm{Var}(\widehat{\mathrm{SM}}^{\mathrm{hyb}}_{m,n}(\mathbf{x},\mathbf{y})) = \mathrm{Var}\left(\widehat{\lambda}_n(\theta)\widehat{\mathrm{SM}}^{++}_m(\mathbf{x},\mathbf{y})\right) + \mathrm{Var}\left((1-\widehat{\lambda}_n(\theta))\widehat{\mathrm{SM}}^{\mathrm{trig}}_m(\mathbf{x},\mathbf{y})\right) + \\
2\mathrm{Cov}\left(\widehat{\lambda}_n(\theta)\widehat{\mathrm{SM}}^{++}_m(\mathbf{x},\mathbf{y}), (1-\widehat{\lambda}_n(\theta))\widehat{\mathrm{SM}}^{\mathrm{trig}}_m(\mathbf{x},\mathbf{y})\right)
\end{aligned} \tag{30}$$

We will focus now on the covariance term. To simplify the notation, we will drop index n in $\widehat{\lambda}_n$. Assume first easier to analyze case where $\widehat{\mathrm{SM}}^{\mathrm{trig}}_m(\mathbf{x},\mathbf{y})$ and $\widehat{\mathrm{SM}}^{++}_m(\mathbf{x},\mathbf{y})$ are chosen independently. Then the following is true:

$$\mathrm{Cov}\left(\widehat{\lambda}(\theta)\widehat{\mathrm{SM}}_m^{++}(\mathbf{x},\mathbf{y}),(1-\widehat{\lambda}(\theta))\widehat{\mathrm{SM}}_m^{\mathrm{trig}}(\mathbf{x},\mathbf{y})\right) =$$
$$\mathbb{E}[\widehat{\lambda}(\theta)(1-\widehat{\lambda}(\theta))\widehat{\mathrm{SM}}_m^{++}(\mathbf{x},\mathbf{y})\widehat{\mathrm{SM}}_m^{\mathrm{trig}}(\mathbf{x},\mathbf{y})] - \mathbb{E}[\widehat{\lambda}(\theta)\widehat{\mathrm{SM}}_m^{++}(\mathbf{x},\mathbf{y})]\mathbb{E}[(1-\widehat{\lambda}(\theta))\widehat{\mathrm{SM}}_m^{\mathrm{trig}}(\mathbf{x},\mathbf{y})]$$
$$= \left(\mathbb{E}[\widehat{\lambda}(\theta)(1-\widehat{\lambda}(\theta))] - \mathbb{E}[\widehat{\lambda}(\theta)]\mathbb{E}[(1-\widehat{\lambda}(\theta))]\right)\mathbb{E}[\widehat{\mathrm{SM}}_m^{++}(\mathbf{x},\mathbf{y})]\mathbb{E}[\widehat{\mathrm{SM}}_m^{\mathrm{trig}}(\mathbf{x},\mathbf{y})]$$
$$= -(\mathrm{SM}(\mathbf{x},\mathbf{y}))^2\mathrm{Var}(\widehat{\lambda}(\theta)) \tag{31}$$

Now assume that $\widehat{\mathrm{SM}}_m^{\mathrm{trig}}(\mathbf{x},\mathbf{y})$ and $\widehat{\mathrm{SM}}_m^{++}(\mathbf{x},\mathbf{y})$ use the exact same random projections. Then, using similar analysis as before, we get:

$$\mathrm{Cov}\left(\widehat{\lambda}(\theta)\widehat{\mathrm{SM}}_m^{++}(\mathbf{x},\mathbf{y}),(1-\widehat{\lambda}(\theta))\widehat{\mathrm{SM}}_m^{\mathrm{trig}}(\mathbf{x},\mathbf{y})\right) =$$
$$\mathbb{E}[\widehat{\mathrm{SM}}_m^{++}(\mathbf{x},\mathbf{y})\widehat{\mathrm{SM}}_m^{\mathrm{trig}}(\mathbf{x},\mathbf{y})]\mathbb{E}[\widehat{\lambda}(\theta)(1-\widehat{\lambda}(\theta))] - (\mathrm{SM}(\mathbf{x},\mathbf{y}))^2\mathbb{E}[\widehat{\lambda}(\theta)]\mathbb{E}[(1-\widehat{\lambda}(\theta))] \tag{32}$$

This time however it is no longer the case that $\mathbb{E}[\widehat{\mathrm{SM}}_m^{++}(\mathbf{x},\mathbf{y})\widehat{\mathrm{SM}}_m^{\mathrm{trig}}(\mathbf{x},\mathbf{y})]\mathbb{E}[\widehat{\lambda}(\theta)(1-\widehat{\lambda}(\theta))] = \mathbb{E}[\widehat{\mathrm{SM}}_m^{++}(\mathbf{x},\mathbf{y})]\mathbb{E}[\widehat{\mathrm{SM}}_m^{\mathrm{trig}}(\mathbf{x},\mathbf{y})] = (\mathrm{SM}(\mathbf{x},\mathbf{y}))^2$ since $\widehat{\mathrm{SM}}_m^{++}(\mathbf{x},\mathbf{y})$ and $\widehat{\mathrm{SM}}_m^{\mathrm{trig}}(\mathbf{x},\mathbf{y})$ are no longer independent. In order to compute $\mathbb{E}[\widehat{\mathrm{SM}}_m^{++}(\mathbf{x},\mathbf{y})\widehat{\mathrm{SM}}_m^{\mathrm{trig}}(\mathbf{x},\mathbf{y})]\mathbb{E}[\widehat{\lambda}(\theta)(1-\widehat{\lambda}(\theta))]$, we will first introduce useful denotation.

Denote by $\omega_1,...,\omega_m \overset{\mathrm{iid}}{\sim} \mathcal{N}(0,\mathbf{I}_d)$ the random projections sampled to construct both $\widehat{\mathrm{SM}}_m^{++}(\mathbf{x},\mathbf{y})$ and $\widehat{\mathrm{SM}}_m^{\mathrm{trig}}(\mathbf{x},\mathbf{y})$. Denote: $Y_i = \cosh(\omega_i^\top(\mathbf{x}+\mathbf{y})) \overset{\mathrm{def}}{=} \frac{\exp(\omega_i^\top(\mathbf{x}+\mathbf{y}))+\exp(-\omega_i^\top(\mathbf{x}+\mathbf{y}))}{2}$. We have:

$$\widehat{\mathrm{SM}}_m^{++}(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}\|^2+\|\mathbf{y}\|^2}{2}\right)\frac{Y_1+...+Y_m}{m} \tag{33}$$

If we denote: $Z_i = \cos(\omega_i^\top(\mathbf{x}-\mathbf{y}))$, then we can write $\widehat{\mathrm{SM}}_m^{\mathrm{trig}}(\mathbf{x},\mathbf{y})$ as:

$$\widehat{\mathrm{SM}}_m^{\mathrm{trig}}(\mathbf{x},\mathbf{y}) = \exp\left(\frac{\|\mathbf{x}\|^2+\|\mathbf{y}\|^2}{2}\right)\frac{Z_1+...+Z_m}{m} \tag{34}$$

We can then rewrite $\mathbb{E}[\widehat{\mathrm{SM}}_m^{++}(\mathbf{x},\mathbf{y})\widehat{\mathrm{SM}}_m^{\mathrm{trig}}(\mathbf{x},\mathbf{y})]$ as:

$$\mathbb{E}[\widehat{\mathrm{SM}}_m^{++}(\mathbf{x},\mathbf{y})\widehat{\mathrm{SM}}_m^{\mathrm{trig}}(\mathbf{x},\mathbf{y})] = \frac{1}{m^2}\left[\sum_{i\neq j}\mathbb{E}[Y_iZ_j] + \sum_{i=1}^{m}\mathbb{E}[Y_iZ_i]\right] =$$
$$\frac{1}{m^2}\left[\binom{m}{2}(\mathrm{SM}(\mathbf{x},\mathbf{y}))^2 + m\mathbb{E}[\cosh(\omega^\top(\mathbf{x}+\mathbf{y}))\cos(\omega^\top(\mathbf{x}-\mathbf{y}))]\right], \tag{35}$$

where $\omega \sim \mathcal{N}(0,\mathbf{I}_d)$. The equality follows from the unbiasedness of $\widehat{\mathrm{SM}}_m^{\mathrm{trig}}(\mathbf{x},\mathbf{y})$ and $\widehat{\mathrm{SM}}_m^{++}(\mathbf{x},\mathbf{y})$ and the fact that different ω_i are chosen independently. Thus it remains to compute $\rho = \mathbb{E}[\cosh(\omega^\top(\mathbf{x}+\mathbf{y}))\cos(\omega^\top(\mathbf{x}-\mathbf{y}))]$. Note first that $\rho = \mathbb{E}[\exp(\omega^\top(\mathbf{x}+\mathbf{y}))\cos(\omega^\top(\mathbf{x}-\mathbf{y}))]$ since $-\omega \sim \mathcal{N}(0,\mathbf{I}_d)$ and cos is an even function. Denote $\mathbf{z} = \mathbf{x}+\mathbf{y}+\mathbf{i}(\mathbf{x}-\mathbf{y})$. We have:

$$\mathbb{E}[\exp(\omega^\top(\mathbf{x}+\mathbf{y}))\cos(\omega^\top(\mathbf{x}-\mathbf{y}))] = \mathrm{Re}\left[\mathbb{E}[\exp(\omega^\top\mathbf{z})]\right] = \mathrm{Re}[\prod_{i=1}^{d}\exp(\frac{z_i^2}{2})] =$$
$$\mathrm{Re}\left[\exp\left(\frac{\sum_{j=1}^{d}(x_j+y_j)^2 + 2\mathbf{i}(x_j^2-y_j^2) - (x_j-y_j)^2}{2}\right)\right] = (\mathrm{SM}(\mathbf{x},\mathbf{y}))^2\cos(\|\mathbf{x}\|_2^2 - \|\mathbf{y}\|_2^2) \tag{36}$$

