A Compact Kernel Approximation for Efficient 3D Action Recognition

Jacopo Cavazza[1,2], Pietro Morerio[1], and Vittorio Murino[1,3]

[1]Pattern Analysis & Computer Vision (PAVIS) - Istituto Italiano di Tecnologia - *Genova, Italy*
[2]Electrical, Electronics and Telecommunication Engineering and Naval Architecture Department (DITEN) – Università degli Studi di Genova – *Genova, Italy*
[3]Computer Science Department – Università di Verona – *Verona, Italy*
`firstname.lastname@iit.it`

Abstract. 3D action recognition was shown to benefit from a covariance representation of the input data (joint 3D positions). A kernel machine feed with such feature is an effective paradigm for 3D action recognition, yielding state-of-the-art results. Yet, the whole framework is affected by the well-known scalability issue. In fact, in general, the kernel function has to be evaluated for all pairs of instances inducing a Gram matrix whose complexity is quadratic in the number of samples. In this work we reduce such complexity to be linear by proposing a novel and explicit feature map to approximate the kernel function. This allows to train a linear classifier with an explicit feature encoding, which implicitly implements a Log-Euclidean machine in a scalable fashion. Not only we prove that the proposed approximation is unbiased, but also we work out an explicit strong bound for its variance, attesting a theoretical superiority of our approach with respect to existing ones. Experimentally, we verify that our representation provides a compact encoding and outperforms other approximation schemes on a number of publicly available benchmark datasets for 3D action recognition.

Keywords: Action Recognition, 3D, Kernel, Feature Map

1 Introduction

Action recognition is a key research domain in video/image processing and computer vision, being nowadays ubiquitous in human-robot interaction, autonomous driving vehicles, elderly care and video-surveillance to name a few [21]. Yet, challenging difficulties arise due to visual ambiguities (illumination variations, texture of clothing, general background noise, view heterogeneity, occlusions). As an effective countermeasure, joint-based skeletal representations (extracted from depth images) are a viable solution.

Combined with a skeletal representation, the symmetric and positive definite (SPD) covariance operator scores a sound performance in 3D action recognition [22,9,5]. Indeed, while properly modeling the skeletal dynamics with a second order statistic, the covariance operator is also naturally able to handle different temporal duration of action instances. This avoids slow pre-processing stages such as time warping or interpolation [20]. In addition, the superiority of such representation can be attested by achieving state-of-the-art performance by means of a relative simple classification pipeline [22,5] where, basically[1], a non-linear Support Vector Machine (SVM) is trained using the Log-Euclidean kernel

$$K_{\ell E}(\mathbf{X},\mathbf{Y}) = \exp\left(-\frac{1}{2\sigma^2}\|\log\mathbf{X} - \log\mathbf{Y}\|_F^2\right) \tag{1}$$

to compare covariance operators $\mathbf{X}$, $\mathbf{Y}$. In (1), for any SPD matrix $\mathbf{X}$, we define

$$\log\mathbf{X} = \mathbf{U}\mathrm{diag}(\log\lambda_1,\ldots,\log\lambda_d)\mathbf{U}^\top, \tag{2}$$

being $\mathbf{U}$ the matrix of eigenvectors which diagonalizes $\mathbf{X}$ in terms of the eigenvalues $\lambda_1 \geq \cdots \geq \lambda_d > 0$. Very intuitively, for any fixed bandwidth $\sigma > 0$, $K_{\ell E}(\mathbf{X},\mathbf{Y})$ is actually computing a radial basis Gaussian function by comparing the covariance operators $\mathbf{X}$ and $\mathbf{Y}$ by means of the Frobenius norm $\|\cdot\|_F$ (after $\mathbf{X},\mathbf{Y}$ have been log-projected). Computationally, the latter stage is not problematic (see Section 3) and can be performed for each covariance operator *before* computing the kernel. In addition to its formal properties in Riemannian geometry, this makes (1) widely used in practice [9,22,5].

However, the modern big data regime mines the applicability of such a kernel function. Indeed, since (1) has to be computed for *every pair* of instances in the dataset, the so produced Gram matrix has prohibitive size. So its storage becomes time- and memory-expensive and the related computations (required to train the model) are simply unfeasible.