Thus we conclude that:

$$\mathbb{E}[\widehat{\mathrm{SM}}_m^{++}(\mathbf{x},\mathbf{y})\widehat{\mathrm{SM}}_m^{\mathrm{trig}}(\mathbf{x},\mathbf{y})] = (1-\frac{1}{m})(\mathrm{SM}(\mathbf{x},\mathbf{y}))^2 + \frac{1}{m}(\mathrm{SM}(\mathbf{x},\mathbf{y}))^2\cos(\|\mathbf{x}\|_2^2 - \|\mathbf{y}\|_2^2) \tag{37}$$

Therefore we get the formulae for the covariance term in both: the setting where random projections of $\widehat{\mathrm{SM}}_m^{\mathrm{trig}}(\mathbf{x},\mathbf{y})$ and $\widehat{\mathrm{SM}}_m^{++}(\mathbf{x},\mathbf{y})$ are shared (variant II) and when they are not (variant I). The following is true for $Z = \frac{1}{m}(1-\cos(\|\mathbf{x}\|_2^2 - \|\mathbf{y}\|_2^2)\mathbb{E}[\widehat{\lambda}(\theta)(1-\widehat{\lambda}(\theta))]$:

$$\mathrm{Cov}(\widehat{\lambda}(\theta)\widehat{\mathrm{SM}}_m^{++}(\mathbf{x},\mathbf{y}), (1-\widehat{\lambda}(\theta))\widehat{\mathrm{SM}}_m^{\mathrm{trig}}(\mathbf{x},\mathbf{y})) = \begin{cases} -(\mathrm{SM}(\mathbf{x},\mathbf{y}))^2\mathrm{Var}(\widehat{\lambda}(\theta)) \text{ for variant I} \\ -(\mathrm{SM}(\mathbf{x},\mathbf{y}))^2(\mathrm{Var}(\widehat{\lambda}(\theta)) + Z) \text{ for variant II} \end{cases} \tag{38}$$

We also have the following:

$$\begin{aligned} \mathrm{Var}(\widehat{\lambda}(\theta)\widehat{\mathrm{SM}}_m^{++}(\mathbf{x},\mathbf{y})) = \mathbb{E}[(\widehat{\lambda}(\theta))^2(\widehat{\mathrm{SM}}_m^{++}(\mathbf{x},\mathbf{y}))^2] - (\mathbb{E}[\widehat{\lambda}(\theta)\widehat{\mathrm{SM}}_m^{++}(\mathbf{x},\mathbf{y})])^2 = \\ \mathbb{E}[(\widehat{\lambda}(\theta))^2]\left(\mathrm{MSE}(\widehat{\mathrm{SM}}_m^{++}(\mathbf{x},\mathbf{y})) + (\mathrm{SM}(\mathbf{x},\mathbf{y}))^2\right) - (\mathbb{E}[\widehat{\lambda}(\theta)])^2(\mathrm{SM}(\mathbf{x},\mathbf{y}))^2 = \\ (\mathrm{SM}(\mathbf{x},\mathbf{y}))^2\mathrm{Var}(\widehat{\lambda}(\theta)) + \mathbb{E}[(\widehat{\lambda}(\theta))^2]\mathrm{MSE}(\widehat{\mathrm{SM}}_m^{++}(\mathbf{x},\mathbf{y})) \end{aligned} \tag{39}$$

and furthermore (by the analogous analysis):

$$\mathrm{Var}((1-\widehat{\lambda}(\theta))\widehat{\mathrm{SM}}_m^{\mathrm{trig}}(\mathbf{x},\mathbf{y})) = (\mathrm{SM}(\mathbf{x},\mathbf{y}))^2\mathrm{Var}(\widehat{\lambda}(\theta)) + \mathbb{E}[(1-\widehat{\lambda}(\theta))^2]\mathrm{MSE}(\widehat{\mathrm{SM}}_m^{\mathrm{trig}}(\mathbf{x},\mathbf{y})) \tag{40}$$

By putting the derived formula for the above variance terms as well as covariance terms back in the Equation 30, we complete the proof of the theorem (note that the mean squared error of the hybrid estimator is its variance since it is unbiased).

□

D ANGULAR HYBRID ESTIMATORS

D.1 PROOF OF LEMMA 3.2

Proof. The formula for the expectation is directly implied by the formula (1) from Sec. 2.1 of Cho & Saul (2012) for the zeroth-order arc-cosine kernel $k_0(\mathbf{x},\mathbf{y}) \stackrel{\mathrm{def}}{=} 1 - \frac{\theta_{\mathbf{x},\mathbf{y}}}{\pi}$. It also follows directly from Goemans-Williamson algorithm (Goemans & Williamson, 2004). We will now derive the formula for $c = \mathbb{E}[\widehat{\lambda}^2(\mathbf{x},\mathbf{y})]$. Since λ depends only on the angle $\theta = \theta_{\mathbf{x},\mathbf{y}}$, we will refer to it as: $\lambda(\theta)$ and to its estimator as $\widehat{\lambda}(\theta)$. Denote:

$$\phi_n^{\mathrm{ang}}(\mathbf{z}) = \frac{1}{\sqrt{n}}(\mathrm{sgn}(\tau_1^\top \mathbf{z}), ..., \mathrm{sgn}(\tau_n^\top \mathbf{z}))^\top \tag{41}$$

Denote: $X_i = (\phi_n^{\mathrm{ang}}(\mathbf{x}))[i]\phi_n^{\mathrm{ang}}(\mathbf{y})[i]$. We have:

$$\widehat{\lambda}(\theta) = \frac{1}{2}\left(1 - \sum_{i=1}^n X_i\right). \tag{42}$$

Note first that by the construction of $\widehat{\lambda}(\theta)$, we have: $\mathbb{E}[\widehat{\lambda}(\theta)] = \frac{\theta}{\pi}$ and thus: $\mathbb{E}[\sum_{i=1}^n X_i] = 1 - \frac{2\theta}{\pi}$. Therefore we conclude that:

$$\begin{aligned} c = \frac{1}{4}\mathbb{E}\left[1 - 2\sum_{i=1}^n X_i + \left(\sum_{i=1}^n X_i\right)^2\right] = \frac{1}{4}\left(1 - 2(1-\frac{2\theta}{\pi}) + \sum_{i=1}^n \mathbb{E}[X_i^2] + \sum_{i \neq j}\mathbb{E}[X_i]\mathbb{E}[X_j]\right) \\ = \frac{1}{4}\left(1 - 2(1-\frac{2\theta}{\pi}) + n\cdot\frac{1}{n^2} + n(n-1)\cdot\frac{1}{n^2}(1-\frac{2\theta}{\pi})^2\right) = \frac{1}{4}\left(4\frac{\theta}{\pi} + (1-\frac{1}{n})(1-\frac{2\theta}{\pi})^2\right) \\ = \frac{\theta}{\pi}\left(\frac{\theta}{\pi} - \frac{\theta}{n\pi} + \frac{1}{n}\right) \end{aligned} \tag{43}$$

□

D.2 EXPLICIT FORMULA FOR THE MSE OF THE ANGULAR HYBRID ESTIMATOR

Theorem 3.1 and Lemma 3.2 immediately imply the following result.

Theorem D.1 (MSE of the angular hybrid estimator)**.** *Take the angular hybrid estimator* $\widehat{\mathrm{SM}}_{m,n}^{\mathrm{anghyb}}(\mathbf{x},\mathbf{y})$*, where* $\widehat{\mathrm{SM}}_{m}^{\mathrm{trig}}(\mathbf{x},\mathbf{y})$ *and* $\widehat{\mathrm{SM}}_{m}^{++}(\mathbf{x},\mathbf{y})$ *are chosen independently i.e. their random projections are chosen independently (note that we always assume that* $\widehat{\lambda}(\mathbf{x},\mathbf{y})$ *is chosen independently from* $\widehat{\mathrm{SM}}_{m}^{\mathrm{trig}}(\mathbf{x},\mathbf{y})$ *and* $\widehat{\mathrm{SM}}_{m}^{++}(\mathbf{x},\mathbf{y})$*). Then the following holds:*

$$\mathrm{MSE}(\widehat{\mathrm{SM}}_{m,n}^{\mathrm{anghyb}}(\mathbf{x},\mathbf{y})) = \frac{\theta}{\pi}\left(\frac{\theta}{\pi} - \frac{\theta}{n\pi} + \frac{1}{n}\right)\frac{1}{2m}\exp(\|\mathbf{z}\|^2)\mathrm{SM}^2(\mathbf{x},\mathbf{y})(1-\exp(-\|\mathbf{z}\|^2))^2 + \\ = \left(1-\frac{\theta}{\pi}\right)\left(1-\frac{\theta}{\pi}+\frac{\theta}{n\pi}\right)\frac{1}{2m}\exp(\|\mathbf{z}\|^2)\mathrm{SM}^{-2}(\mathbf{x},\mathbf{y})(1-\exp(-\|\Delta\|^2))^2 \tag{44}$$

for $\Delta = \mathbf{x}-\mathbf{y}$ *and* $\mathbf{z} = \mathbf{x}+\mathbf{y}$*. Furthermore, if* $\widehat{\mathrm{SM}}_{m}^{\mathrm{trig}}(\mathbf{x},\mathbf{y})$ *and* $\widehat{\mathrm{SM}}_{m}^{++}(\mathbf{x},\mathbf{y})$ *apply the* ***exact*** *same sets of random projections, the mean squared error of the hybrid estimator is further reduced, namely we have:*

$$\mathrm{MSE}(\widehat{\mathrm{SM}}_{m,n}^{\mathrm{anghyb}}(\mathbf{x},\mathbf{y})) = \frac{\theta}{\pi}\left(\frac{\theta}{\pi} - \frac{\theta}{n\pi} + \frac{1}{n}\right)\frac{1}{2m}\exp(\|\mathbf{z}\|^2)\mathrm{SM}^2(\mathbf{x},\mathbf{y})(1-\exp(-\|\mathbf{z}\|^2))^2 + \\ \left(1-\frac{\theta}{\pi}\right)\left(1-\frac{\theta}{\pi}+\frac{\theta}{n\pi}\right)\frac{1}{2m}\exp(\|\mathbf{z}\|^2)\mathrm{SM}^{-2}(\mathbf{x},\mathbf{y})(1-\exp(-\|\Delta\|^2))^2 \\ -\frac{2}{m}\mathrm{SM}^2(\mathbf{x},\mathbf{y})(1-\cos(\|\mathbf{x}\|_2^2-\|\mathbf{y}\|_2^2))\frac{\theta}{\pi}\left(1-\frac{1}{n}-\frac{\theta}{\pi}+\frac{\theta}{n\pi}\right) \tag{45}$$

Thus if $\|\mathbf{x}\|_2 = \|\mathbf{y}\|_2 = r$ then regardless of whether the same sets of random projections are applied or not, we get:

$$\mathrm{MSE}(\widehat{\mathrm{SM}}_{m,n}^{\mathrm{anghyb}}(\mathbf{x},\mathbf{y})) = \frac{\theta}{\pi}\left(\frac{\theta}{\pi} - \frac{\theta}{n\pi} + \frac{1}{n}\right)\frac{1}{2m}\exp(8r^2\cos^2(\frac{\theta}{2})-2r^2)\cdot \\ (1-\exp(-4r^2\cos^2(\frac{\theta}{2})))^2 + \frac{\theta}{\pi}\left(1-\frac{1}{n}-\frac{\theta}{\pi}+\frac{\theta}{n\pi}\right)\frac{1}{2m}\exp(2r^2)(1-\exp(-4r^2\sin^2(\frac{\theta}{2})))^2 \tag{46}$$

D.3 PROOF OF THEOREM 3.5

Proof. Note first that from the derived above formula of the MSE of the lambda-angular bipolar hybrid estimator and the definition of the max-relative-error, we obtain:

$$\epsilon_{S(r)}(\widehat{\mathrm{SM}}_{m,n}^{\mathrm{anghyb}}) = \frac{\exp(r^2)}{\sqrt{2m}}\sqrt{\max_{\theta\in[0,\pi]} h_r(\theta)}, \tag{47}$$

where:

$$h_r(\theta) = a_r(\theta) + a_r(\pi-\theta) \tag{48}$$

and $a_r(\theta)$ is defined as:

$$a_r(\theta) = \frac{\theta}{\pi}(\frac{\theta}{\pi} - \frac{\theta}{n\pi} + \frac{1}{n})\exp(2r^2\cos(\theta))\left(1-\exp(-4r^2\cos^2(\frac{\theta}{2}))\right)^2. \tag{49}$$

Therefore we have:

$$\epsilon_{S(r)}(\widehat{\mathrm{SM}}_{m,n}^{\mathrm{anghyb}}) \leq \frac{\exp(r^2)}{\sqrt{m}}\sqrt{\max_{\theta\in[0,\pi]} a_r(\theta)} \tag{50}$$

Notice that:

$$a_r(\theta) \leq b_r(\theta)(1-\exp(-4r^2))^2 \tag{51}$$

where:

$$b_r(\theta) = b_r^1(\theta) + b_r^2(\theta) \tag{52}$$

and

$$b_r^1(\theta) = (1 - \frac{1}{n})\frac{\theta^2}{\pi^2}\exp(2r^2\cos(\theta)), \tag{53}$$

$$b_r^2(\theta) = \frac{1}{n} \cdot \frac{\theta}{\pi}\exp(2r^2\cos(\theta)). \tag{54}$$

Therefore:

$$\max_{\theta\in[0,\pi]} b_r(\theta) \leq \max_{\theta\in[0,\pi]} b_r^1(\theta) + \max_{\theta\in[0,\pi]} b_r^2(\theta) \tag{55}$$

Denote: $b^1 = \max_{\theta\in[0,\pi]} b_r^1(\theta)$ and $b^2 = \max_{\theta\in[0,\pi]} b_r^2(\theta)$. Note that:

$$\frac{db_r^1(\theta)}{d\theta} = \exp(2r^2\cos(\theta))(1 - \frac{1}{n})\frac{2\theta}{\pi^2}(1 - r^2\theta\sin(\theta)) \tag{56}$$

and

$$\frac{db_r^2(\theta)}{d\theta} = \exp(2r^2\cos(\theta))\frac{1}{n\pi}(1 - 2r^2\theta\sin(\theta)) \tag{57}$$

Thus, from the properties of function: $\theta \to \theta\sin(\theta)$ and the fact that $r \geq 1$, we conclude that both derivatives are first non-negative, then non-positive and then non-negative and that the unique local maximum on the interval $[0, \pi]$ is achieved for $\theta \leq \frac{\pi}{2}$. Note also that $b_r^1(\theta), b_r^1(\theta) \geq 0$ and $b_r^1(0) = b_r^2(0) = 0$, $b_r^1(\pi) = (1 - \frac{1}{n})\exp(-2r^2)$, $b_r^2(\pi) = \frac{1}{n}\exp(-2r^2)$. We conclude that global maximum for b_r^i on the interval $[0, \pi]$ for $i = 1, 2$ is achieved either in its unique local maximum on that interval or for $\theta = \pi$. Let us consider first: b_r^1. In its local maximum on $[0, \pi]$ we have:

$$\theta^*\sin(\theta^*) = \frac{1}{r^2} \tag{58}$$

Since $\theta \leq \sin(\theta) \cdot \frac{\pi}{2}$ on $[0, \frac{\pi}{2}]$, we get:

$$(\theta^*)^2 \leq \frac{\pi}{2}\frac{1}{r^2}, \tag{59}$$

i.e.:

$$\theta^* \leq \sqrt{\frac{\pi}{2}}\frac{1}{r} \tag{60}$$

Therefore:

$$b_r^1(\theta^*) \leq (1 - \frac{1}{n})\frac{1}{2\pi r^2}\exp(2r^2) \geq b_r^1(\pi) \tag{61}$$

We thus conclude that:

$$\max_{\theta\in[0,\pi]} b_r^1(\theta) \leq (1 - \frac{1}{n})\frac{1}{2\pi r^2}\exp(2r^2) \tag{62}$$

By the completely analogous analysis applied to b_r^2, we obtain:

$$\max_{\theta\in[0,\pi]} b_r^1(\theta) \leq \frac{1}{2n\sqrt{\pi}r^2}\exp(2r^2) \tag{63}$$

Now, using Equation 50, Equation 51, and Equation 55, we obtain:

$$\epsilon_{S(r)}(\widehat{\mathrm{SM}})_{m,n}^{\mathrm{anghyb}} \leq \frac{\exp(2r^2)}{\sqrt{2m}r}(1 - \exp(-4r^2))\sqrt{\frac{1}{\pi} - \frac{1}{n\pi} + \frac{1}{n\sqrt{\pi}}} \tag{64}$$

and that completes the first part of the proof (proof of Inequality 26). The equations on the limits are directly implied by the fact that:

$$\epsilon_{S(r)}(\widehat{\mathrm{SM}}_{m,n}^{\mathrm{anghyb}}) = \frac{\exp(2r^2)}{\sqrt{2m}}\sqrt{h_r(\theta)}. \tag{65}$$

□

D.4 DISCUSSION OF THEOREM 3.5

Theorem 3.5 leads to yet another important conclusion not discussed in the main body of the paper. Note that asymptotically as $\theta \rightarrow 0$ or $\theta \rightarrow \pi$, the relative error decreases (as a function of m and n) as $\frac{1}{\sqrt{mn}} = \Theta(\frac{1}{d_{\mathrm{est}}})$, where d_{est} stands for the number of random features of the resulting estimator. Note also that for the regular estimator the rate of decrease is also $\Theta(\frac{1}{d_{\mathrm{est}}})$ (here we treat all other arguments of the formula for the relative error as constants; we already know that the dependence on them of the relative error for the HRF estimators is superior to the one for regular estimators). The key difference from the computational point of view is that for the HRF estimator, the random feature map can be constructed in time $O(nd + md + mn)$, whereas for the regular estimator it requires time $O(mnd)$ (see: Sec. L). Thus for the regular estimator random feature map computation is much more expensive. An important application, where random feature map computations takes substantial time of the overall compute time are implicit-attention Transformers, such as Performers (Choromanski et al., 2021b). This is also the setting, where an overwhelming fraction of the approximate softmax kernel values will be extreme (very small or large) since an overwhelming fraction of the attention matrix entries after some training will have extreme values (very small or large). In the setting, where the lengths of queries and keys do not vary much (for instance a prominent case where they all have the same fixed length) that corresponds to angle values $\theta \rightarrow \pi$ or $\theta \rightarrow 0$. This is exactly the scenario from the second part of Theorem 3.5.

E PROOF OF LEMMA 3.4

Proof. Equation 24 follows directly from Lemma 2.2. Notice that the relative error $\epsilon_{\theta,r}(\widehat{\mathrm{SM}}_m^{\mathrm{trig}})$ is an increasing function of $\sin^2(\frac{\theta}{2})$ and thus is largest for $\theta = \pi$. Plugging in this value into the formula of the relative error gives us the expression from the statement of the lemma. Completely analogous analysis holds for $\widehat{\mathrm{SM}}_m^{++}$. □

F GAUSSIAN HYBRID ESTIMATORS

In this section we will provide more results regarding Gaussian hybrid estimators. We will focus on the bipolar scenario and within this scenario on the so-called *normalized* variant described below.

If $\mathbf{x}$ and $\mathbf{y}$ have fixed L_2-norm ($\|\mathbf{x}\|_2 = \|\mathbf{y}\|_2 = r$), we can propose a normalized bipolar Gaussian hybrid estimator such that $\lambda(\mathbf{x}, \mathbf{y}) = 0$ if $\theta_{\mathbf{x},\mathbf{y}} = 0$ and $\lambda(\mathbf{x}, \mathbf{y}) = 1$ if $\theta_{\mathbf{x},\mathbf{y}} = \pi$. Furthermore the variance of that estimator, that we call shortly $\widehat{\mathrm{SM}}_{m,n}^{\mathrm{gausshyb}}$, will be zeroed out for $\theta_{\mathbf{x},\mathbf{y}} = 0$ or $\theta_{\mathbf{x},\mathbf{y}} = \pi$, but not for both.