The latter inconvenient can be solved as follows. According to the well established kernel theory [2], the Kernel (1) induces an infinite-dimension feature map φ, meaning that $K_{\ell E}(\mathbf{X},\mathbf{Y}) = \langle\varphi(\mathbf{X}),\varphi(\mathbf{Y})\rangle$. However, if we are able to obtain an explicit feature map Φ such that $K_{\ell E}(\mathbf{X},\mathbf{Y}) \approx \langle\Phi(\mathbf{X}),\Phi(\mathbf{Y})\rangle$, we can directly compute a finite-dimensional feature representation $\Phi(\mathbf{X})$ for each action instance separately. Then, with a compact Φ, we can train a linear SVM instead of its kernelized version. This is totally feasible and quite efficient even in the big data regime [7]. Therefore, the whole pipeline will actually provide a scalable implementation of a Log-Euclidean SVM, whose cost is reduced from quadratic to linear.

In our work we specifically tackle the aforementioned issue through the following main contributions.

1. We propose a novel compact and explicit feature map to approximate the Log-Euclidean kernel within a probabilistic framework.
2. We provide a rigorous mathematical formulation, proving that the proposed approximation has null bias and bounded variance.

[1] For the sake of precision, let us notice that [22] take advantage of multiple kernel learning in combining several low-level representations and [5] replaces the classical covariance operator with a kernelization.

3. We compare the proposed feature map approximation against alternative approximation schemes, showing the formal superiority of our framework.
4. We experimentally evaluate our method against the very same approximation schemes over six 3D action recognition datasets, confirming with practice our theoretical findings.

The rest of the paper is outlined as follows. In Section 2 we review the most relevant related literature. Section 3 proposes the novel approximation and discusses its foundation. We compare it with alternative paradigms in Section 4. Section 5 draws conclusions and the Appendix A reports all proofs of our theoretical results.

2 Related work

In this Section, we summarize the most relevant works in both covariance-based 3D action recognition and kernels' approximations.

Originally envisaged for image classification and detection tasks, the covariance operator has experienced a growing interest for action recognition, experiencing many different research trends: [9] extends it to the infinite dimensional case, while [10] hierarchically combines it in a temporal pyramid; [22,12] investigate the conceptual analogy with trial-specific kernel matrices and [5] further proposes a new kernelization as to model arbitrary, non-linear relationships conveyed by the raw data. However, those kernel methods usually do not scale up easily to big datasets due to demanding storage and computational costs. As a solution, the exact kernel representation can be replaced by an approximated, more efficient version. In the literature, this is done according to the following mainstream approaches.

(i) The kernel Gram matrix is replaced with a surrogate low-rank version, in order to alleviate both memory and computational costs. Within these methods, [1] applied Cholesky decomposition and [24] exploited Nyström approximation.

(ii) Instead of the exact kernel function k, an explicit feature map Φ is computed, so that the induced linear kernel $\langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle$ approximates $k(\mathbf{x}, \mathbf{y})$. Our work belong to this class of methods.

In this context, Rahimi & Recht [17] exploited the formalism of the Fourier Transform to approximate shift invariant kernels $k(\mathbf{x}, \mathbf{y}) = k(\mathbf{x} - \mathbf{y})$ through an expansion of trigonometric functions. Leveraging on a similar idea, Le et al. [13] sped up the computation by exploiting the Walsh-Hadamard transform, downgrading the running cost of [17] from linear to log-linear with respect to the data dimension. Recently, Kar & Karnick [11] proposed an approximated feature maps for dot product kernels $k(\mathbf{x}, \mathbf{y}) = k(\langle \mathbf{x}, \mathbf{y} \rangle)$ by directly exploiting the MacLaurin expansion of the kernel function.

Instead of considering a generic class of kernels, our work specifically focuses on the log-Euclidean one, approximating it through a novel unbiased estimator which ensures a explicit bound for variance (as only provided by [13]) and resulting in a superior classification performance with respect to [17,13,11].