The coefficient-function $\lambda : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ is defined as:

$$\lambda(\mathbf{x}, \mathbf{y}) = \frac{1 - \exp(-\frac{\sigma^2}{2}\|\mathbf{x} - \mathbf{y}\|^2)}{\rho} \tag{66}$$

where ρ is given as:

$$\rho = 1 - \exp(-2\sigma^2 r^2) \tag{67}$$

The hyperparameter σ controls the smoothness of the Gaussian kernel.

The exponential in λ can be estimated either by trigonometric or positive random features. It is easy to see that for the fixed input lengths, in the former case the variance of the estimator is zero for $\theta_{\mathbf{x},\mathbf{y}} = 0$ and in the latter it is zero for $\theta_{\mathbf{x},\mathbf{y}} = \pi$, but the variance never zeroes out for both $\theta_{\mathbf{x},\mathbf{y}} = 0$ and $\theta_{\mathbf{x},\mathbf{y}} = \pi$. In principle, one can derive entire hierarchy of HRF estimators, where the coefficients in an HRF estimator from one level of hierarchy are estimated with the use of HRF estimators from the higher level. In that context, angular hybrid estimator can be applied to provide approximation of the coefficients of the normalized bipolar Gaussian hybrid estimator leading to variance zeroing out for both $\theta_{\mathbf{x},\mathbf{y}} = 0$ and $\theta_{\mathbf{x},\mathbf{y}} = \pi$. However such nested constructions are not the topic of this paper. From now on we will assume that trigonometric random features are applied to estimate λ-coefficient, Therefore we have:

$$\widehat{\lambda}(\mathbf{x},\mathbf{y}) = \frac{1}{\rho} - \frac{1}{n\rho}\sum_{i=1}^{n}\cos(\sigma(\omega_i^\top(\mathbf{x}-\mathbf{y}))) \tag{68}$$

and that leads to the following choice of: a, γ, μ from Equation 16:

$$\begin{cases} a = \frac{1}{\rho} \\ \gamma(\mathbf{z}) = \mu(\mathbf{z}) = \frac{\mathbf{i}}{\sqrt{n\rho}}(\sin(\tau_1^\top\mathbf{z}), \cos(\tau_1^\top\mathbf{z}), ..., \sin(\tau_n^\top\mathbf{z}), \cos(\tau_n^\top\mathbf{z}))^\top \end{cases} \tag{69}$$

where $\tau_1, ..., \tau_n \sim \mathcal{N}(0, \mathbf{I}_d)$ and $\mathbf{i}^2 = -1$.

Using Theorem 3.1, we conclude that:

Theorem F.1 (MSE of the normalized bipolar Gaussian estimator)**.** *Take the normalized bipolar Gaussian estimator $\widehat{\mathrm{SM}}_{m,n}^{\mathrm{gausshyb}}(\mathbf{x},\mathbf{y})$, where $\widehat{\mathrm{SM}}_{m}^{\mathrm{trig}}(\mathbf{x},\mathbf{y})$ and $\widehat{\mathrm{SM}}_{m}^{++}(\mathbf{x},\mathbf{y})$ are chosen independently i.e. their random projections are chosen independently (note that we always assume that $\widehat{\lambda}(\mathbf{x},\mathbf{y})$ is chosen independently from $\widehat{\mathrm{SM}}_{m}^{\mathrm{trig}}(\mathbf{x},\mathbf{y})$ and $\widehat{\mathrm{SM}}_{m}^{++}(\mathbf{x},\mathbf{y})$). Denote: $\Delta = \mathbf{x}-\mathbf{y}$. Then the following holds:*

$$\mathrm{MSE}(\widehat{\mathrm{SM}}_{m,n}^{\mathrm{gausshyb}}(\mathbf{x},\mathbf{y})) = \mathbb{E}[\widehat{\lambda}^2(\mathbf{x},\mathbf{y})]\mathrm{MSE}(\widehat{\mathrm{SM}}_{m}^{++}(\mathbf{x},\mathbf{y})) + \mathbb{E}[(1-\widehat{\lambda}(\mathbf{x},\mathbf{y}))^2]\mathrm{MSE}(\widehat{\mathrm{SM}}_{m}^{\mathrm{trig}}(\mathbf{x},\mathbf{y})) \tag{70}$$

where

$$\mathbb{E}[(\widehat{\lambda}(\mathbf{x},\mathbf{y}))^2] = \frac{1}{\rho^2}(1-\exp(-\frac{\sigma^2}{2}\|\Delta\|^2))^2 + \frac{1}{2\rho^2 n}(1-\exp(-\sigma^2\|\Delta\|^2))^2 \tag{71}$$

$$\mathbb{E}[(1-\widehat{\lambda}(\mathbf{x},\mathbf{y}))^2] = \frac{1}{\rho^2}((1-\rho)-\exp(-\frac{\sigma^2}{2}\|\Delta\|^2))^2 + \frac{1}{2\rho^2 n}(1-\exp(-\sigma^2\|\Delta\|^2))^2. \tag{72}$$

*Furthermore, if $\widehat{\mathrm{SM}}_{m}^{\mathrm{trig}}(\mathbf{x},\mathbf{y})$ and $\widehat{\mathrm{SM}}_{m}^{++}(\mathbf{x},\mathbf{y})$ apply the **exact** same sets of random projections, the mean squared error of the hybrid estimator remains the same.*

Proof. We can calculate $\mathbb{E}[(\widehat{\lambda}(\mathbf{x},\mathbf{y}))^2]$ and $\mathbb{E}[(1-\widehat{\lambda}(\mathbf{x},\mathbf{y}))^2]$ as follows:

$$\begin{aligned} \mathbb{E}[(1-\widehat{\lambda}(\mathbf{x},\mathbf{y}))^2] &= \frac{1}{\rho^2}\mathbb{E}[(\rho-1+\frac{1}{n}\sum_{i=1}^{n}\cos(\sigma\omega^\top(\mathbf{x}-\mathbf{y})))^2] \\ &= \frac{(1-\rho)^2}{\rho^2} + \frac{2(\rho-1)}{\rho^2}\mathbb{E}[\cos(\sigma\omega^\top(\mathbf{x}-\mathbf{y}))] + \frac{1}{\rho^2}\mathbb{E}[(\frac{1}{k}\sum_{i=1}^{n}\cos(\sigma\omega^\top(\mathbf{x}-\mathbf{y})))^2] \\ &= \frac{(1-\rho)^2}{\rho^2} + \frac{2(\rho-1)}{\rho^2}\exp(-\frac{\sigma^2}{2}\|\Delta\|^2) + \frac{1}{\rho^2}\exp(-\sigma^2\|\Delta\|^2) + \frac{1}{2\rho^2 n}(1-\exp(-\sigma^2\|\Delta\|^2))^2 \\ &= \frac{1}{\rho^2}((1-\rho)-\exp(-\frac{\sigma^2}{2}\|\Delta\|^2))^2 + \frac{1}{2\rho^2 n}(1-\exp(-\sigma^2\|\Delta\|^2))^2 \end{aligned} \tag{73}$$

$$\begin{aligned} \mathbb{E}[(\widehat{\lambda}(\mathbf{x},\mathbf{y}))^2] &= 1 - 2\mathbb{E}[1-\widehat{\lambda}(\mathbf{x},\mathbf{y})] + \mathbb{E}[(1-\widehat{\lambda}(\mathbf{x},\mathbf{y}))^2] \\ &= -1 + \frac{2}{\rho} - \frac{2}{\rho}\exp(-\frac{\sigma^2}{2}\|\Delta\|^2) + \mathbb{E}[(1-\widehat{\lambda}(\mathbf{x},\mathbf{y}))^2] \\ &= \frac{1}{\rho^2}(1-\exp(-\frac{\sigma^2}{2}\|\Delta\|^2))^2 + \frac{1}{2\rho^2 n}(1-\exp(-\sigma^2\|\Delta\|^2))^2 \end{aligned} \tag{74}$$

By plugging in these two formulae in the expression for MSE, we obtain the first half of the theorem.

Note that if $\widehat{\mathrm{SM}}_m^{\mathrm{trig}}(\theta, r)$ and $\widehat{\mathrm{SM}}_m^{++}(\theta, r)$ apply the exact same sets of random projections, then $(1 - \cos(\|\mathbf{x}\|_2^2 - \|\mathbf{y}\|_2^2))$ in $\mathrm{MSE}(\widehat{\mathrm{SM}}_{m,n}^{\mathrm{gausshyb}}(\mathbf{x}, \mathbf{y}))$ becomes zero in Theorem F.1, therefore the MSE remains the same. We thus obtain the second part of the theorem and that completes the proof.

□

Now let us assume the setting of the same-lengths inputs. We denote the length by r and an angle between inputs by θ. We can rewrite Eq. 66 and provide equivalent definition for λ by replacing $\mathbf{x}$ and $\mathbf{y}$ with their angle θ and norm r:

$$\lambda(\theta, r) = \frac{1 - \exp(-2\sigma^2 r^2 \sin(\frac{\theta}{2})^2)}{\rho} \tag{75}$$

We obtain the following version of the above theorem:

Theorem F.2 (MSE of the normalized bipolar Gaussian hybrid estimator for same-length inputs)**.** *Assume that $\|\mathbf{x}\|_2 = \|\mathbf{y}\|_2 = r$ and denote: $\theta = \theta_{\mathbf{x},\mathbf{y}}$. Take the normalized bipolar Gaussian hybrid estimator $\widehat{\mathrm{SM}}_{m,n}^{\mathrm{gausshyb}}(\theta, r)$, where $\widehat{\mathrm{SM}}_m^{\mathrm{trig}}(\theta, r)$ and $\widehat{\mathrm{SM}}_m^{++}(\theta, r)$ are chosen independently i.e. their random projections are chosen independently (note that we always assume that $\widehat{\lambda}(\theta, r)$ is chosen independently from $\widehat{\mathrm{SM}}_m^{\mathrm{trig}}(\theta, r)$ and $\widehat{\mathrm{SM}}_m^{++}(\theta, r)$). Then the following holds:*

$$\mathrm{MSE}(\widehat{\mathrm{SM}}_{m,n}^{\mathrm{gausshyb}}(\theta, r)) = \mathbb{E}[\widehat{\lambda}^2(\theta, r)]\mathrm{MSE}(\widehat{\mathrm{SM}}_m^{++}(\theta, r)) + \mathbb{E}[(1 - \widehat{\lambda}(\theta, r))^2]\mathrm{MSE}(\widehat{\mathrm{SM}}_m^{\mathrm{trig}}(\theta, r)) \tag{76}$$

where

$$\mathbb{E}[(\widehat{\lambda}(\theta, r))^2] = \frac{1}{\rho^2}(1 - \exp(-2\sigma^2 r^2 \sin(\frac{\theta}{2})^2))^2 + \frac{1}{2\rho^2 n}(1 - \exp(-4\sigma^2 r^2 \sin(\frac{\theta}{2})^2))^2 \tag{77}$$

$$\mathbb{E}[(1-\widehat{\lambda}(\theta, r))^2] = \frac{1}{\rho^2}((1-\rho)-\exp(-2\sigma^2 r^2 \sin(\frac{\theta}{2})^2))^2+\frac{1}{2\rho^2 n}(1-\exp(-4\sigma^2 r^2 \sin(\frac{\theta}{2})^2))^2 \tag{78}$$

*Furthermore, if $\widehat{\mathrm{SM}}_m^{\mathrm{trig}}(\theta, r)$ and $\widehat{\mathrm{SM}}_m^{++}(\theta, r)$ apply the **exact** same sets of random projections, the mean squared error of the hybrid estimator will remains the same.*

Proof. For $\|\mathbf{x}\|^2 = \|\mathbf{y}\|^2 = r^2$, the following holds for $\Delta = \mathbf{x} - \mathbf{y}$:

$$\|\Delta\|^2 = 4r^2 \sin(\frac{\theta}{2})^2 \tag{79}$$

And this theorem holds trivially by replacing $\|\Delta\|^2$ with the above formula in Theorem F.1.

□

G POINTWISE SOFTMAX KERNEL ESTIMATION EXPERIMENTS

In Fig. 7 we present complete ablation studies regarding empirical relative errors of different RF-based estimators of the softmax kernel over different lengths of inputs $\mathbf{r}$ and angles $\theta_{\mathbf{x},\mathbf{y}}$.

H COMPLEX EXPONENTIAL ESTIMATORS FOR CLUSTERED DATA

H.1 INSTANTIATING COMPLEX EXPONENTIAL ESTIMATORS FOR CLUSTERED DATA

Assume that the inputs $\mathbf{x}$ (queries) can be modeled by n_q clusters and that the inputs $\mathbf{y}$ (keys) can be modeled by n_k clusters (e.g. via k-means clustering algorithm). Denote the center of each cluster as $\mathbf{r}_i \in \mathbb{R}^d$, $(i = 1, ..., n_k)$ and $\mathbf{r}_j \in \mathbb{R}^d$, $(j = 1, ..., n_q)$. Then there exist $n_q n_k$ pairs of $(\mathbf{r}_i, \mathbf{r}_j)$, $(i = 1, ..., n_q, j = 1, ..., n_k)$, so we can construct $n_q n_k$ softmax kernel estimators to estimate cross-group softmax kernel values.

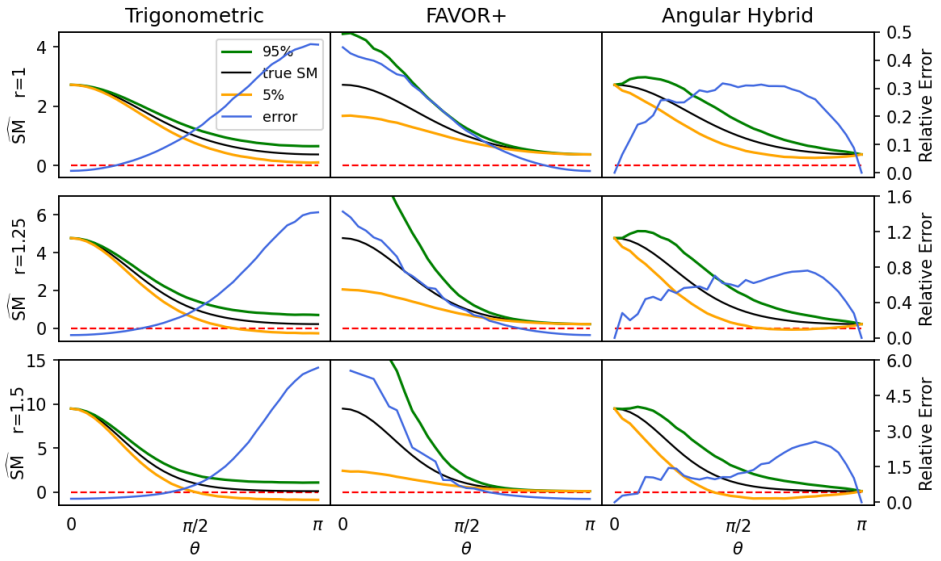

Figure 7: Pointwise estimation of $\mathrm{SM}(\mathbf{x}, \mathbf{y})$ for the same-length 64-dim inputs ($r = 1.0$, $r = 1.25$ and $r = 1.5$) and various angles $\theta_{\mathbf{x},\mathbf{y}}$. We used $s = 10000$ estimated softmax values in each subplot. The true softmax value and the 5th and 95th quantile estimated values are shown by the left y-axis, and the empirical relative errors are shown by the right y-axis. Trigonometric estimator and FAVOR+ applied 128 random features. To make fair comparison, for the hybrid variant the configuration leading to the similar number of FLOPS operations per random feature map creation was applied.

Consider $\mathbf{z} = \mathbf{A}\mathbf{r}_i + (\mathbf{A}^\top)^{-\mathbf{1}}\mathbf{r}_j$ for an invertible (in $\mathbb{C}^{d\times d}$) matrix $\mathbf{A} \in \mathbb{C}^{d\times d}$. An estimator based on this mechanism has variance equal to 0 if:

$$\mathbf{z} = \mathbf{A}\mathbf{r}_i + (\mathbf{A}^\top)^{-\mathbf{1}}\mathbf{r}_j = 0 \tag{80}$$

From now on we constrain $\mathbf{A}$ to be diagonal, so $\mathbf{A} = \mathbf{A}^\top$. We can rewrite the above equation as:

$$\mathbf{z} = \mathbf{A}\mathbf{r}_i + (\mathbf{A})^{-\mathbf{1}}\mathbf{r}_j = 0 \tag{81}$$

Since position (k, k) in matrix $\mathbf{A}$ is a complex number of the form $\alpha_k + \beta_k i$, we need to satisfy the following equation for each $k = 1, ..., d$:

$$(\alpha_k + \beta_k i)r_{i,k} + \frac{1}{\alpha_k + \beta_k i} r_{j,k} = 0 \tag{82}$$

$$(\alpha_k + \beta_k i)r_{i,k} + \frac{\alpha_k - \beta_k i}{\alpha_k^2 + \beta_k^2} r_{j,k} = 0 \tag{83}$$

where $r_{i,k}$ is the k-th entry of vector $\mathbf{r}_i$.

We can simplify this equation and separate into real and imaginary part:

$$\begin{cases} \mathbf{Re}: (\alpha_k^2 + \beta_k^2)\alpha_k r_{i,k} + \alpha_k r_{j,k} = 0 \\ \mathbf{Im}: (\alpha_k^2 + \beta_k^2)\beta_k r_{i,k} - \beta_k r_{j,k} = 0 \end{cases} \tag{84}$$

Our goal now is to choose values for α_k and β_k given $\mathbf{r}_i$ and $\mathbf{r}_j$.

If $r_{i,k} r_{j,k} > 0$, we can set:

$$\begin{cases} \alpha_k = 0 \\ \beta_k = \sqrt{r_{j,k}/r_{i,k}} \end{cases} \tag{85}$$

And if $r_{i,k} r_{j,k} < 0$, we can set:

$$\begin{cases} \alpha_k = \sqrt{-r_{j,k}/r_{i,k}} \\ \beta_k = 0 \end{cases} \tag{86}$$

When $r_{i,k} = r_{j,k} = 0$, we can take $\alpha_k = \beta_k = 1$. If $r_{i,k} = 0, r_{j,k} \neq 0$ or the opposite, then we cannot satisfy the above equation perfectly. We can take α_k to some large positive value and set $\beta_k = 0$ when $r_{i,k} = 0, r_{j,k} \neq 0$, and set $\alpha_k = \beta_k$ to some small positive value close to zero when $r_{i,k} \neq 0, r_{j,k} = 0$.

When restricting $\mathbf{A} = \mathrm{diag}(a_1, \ldots, a_d)$ to $\mathbb{R}^{d \times d}$, it is easily seen that to minimize $|\mathbf{A}\mathbf{r}_i + (\mathbf{A}^\top)^{-1}\mathbf{r}_j|^2$, we can take

$$a_k = \sqrt{|r_{j,k}/r_{i,k}|} \tag{87}$$

if $r_{i,k} \neq 0$. And if $r_{i,k} = 0, r_{j,k} \neq 0$, a_k can be set to a large positive number. We can set $a_k = 1$ if $r_{i,k} = 0, r_{j,k} = 0$.

In the analysis below we denote matrix $\mathbf{A}$ calculated from Eq. 84 and Eq. 87 (depending on whether we are using real or complex matrices) given $\mathbf{r}_i$ and $\mathbf{r}_j$ as $\mathbf{A}^{i,j}$.