3 The proposed approximated feature map

In this Section, we present the main theoretical contribution of this work, namely i) a random, explicit feature map $\varPhi$ such that $\langle \varPhi(\mathbf{X}), \varPhi(\mathbf{Y}) \rangle \approx K_{\ell E}(\mathbf{X}, \mathbf{Y})$, ii) the proof of its unbiasedness and iii) a strong theoretical bound on its variance.

Construction of the approximated feature map. In order to construct a ν dimensional feature map $\mathbf{X} \mapsto \varPhi(\mathbf{X}) = [\varPhi_1(\mathbf{X}), \dots, \varPhi_\nu(\mathbf{X})] \in \mathbb{R}^\nu$, for any $d \times d$ SPD matrix $\mathbf{X}$, fix a probability distribution ρ supported over $\mathbb{N}$. Precisely, each component $\varPhi_1, \dots, \varPhi_\nu$ of our ν-dimensional feature map $\varPhi$ is independently computed according to the following algorithm.

```
foreach $j = 1, \dots, \nu$ do
    Sample $n$ according to $\rho$
    Sample the $d^n \times d^n$ matrix $\mathbf{W}$ of independent Gaussian distributed weights
      with zero mean and $\sigma^2/\sqrt{n}$ variance
    Compute $\log(\mathbf{X})^{\otimes n} = \log \mathbf{X} \otimes \cdots \otimes \log \mathbf{X}$, $n$ times.
    Assign
        $\varPhi_j(\mathbf{X}) = \frac{1}{\sigma^{2n}} \sqrt{\frac{\exp(-\sigma^{-2})}{\nu \rho(n) n!}} \operatorname{tr}(\mathbf{W}^\top \log(\mathbf{X})^{\otimes n}).$    (3)
end
```

The genesis of (3) can be explained by inspecting the feature map φ associated to the kernel $K(x,y) = \exp\left(-\frac{1}{2\sigma^2}|x-y|^2\right)$, where $x, y \in \mathbb{R}$ for simplicity. It results $\varphi(x) \propto \left[1, \sqrt{\frac{1}{1!\sigma^2}}x, \sqrt{\frac{1}{2!\sigma^4}}x^2, \sqrt{\frac{1}{3!\sigma^6}}x^3, \dots\right]$. Intuitively, we can say that (3) approximates the infinite dimensional $\varphi(x)$ by randomly selecting one of its components: this is the role played by $n \sim \rho$. In addition, we introduce the log mapping and replace the exponentiation with a Kronecker product. As a consequence, the random weights $\mathbf{W}$ ensure that $\varPhi(\mathbf{X})$ achieves a sound approximation of (1), in terms of unbiasedness and rapidly decreasing variance. In the rest of the Section we discuss the theoretical foundation of our analysis, where all proofs have been moved to Appendix A for convenience.

Unbiased estimation. In order for a statistical estimator to be reliable, we need it to be at least *unbiased*, *i.e.*, its expected value must be equal to the exact function it is approximating. The unbiasedness of the feature map $\varPhi$ of eq. (3) for the Log-Euclidean kernel (1) is proved by the following result.

Theorem 1. *Let ρ be a generic probability distribution over $\mathbb{N}$ and consider $\mathbf{X}$ and $\mathbf{Y}$, two generic SPD matrices such that $\|\log \mathbf{X}\|_F = \|\log \mathbf{Y}\|_F = 1$. Then, $\langle \varPhi(\mathbf{X}), \varPhi(\mathbf{Y}) \rangle$ is an unbiased estimator of (1). That is*

$$\mathbb{E}[\langle \varPhi(\mathbf{X}), \varPhi(\mathbf{Y}) \rangle] = K_{\ell E}(\mathbf{X}, \mathbf{Y}), \tag{4}$$

where the expectation is computed over n and $\mathbf{W}$ which define $\varPhi_j(\mathbf{X})$ as in (3).

Once averaging upon all possible realizations of n sampled from ρ and the Gaussian weights $\mathbf{W}$, Theorem 1 guarantees that the linear kernel $\langle \varPhi(\mathbf{X}), \varPhi(\mathbf{Y}) \rangle$ induced by $\varPhi$ is equal to $K_{\ell E}(\mathbf{X}, \mathbf{Y})$. This formalizes the unbiasedness of our approximation.

On the assumption $\|\log \mathbf{X}\|_F = \|\log \mathbf{Y}\|_F = 1$. Under a practical point of view, this assumption may seem unfavorable, but this is not the case. The reason is provided by equation (2), which is very convenient to compute the logarithm of a SPD matrix. Since in (3), $\varPhi(\mathbf{X})$ is explicitly dependent on $\log \mathbf{X}$, we can simply use (2) and then divide each entry of the obtained matrix by $\|\log \mathbf{X}\|_F$. This is a non-restrictive strategy to satisfy our assumption and actually analogous to require input vectors to have unitary norm, which is very common in machine learning [2].

Low-variance. One can note that, in Theorem 1, even by choosing $\nu = 1$ (a scalar feature map), $\varPhi(\mathbf{X}) = [\varPhi_1(\mathbf{X})] \in \mathbb{R}$ is unbiased for (1). However, since $\varPhi$ is an approximated finite version of the exact infinite feature map associated to (1), one would expect the quality of the approximation to be very bad in the scalar case, and to improve as ν grows larger. This is indeed true, as proved by the following statement.

Theorem 2. *The variance of* $\langle \varPhi(\mathbf{X}), \varPhi(\mathbf{Y}) \rangle$ *as estimator of* (1) *can be explicitly bounded. Precisely,*

$$\mathbb{V}_{n,\mathbf{W}}(K_{\varPhi}(\mathbf{X},\mathbf{Y})) \leq \frac{\mathcal{C}_\rho}{\nu^3} \exp\left(\frac{3-2\sigma^2}{\sigma^4}\right), \tag{5}$$

where $\|\log \mathbf{X}\|_F = \|\log \mathbf{Y}\|_F = 1$ *and the variance is computed over all possible realizations of* $n \sim \rho$ *and* $\mathbf{W}$*, the latter being element-wise sampled from a* $\mathcal{N}(0, \sigma^2/\sqrt{n})$ *distribution. Also,* $\mathcal{C}_\rho \overset{\text{def}}{=} \sum_{n=0}^{\infty} \frac{1}{\rho(n) n!}$*, the series being convergent.*

Let us discuss the bound on the variance provided by Theorem 2. Since the bandwidth σ of the kernel function (1) we want to approximate is fixed, the term $\exp\left(\frac{3-2\sigma^2}{\sigma^4}\right)$ can be left out from our analysis. The bound in (5) is linear in $\mathcal{C}_\rho$ and inversely cubic in ν. When ν grows, the increased dimensionality of our feature map $\varPhi$ makes the variance rapidly vanishing, ensuring that the *approximated kernel* $K_{\varPhi}(\mathbf{X},\mathbf{Y}) = \langle \varPhi(\mathbf{X}), \varPhi(\mathbf{Y}) \rangle$ converges to the target one, that is $K_{\ell E}$. Such trend may be damaged by big values of $\mathcal{C}_\rho$. Since the latter depends on the distribution ρ, let us fix it to be the geometric distribution $\mathcal{G}(\theta)$ with parameter $0 \leq \theta < 1$. This yields

$$\mathcal{C}_\rho \propto \sum_{n=0}^{\infty} \frac{1}{(1-\theta)^n \cdot n!} = \exp\left(\frac{1}{1-\theta}\right). \tag{6}$$

There is a tradeoff between a low variance (*i.e.*, $\mathcal{C}_\rho$ small) and a reduced computational cost for $\varPhi$ (*i.e.*, n small). Indeed, choosing $\theta \approx 1$ makes $\mathcal{C}_\rho$ big in (6). In this case, the integer n sampled from $\rho = \mathcal{G}(\theta)$ is small with great probability: this leads to a reduced number of Kronecker products to be computed in $\log(\mathbf{X})^{\otimes n}$. Conversely, when $\theta \approx 0$, despite n and the related computational cost of $\log(\mathbf{X})^{\otimes n}$ are likely to grow, $\mathcal{C}_\rho$ is small, ensuring a low variance for the estimator.