From the analysis in the main body of the paper we know that for arbitrary vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$:

$$\exp(\frac{\|\mathbf{A}^{i,j}\mathbf{x}\|^2}{2})\exp(\frac{\|(\mathbf{A}^{i,j})^{-1}\mathbf{y}\|^2}{2})\mathrm{SM}(\mathbf{x}, \mathbf{y}) = \mathbb{E}[\exp(\omega^\top(\mathbf{A}^{i,j}\mathbf{x} + (\mathbf{A}^{i,j})^{-1}\mathbf{y})] \tag{88}$$

Therefore, the softmax kernel can be estimated as:

$$\mathrm{SM}(\mathbf{x}, \mathbf{y}) \approx \Psi^m_{\mathbf{A}^{i,j}}(\mathbf{x})^\top \Psi^m_{(\mathbf{A}^{i,j})^{-1}}(\mathbf{y}) \tag{89}$$

with $\Psi^m_{\mathbf{A}^{i,j}}(\mathbf{x})$ and $\Psi^m_{(\mathbf{A}^{i,j})^{-1}}(\mathbf{y})$ given by:

$$\Psi^m_{\mathbf{A}^{i,j}}(\mathbf{x}) \stackrel{\mathrm{def}}{=} \frac{1}{\sqrt{m}}\exp(-\frac{\|\mathbf{A}^{i,j}\mathbf{x}\|^2}{2})(\exp(\omega_1^\top \mathbf{A}^{i,j}\mathbf{x}), ..., \exp(\omega_m^\top \mathbf{A}^{i,j}\mathbf{x}))^\top \tag{90}$$

$$\Psi^m_{(\mathbf{A}^{i,j})^{-1}}(\mathbf{y}) \stackrel{\mathrm{def}}{=} \frac{1}{\sqrt{m}}\exp(-\frac{\|(\mathbf{A}^{i,j})^{-1}\mathbf{y}\|^2}{2})(\exp(\omega_1^\top (\mathbf{A}^{i,j})^{-1}\mathbf{y}), ..., \exp(\omega_m^\top (\mathbf{A}^{i,j})^{-1}\mathbf{y}))^\top \tag{91}$$

for $\omega_1, ..., \omega_m \sim \mathcal{N}(0, \mathbf{I}_d)$.

Such a softmax kernel estimator has variance equal to zero if $\mathbf{x} = \mathbf{r}_i$ and $\mathbf{y} = \mathbf{r}_j$ if we are using the complex matrix and has small variance if we are using the real matrix. For other pairs of vectors $(\mathbf{x}, \mathbf{y})$ close to $\mathbf{r}_i$ and $\mathbf{r}_j$, the variance is close to zero, or relatively small.

We can also take the Gaussian λ-coefficient estimator that gets it maximum value when $\mathbf{x} = \mathbf{r}_i, \mathbf{y} = \mathbf{r}_j$ as $\lambda^{i,j}(\mathbf{x}, \mathbf{y})$. Such an estimator could be constructed with $\mathbf{A}^{i,j}$ in a similar way.

It can be easily linearized since:

$$\begin{gathered}\lambda^{i,j}(\mathbf{x}, \mathbf{y}) = \exp(-\frac{\|\mathbf{A}^{i,j}\mathbf{x} + (\mathbf{A}^{i,j})^{-1}\mathbf{y}\|^2}{2\tau^2}) = \\ \exp(-\frac{\|\mathbf{A}^{i,j}\mathbf{x}\|^2}{2\tau^2})\exp(-\frac{\|(\mathbf{A}^{i,j})^{-1}\mathbf{y}\|^2}{2\tau^2})\mathrm{SM}(\frac{\mathbf{x}}{\tau}, \frac{-\mathbf{y}}{\tau}),\end{gathered} \tag{92}$$

where $\mathrm{SM}(\mathbf{x}/\tau, -\mathbf{y}/\tau)$ could be estimated with similar random feature maps as given by Eq. 89.

Therefore, the Gaussian λ-coefficients can be estimated as:

$$\lambda^{i,j}(\mathbf{x}, \mathbf{y}) = \exp(-\frac{\|\mathbf{A}^{i,j}\mathbf{x} + (\mathbf{A}^{i,j})^{-1}\mathbf{y}\|^2}{2\tau^2}) \approx \Psi^{\tau,m}_{\mathbf{A}^{i,j}}(\mathbf{x})^\top \Psi^{\tau,m}_{(\mathbf{A}^{i,j})^{-1}}(\mathbf{y}) \tag{93}$$

with $\Psi^{\tau,m}_{\mathbf{A}^{i,j}}(\mathbf{x})$ and $\Psi^{\tau,m}_{(\mathbf{A}^{i,j})^{-1}}(-\mathbf{y})$ given by:

$$\Psi^{\tau,m}_{\mathbf{A}^{i,j}}(\mathbf{x}) = \frac{1}{\sqrt{m}}\exp(-\frac{\|\mathbf{A}^{i,j}\mathbf{x}\|^2}{\tau^2})(\exp(\omega_1^\top \mathbf{A}^{i,j}\mathbf{x}/\tau), ..., \exp(\omega_m^\top \mathbf{A}^{i,j}\mathbf{x}/\tau))^\top \tag{94}$$

$$\Psi^{\tau,m}_{(\mathbf{A}^{i,j})^{-1}}(-\mathbf{y}) = \frac{1}{\sqrt{m}}\exp(-\frac{\|(\mathbf{A}^{i,j})^{-1}\mathbf{y}\|^2}{\tau^2})(\exp(-\omega_1^\top (\mathbf{A}^{i,j})^{-1}\mathbf{y}/\tau), ..., \exp(-\omega_m^\top (\mathbf{A}^{i,j})^{-1}\mathbf{y}/\tau))^\top \tag{95}$$

for $\omega_1, ..., \omega_m \sim \mathcal{N}(0, \mathbf{I}_d)$ chosen independently and that were applied in base estimators.

We summarize this section with the following two constructions of the hybrid estimators for the clustered data that are implied by the above analysis.

Definition H.1 (Hybrid Gaussian-Mixtures estimators for clustered data)**.** *Assume that inputs* $\mathbf{x}$ *(queries) and* $\mathbf{y}$ *(keys) can be modeled by* n_q *and* n_k *clusters respectively with centers* $\mathbf{r}_i$ *and* $\mathbf{r}_j \in \mathbb{R}^d$*, (*$i = 1, ..., n_k$*,* $j = 1, ..., n_q$*). We denote* $\mathbf{A}^{i,j}$ *as the complex matrix satisfying Eq. (84) with center* $\mathbf{r}_i, \mathbf{r}_j$*. Furthermore, we denote* $\mathcal{E} = (\widehat{\mathrm{SM}}^{i,j}(\mathbf{x}, \mathbf{y}))_{i=1,j=1}^{n_q,n_k}$ *as a list of estimators of the softmax kernel* $\mathrm{SM}^{i,j}(\mathbf{x}, \mathbf{y})$ *and* $\Lambda = \widehat{\lambda}^{i,j}(\mathbf{x}, \mathbf{y}))_{i=1,j=1}^{n_q,n_k}$ *as a list of estimators of* $\lambda^{i,j}(\mathbf{x}, \mathbf{y})$ *constructed independently from* $\mathcal{E}$*. We also use one additional base estimator (e.g.* $\widehat{\mathrm{SM}}^{\mathrm{trig}}(\mathbf{x}, \mathbf{y})$ *or* $\widehat{\mathrm{SM}}^{++}(\mathbf{x}, \mathbf{y})$*) and denote the estimator of its softmax kernel as* $\widehat{\mathrm{SM}}^{0}(\mathbf{x}, \mathbf{y})$ *and* λ*-coefficient as* $\widehat{\lambda}^0(\mathbf{x}, \mathbf{y})$*. Then our hybrid estimator takes the following form:*

$$\widehat{\mathrm{SM}}^{\mathcal{E},\Lambda}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n_q}\sum_{j=1}^{n_k} \widehat{\lambda}^{i,j}(\mathbf{x}, \mathbf{y})\widehat{\mathrm{SM}}^{i,j}(\mathbf{x}, \mathbf{y}) + \widehat{\lambda}^0(\mathbf{x}, \mathbf{y})\widehat{\mathrm{SM}}^{0}(\mathbf{x}, \mathbf{y}) \tag{96}$$

with constraint:

$$\sum_{i=1}^{n_q}\sum_{j=1}^{n_k} \widehat{\lambda}^{i,j}(\mathbf{x}, \mathbf{y}) + \widehat{\lambda}^0(\mathbf{x}, \mathbf{y}) = 1 \tag{97}$$

and where the estimators of λ*-coefficients and base estimators are given as:*

$$\widehat{\lambda}^{i,j}(\mathbf{x}, \mathbf{y}) = \Psi^{\tau,m}_{\mathbf{A}^{i,j}}(\mathbf{x})^\top \Psi^{\tau,m}_{(\mathbf{A}^{i,j})^{-1}}(-\mathbf{y}) \tag{98}$$

$$\widehat{\mathrm{SM}}^{i,j}(\mathbf{x}, \mathbf{y}) = \Psi^{m}_{\mathbf{A}^{i,j}}(\mathbf{x})^\top \Psi^{m}_{(\mathbf{A}^{i,j})^{-1}}(\mathbf{y}) \tag{99}$$

for $\Psi^{m}_{\mathbf{A}^{i,j}}(\mathbf{x}), \Psi^{m}_{(\mathbf{A}^{i,j})^{-1}}(\mathbf{y}), \Psi^{\tau,m}_{\mathbf{A}^{i,j}}(\mathbf{x}), \Psi^{\tau,m}_{(\mathbf{A}^{i,j})^{-1}}(-\mathbf{y})$ *given by Eq. (90, 91, 94, 95).*

Definition H.2 (Hybrid Zero-One-Mixtures estimators for clustered data)**.** *Assume that inputs* $\mathbf{x}$ *(queries) and* $\mathbf{y}$ *(keys) can be modeled by* n_q *and* n_k *clusters respectively with centers* $\mathbf{r}_i$ *and* $\mathbf{r}_j \in \mathbb{R}^d$*, (*$i = 1, ..., n_k$*,* $j = 1, ..., n_q$*). We denote* $\mathbf{A}^{i,j}$ *as the complex matrix satisfying Eq. (84) with center* $\mathbf{r}_i, \mathbf{r}_j$*. Furthermore, we denote* $\mathcal{E} = (\widehat{\mathrm{SM}}^{i,j}(\mathbf{x}, \mathbf{y}))_{i=1,j=1}^{n_q,n_k}$ *as a list of estimators of the softmax kernel* $\mathrm{SM}^{i,j}(\mathbf{x}, \mathbf{y})$ *and* $\Lambda = \widehat{\lambda}^{i,j}(\mathbf{x}, \mathbf{y}))_{i=1,j=1}^{n_q,n_k}$ *as a list of estimators of* $\lambda^{i,j}(\mathbf{x}, \mathbf{y})$ *constructed independently from* $\mathcal{E}$*.*