As a final theoretical result, Theorems 1 and 2 immediately yield that

$$\mathbb{P}[|K_{\Phi}(\mathbf{X},\mathbf{Y})-K_{\ell E}(\mathbf{X},\mathbf{Y})|\geq\epsilon]\leq\frac{\mathcal{C}_{\rho}}{\nu^{3}\epsilon^{2}}\exp\left(\frac{3-2\sigma^{2}}{\sigma^{4}}\right) \tag{7}$$

for every pairs of unitary Frobenius normed SPD matrices $\mathbf{X}, \mathbf{Y}$ and $\epsilon > 0$, as a straightforward implication of the Chebyshev inequality. This ensures that K_{Φ} differs in module from $K_{\ell E}$ by more than ϵ with a (low) probability $\mathbb{P}$, which is inversely cubic and quadratic in ν and ϵ, respectively.

Final remarks. To sum up, we have presented a constructive algorithm to compute a ν-dimensional feature map Φ whose induced linear kernel is an unbiased estimator of the log-Euclidean one. Additionally, we ensure an explicit bound on the variance which rapidly vanishes as ν grows (inversely cubic decrease). This implies that $\langle\Phi(\mathbf{X}),\Phi(\mathbf{Y})\rangle$ and $K_{\ell E}(\mathbf{X},\mathbf{Y})$ are equal with very high probability, even at low ν values. This implements a Log-Euclidean kernel in a scalable manner, downgrading the quadratic cost of computing $K_{\ell E}(\mathbf{X},\mathbf{Y})$ for every $\mathbf{X},\mathbf{Y}$ into the linear cost of evaluating the feature map $\Phi(\mathbf{X})$ as in (3) for every $\mathbf{X}$. Practically, this achieve a linear implementation of the log-Euclidean SVM.

4 Results

In this Section, we compare our proposed approximated feature map versus the alternative ones by Rahimi & Recht [17], Kar & Karnick [11] and Le et al. [13] (see Section 2).

Theoretical Comparison. Let us notice that all approaches [17,11,13] are applicable also to the log-Euclidean kernel (1). Indeed, [17,13] includes our case of study since $K_{\ell E}(\mathbf{X},\mathbf{Y}) = k(\log\mathbf{X} - \log\mathbf{Y})$ is logarithmic shift invariant. At the same time, thanks to the assumption $\|\log\mathbf{X}\|_F = \|\log\mathbf{Y}\|_F = 1$ as in Theorem 1, we obtain $K_{\ell E}(\mathbf{X},\mathbf{Y}) = k(\langle\log\mathbf{X},\log\mathbf{Y}\rangle)$ (see (13) in Appendix A), thus satisfying the hypothesis of Kar & Karnick [11].

As we proved in Theorem 1, all works [17,11,13] can also guarantee an unbiased estimation for the exact kernel function.

Actually, what makes our approach superior is the explicit bound on the variance (see Table 1). Indeed, [17,11] are totally lacking in this respect. Moreover, despite an analogous bound is provided in [13, Theorem 4], it only ensures a $O(1/\nu)$ convergence rate for the variance with respect to the feature dimensionality ν. Differently, we can guarantee a $O(1/\nu^3)$ trend. This implies that, *we achieve a better approximation of the kernel with a lower dimensional feature representation*, which ease the training of the linear SVM [7].

Experimental Comparison. We reported here the experimental comparison on 3D action recognition between our proposed approximation and the paradigms of [17,11,13].

Datasets. For the experiments, we considered UTKinect [23], Florence3D [19], MSR-Action-Pairs (MSR-*pairs*) [16], MSR-Action3D [14], [3], HDM-05 [15] and MSRC-Kinect12 [8] datasets.

proposed	Rahimi & Recht [17]	Kar & Karninck [11]	Le et al. [13]
$O(1/\nu^3)$	`missing`	`missing`	$O(1/\nu)$

Table 1. Theoretical comparison between explicit bounds on variance between the proposed approximation and [17,11,13]: the quicker the decrease, the better the bound. Here, ν denotes the dimensionality of the approximated feature vector.