Then our hybrid estimator takes the following form:

$$\widehat{\mathrm{SM}}^{\mathcal{E},\Lambda}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n_q}\sum_{j=1}^{n_k} \widehat{\lambda}^{i,j}(\mathbf{x}, \mathbf{y})\widehat{\mathrm{SM}}^{i,j}(\mathbf{x}, \mathbf{y}) \tag{100}$$

with constraint:

$$\sum_{i=1}^{n_q}\sum_{j=1}^{n_k} \widehat{\lambda}^{i,j}(\mathbf{x}, \mathbf{y}) = 1 \tag{101}$$

and where the estimators of λ-coefficients and base estimators are given as:

$$\widehat{\lambda}^{i,j}(\mathbf{x},\mathbf{y}) = \Psi_i(\mathbf{x})\Psi_j(\mathbf{y}) \tag{102}$$

$$\widehat{\mathrm{SM}}^{i,j}(\mathbf{x},\mathbf{y}) = \Psi^m_{\mathbf{A}^{i,j}}(\mathbf{x})^\top \Psi^m_{(\mathbf{A}^{i,j})^{-1}}(\mathbf{y}) \tag{103}$$

for $\Psi^m_{\mathbf{A}^{i,j}}(\mathbf{x}), \Psi^m_{(\mathbf{A}^{i,j})^{-1}}(\mathbf{y})$ given by Eq. (90, 91), with $\Psi_i(\mathbf{x})$ being a scalar indicating whether $\mathbf{x}$ belongs to the i-th cluster of the n_q clusters and similarly with $\Psi_j(\mathbf{y})$ being a scalar indicating whether $\mathbf{y}$ belongs to the j-th of the n_k clusters.

In our experiments we used the formulation from the second definition with $\mathbf{A}$ constrained to be real diagonal. In other words, we are using Eq. 87 to construct $\mathbf{A}$.

H.2 Additional experimental details

To create the synthetic data, we first generate two random vectors X_0, Y_0 in $\mathbb{R}^{50}$. Then we generate four random orthogonal matrices $O^{i,j} \in \mathbb{R}^{50\times 50}$ where $i = 1, 2$ and $j = 1, 2$ and use $X_i = O^{i,1}X_0$ and $Y_i = O^{i,2}Y_0$ for $i = 1, 2$ as the mean vectors for our Gaussian clusters. For each pair of clusters, the minimal value s of Eq. 20 may be non-zero if $\mathrm{sign}(\mathbf{x}_{i,k}) = \mathrm{sign}(\mathbf{y}_{j,k})$ for some k due to the usage of real diagonal matrices. To control the value of s, we manually adjust the sign of each element of X_i and Y_i. The data for input $\mathbf{x}$ is the combination of two clusters, each consisting 1000 data points following Gaussian distribution with mean vectors X_1, X_2 and the common covariance matrix $\sigma^2 I$. And similar constructions are done for input $\mathbf{y}$. We create four synthetic data sets, with different values of s and magnitudes of the standard deviation σ. After this step, we normalize the data points by controlling their ℓ_2 norms to make the true softmax value be within a reasonable range. After the data is created, we first use K-means clustering algorithm with $k = 2$ to cluster $\mathbf{x}$ and $\mathbf{y}$, and we obtain centers $(\mathbf{x}_i, \mathbf{y}_j)_{i,j\in\{1,2\}}$. Then we use the set of real diagonal matrices to minimize Eq. 20 for all pairs of centers of $\mathbf{x}$ and $\mathbf{y}$ to generate four complex exponential base estimators. To choose λ coefficients, we use the more efficient approach described above, i.e. using indicator vectors, and these are already computed in the clustering process. For each data set, we compare the performance of these estimators by calculating the mean square error of them, and we do this multiple times with different numbers of random features used. All the results are obtained by averaging over 20 repetitions. In addition to the average performance which can be found in Fig. 3, we also record the maximal and minimal empirical MSE over 20 repetitions for all estimators in the Table 2, where $s_1 = [0.08, 0.05, 0.04, 0.03]$, $s_2 = [0.11, 0.09, 0.08, 0.07]$, $s_3 = [0.05, 0.03, 0.03, 0.02]$, $s_4 = [0.07, 0.06, 0.05, 0.04]$ representing the lists of s values for all pairs of clusters for different synthetic data sets.

I Language Modeling Training Details and Additional Results

For the Language Modeling tasks, we trained a 2-layer LSTM of hidden sizes 200 and 650 on the PennTree Bank (Marcus et al., 1993) and the WikiText2 dataset (Merity et al., 2017) respectively. We tied the weights of the word embedding layer and the decoder layer (Inan et al., 2017; Press & Wolf, 2017). Thus we could treat the language modeling problem as minimizing the cross entropy loss between the dot product of the model output (queries) and the class embedding (keys) obtained from the embedding layer with the target word.

Since the HRF mechanisms can be sensitive to the lengths of the keys and the queries, for the language modeling task on the WikiText2 dataset, we added a regularizer to constrain the lengths of the keys and the queries to be close to 1. However, constraining the lengths close to 1 hurts the model performance and so we chose to use a temperature scaling as in (Rawat et al., 2019). Thus before passing the computed dot product to the softmax layer, we scaled the dot product by $\tau > 1$. Our final loss function for the language modeling task was:

$$\mathcal{L}(\theta) = L_{CE} + \lambda_1 \mathbb{E}(||q||_2 - \mathbf{1})^2 + \lambda_2 \mathbb{E}(||k||_2 - \mathbf{1})^2 \tag{104}$$

Data Set	Number of Random Features	Hybrid-Cluster	FAVOR+	Trigonometric
$s = s_1, \sigma = 1$	20	[0.0016, 0.0183]	[0.0071, 0.1455]	[0.0039, 0.0808]
	50	[0.0011, 0.0081]	[0.0026, 0.2541]	[0.0031, 0.0510]
	80	[0.0005, 0.0052]	[0.0032, 0.0282]	[0.0010, 0.0281]
	100	[0.0006, 0.0036]	[0.0022, 0.0528]	[0.0013, 0.0170]
	120	[0.0004, 0.0025]	[0.0028, 0.0534]	[0.0012, 0.0135]
	150	[0.0003, 0.0020]	[0.0014, 0.0582]	[0.0007, 0.0134]
	200	[0.0002, 0.0010]	[0.0013, 0.0205]	[0.0004, 0.0087]
$s = s_2, \sigma = 1$	20	[0.0032, 0.0265]	[0.0095, 0.2071]	[0.0036, 0.0989]
	50	[0.0012, 0.0079]	[0.0036, 0.2123]	[0.0019, 0.0565]
	80	[0.0007, 0.0059]	[0.0037, 0.0570]	[0.0008, 0.0208]
	100	[0.0007, 0.0040]	[0.0016, 0.0280]	[0.0010, 0.0164]
	120	[0.0005, 0.0025]	[0.0013, 0.0262]	[0.0007, 0.0246]
	150	[0.0004, 0.0024]	[0.0010, 0.0407]	[0.0004, 0.0190]
	200	[0.0003, 0.0018]	[0.0018, 0.0236]	[0.0004, 0.0134]
$s = s_3, \sigma = \sqrt{10}$	20	[0.0083, 0.0300]	[0.0362, 0.1218]	[0.0174, 0.0490]
	50	[0.0046, 0.0183]	[0.0125, 0.1262]	[0.0081, 0.0249]
	80	[0.0029, 0.0105]	[0.0108, 0.0281]	[0.0044, 0.0143]
	100	[0.0024, 0.0056]	[0.0077, 0.0421]	[0.0041, 0.0095]
	120	[0.0020, 0.0046]	[0.0077, 0.0402]	[0.0034, 0.0077]
	150	[0.0015, 0.0041]	[0.0049, 0.0395]	[0.0025, 0.0073]
	200	[0.0011, 0.0031]	[0.0037, 0.0150]	[0.0018, 0.0050]
$s = s_4, \sigma = \sqrt{10}$	20	[0.0097, 0.0294]	[0.0332, 0.1553]	[0.0179, 0.0485]
	50	[0.0049, 0.0172]	[0.0176, 0.1345]	[0.0058, 0.0260]
	80	[0.0031, 0.0113]	[0.0113, 0.0415]	[0.0038, 0.0110]
	100	[0.0028, 0.0061]	[0.0080, 0.0292]	[0.0031, 0.0100]
	120	[0.0022, 0.0045]	[0.0065, 0.0185]	[0.0028, 0.0128]
	150	[0.0016, 0.0044]	[0.0045, 0.0235]	[0.0021, 0.0094]
	200	[0.0012, 0.0031]	[0.0046, 0.0166]	[0.0015, 0.0067]

Table 2: Minimal and maximal empirical MSE with 20 repetitions

where $||\cdot||_2$ is the row-wise L_2 norm of the appropriate matrices, and L_{CE} is the cross-entropy loss. The distribution of the lengths of the keys and the queries are shown in figure 8.

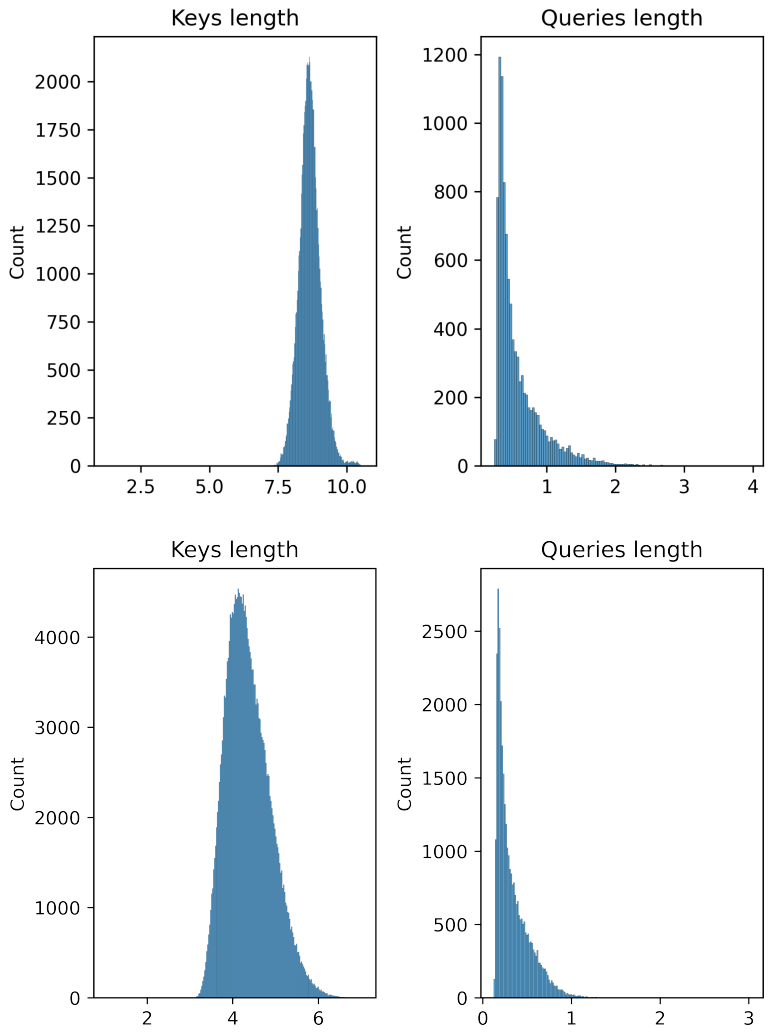

Figure 8: Distribution of the lengths of the keys and queries in the PennTree Bank dataset (Top row) and the WikiText2 dataset (Bottom Row)

For our experiments, we chose $\lambda_1 = \lambda_2 = 2$ and $\tau = 6$. Our model trained with these hyperparameters achieve a perplexity score of 105.35 on the test set after 50 epochs.

Finally we present some additional results on the WikiText2 dataset. We compute the 1-dimensional Wasserstein distance and the Kolmogrov-Smirnov (KS) metric (Ramdas et al., 2017) between the approximated softmax distribution and the true softmax distribution. As in our experiments with the PennTree Bank dataset, for the Angular Hybrid and the Gaussian Hybrid, the number of random features are $8, 16, 32, 64$ for the base estimators, and 8 for the coefficient estimators.

I.1 Language Modeling using HRF

In this subsection, we will describe how one can use HRF to train a LSTM on the language modeling task on the PennTree Bank dataset, similar to the experiments carried out in (Rawat et al., 2019). We trained a 2-layer LSTM model with hidden and output sizes of 200, and used the output as input embedding for sampled softmax. We sampled 40 negative classes out of 10000 classes to

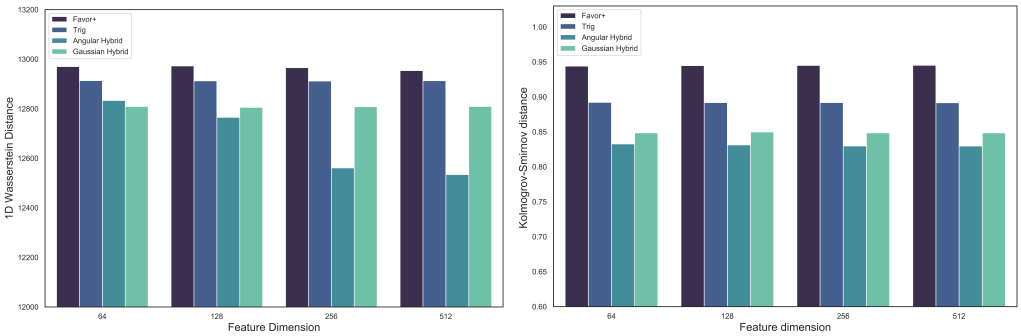

Figure 9: 1-dim Wasserstein distance and Kolmogrov-Smirnov distance on the WikiText2 dataset.

approximate the expected loss in each training epoch. As observed in Fig. 2, the relative error of all three estimators grow exponentially fast with embedding norm r. Therefore, if we keep un-normalized embeddings during sampling, then even though we could get an unbiased estimation of the loss, the variance could be high. Such bias-variance trade off is also mentioned in paper (Rawat et al., 2019). To solve this issue, we used normalized input and class embeddings during sampling to generate biased sampling distribution with lower variance, while keeping un-normalized embeddings to calculate the loss. We trained our model for 50 epochs, with batch size equal to 20, dropout ratio in LSTM equal to 0.5. Implementation details could be seen in our Github repository.

We compared our hybrid estimator with Trigonometric, FAVOR+, the uniform method which sampled 40 negative classes with equal probability, as well as the sampled softmax method which uses the unbiased true probability calculated from full softmax as the sampling probability. Trigonometric and FAVOR+ mechanisms used 1024 random features. To make comparison fair, hybrid variants were using (m, n)-configurations characterized by the similar number of FLOPS needed to construct their corresponding random features as regular RF-estimators. We reported the best hybrid method (which is Angular Hybrid estimator) in Fig. 5. As we can see, FAVOR+ and our hybrid estimator are better than Trigonometric, and they are all better than the uniform method. Moreover, our hybrid estimator has better perplexity than FAVOR+ in the last 10 training epochs, and approximates the sampled softmax method quite well.

J Speech Experiments: Additional Details

Tested Conformer-Performer models consisted of $l = 17$ conformer layers. Each attention layer used $H = 8$ heads. The embedding dimensionality was $p = 512$ and since dimensions were split equally among different heads, query/key dimensionality was set up to $d_{\mathrm{QK}} = 64$. Each input sequence was of length $L \approx 500$. Padding mechanism was applied for all tested variants.

K Downstream Robotics Experiments

In both Robotics experiments we found query/key L_2-normalization technique particularly effective (queries and keys of L_2-norm equal to one) thus we applied it in all the experiments.

K.1 Visual Locomotion with Quadruped Robots

We evaluate HRF-based attention RL policies in a robotic task of learning locomotion on uneven terrains from vision input. We use the quadruped from Unitree called Laikago (lai). It has 12 actuated joints, 3 per leg. The task is to walk forward on a randomized uneven terrain that requires careful foot placement planning based on visual feedback. The ground is made of a series of step-stones with gaps in between. The step stones widths are fixed at 50 cm, the lengths are between $[50, 80]$ cm in length, and the gap size between adjacent stones are between $[10, 20]$ cm. It perceives the ground through 2 depth cameras attached to its body, one on the front and other on the belly facing downwards. We use Implicit Attention Policy (IAP) architecture (masking variant) described

in Choromanski et al. (2021a) which uses Performer-based attention mechanism to process 32×24 depth images from the 2 cameras.

We apply a hierarchical setup to solve this task. The high level uses IAP-rank with masking to process camera images and output the desired foot placement position. The low level employs a position-based swing leg controller, and an model predictive control (MPC) based stance leg controller, to achieve the foot placement decided by high level. The policies are trained with evolutionary strategies (ES). In Fig. 6 (left) we compare FAVOR+ with angular hybrid approximation in IAP. HRF with 8 x 8 random projections ($m = n = 8$) performs as well as the softmax kernel with 256 random projections. A series of image frames along the episode of a learned locomotion HRF-based IAP policy is shown bottom right of Fig. 6. On the top-left corner of the images, the input camera images are attached. The red part of the camera image is the area masked out by self-attention. The policy learns to pay attention to the gaps in the scene in order to avoid stepping on them.

The training curves in Fig. 6 (left) are obtained by averaging over 5 top runs. The shaded regions shows the standard deviation over multiple runs.

K.2 Robotic Manipulation: Bi-manual Sweeping

Here we consider the bi-manual sweep robotic-arm manipulation. The two-robotic-arm system needs to solve the task of placing red balls into green bowls (all distributions are equally rewarded, so in particular robot can just use one bowl).

In this bi-manual sweeping task, the reward per step is between 0 and 1, the fraction of the blocks within the green bowls. The loss for this task is the negative log likelihood between the normalized softmax probability distribution of sampled actions (one positive example, and all others uniform negative counter-examples), and the ground truth one-hot probability distribution for a given observation from the training dataset. For more details see (van den Oord et al., 2018). This is a 12-DoF cartesian control problem, the state observation consists of the current cartesian pose of the arms and a $500 \times 500 \times 3$ RGB image. An example of the camera image is given in Fig. 10.

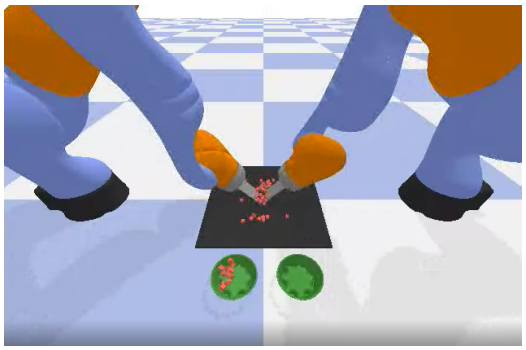

Figure 10: An example of the camera image for the bi-manual sweep robotic-arm manipulation task.

The training curves Fig. 6 (right) are obtained by averaging over 3 learning rates: 10^{-3}, 10^{-4}, 10^{-5}.

L Computational gains of hybrid estimators

Consider the general hybrid estimator of $\mathrm{SM}(\mathbf{x}, \mathbf{y})$ from Equation 14, using $p+1$ base estimators. Assume that the kth base estimator for $k = 1, ..., p+1$ applies m random features in each of the corresponding t_k random maps and that n random features are used in each of the corresponding l_{λ_k} random maps to approximate k^{th} λ-coefficient for $k = 1, ..., p$. From Equation 17, we conclude that time complexity of constructing $\Psi(\mathbf{z})$ is:

$$T = O\left((t_1 + ... + t_{p+1})md + (nd + md + mn)\sum_{k=1}^{p+1}\sum_{r=1}^{p} t_k l_r\right) \tag{105}$$

Thus the resulting $\Theta(mn)$-dimensional random feature map $\Psi(\mathbf{z})$ can be constructed in time $O(nd + md + mn)$ as opposed to $O(mnd)$ as it is the case for the regular estimator, providing substantial

computational gains. This is true if the $(\theta)(mn)$-dimensional HRFs can provide similar quality as their mn-dimensional regular counterparts. It turns out that this is often the case, see: Section D.4.