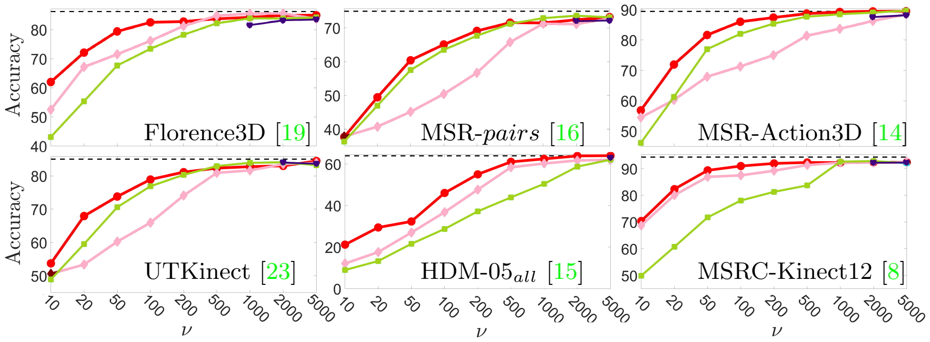

Fig. 1. Experimental comparison of our approximation (red curves) against the schemes ofr Rahimi & Recht [17] (pink curves), Kar & Karnick [11] (green curves) and Le et al. [13] (blue curves). Best viewed in colors.

For each, we follow the usual training and testing splits proposed in the literature. For Florence3D and UTKinect, we use the protocols of [20]. For MSR-Action3D, we adopte the splits originally proposed by [14]. On MSRC-Kinect12, once highly corrupted action instances are removed as in [10], training is performed on odd-index subject, while testing on the even-index ones. On HDM-05, the training exploited all instances of "`bd`" and "`mm`" subjects, being "`bk`", "`dg`" and "`tr`" left out for testing [22], using the 65 action classes protocol of [6].

Data preprocessing. As a common pre-processing step, we normalize the data by computing the relative displacements of all joints $x-y-z$ coordinates and the ones of the hip (central) joint, for each timestamp.

Results. Figure 1 reports the quantitative performance while varying ν in the range 10, 20, 50, 100, 200, 500, 1000, 2000, 5000. When comparing with [13], since the data input size must be a multiple of a power of 2, we zero-padded our vectorized representation to match 4096 and (whenever possible) 2048 and 1024 input dimensionality. These cases are then compared with the results related to $\nu = 5000, 2000, 1000$ for RGW and [17,11], respectively. Since all approaches have a random component, we performed ten repetitions for each method and dimensionality setup, averaging the scored classification performances obtained through a linear SVM with $C = 10$. We employ the publicly available codes for [17,11,13]. Finally, we also report the classification performance with the exact method obtained by feeding an SVM with the log-Euclidean kernel whose bandwidth σ is chosen via cross validation.

Discussion. For large ν values, all methods are able to reproduce the performance of the log-Euclidean kernel (black dotted line). Still, in almost all the cases, our approximation is able to outperform the competitors: for instance, we gapped Rahimi and Recht on both MSR-Pairs and MSR-Action3D, while Kar & Karnick scored a much lower performance on HDM-05 and Florence3D. If comparing to Le et al., the performance is actually closer, but this happens for all the methods which are able to cope the performance of the Log-Euclidean kernel with $\nu \geq 2000, 5000$. Precisely, the true superiority of our approach is evident in the case of a small ν value (ν = 10, 20, 50). Indeed, our approximation always provides a much rapid growing accuracy (MSR-Action3D, Florence3D and UTKinect), with only a few cases where the gap is thinner (Kar & Karnick [11] on MSR-*pairs* and Rahimi & Recth [17] on MSRC-Kinect 12). Therefore, our approach ensures a more descriptive and compact representation, providing a superior classification performance.

5 Conclusions

In this work we propose a novel scalable implementation of a Log-Euclidean SVM to perform proficient classification of SPD (covariance) matrices. We achieve a linear complexity by providing an explicit random feature map whose induced linear kernel is an unbiased estimator of the exact kernel function.

Our approach proved to be more effective than alternative approximations [17,11,13], both theoretically and experimentally. Theoretically, we achieve an explicit bound on the variance on the estimator (such result is totally absent in [17,11]), which is decreasing with inversely cubic pace versus the inverse linear of [13]. Experimentally, through a broad evaluation, we assess the superiority of our representation which is able to provide a superior classification performance at a lower dimensionality.

A Proofs of all theoretical results

In this Appendix we report the formal proofs for both the unbiased approximation (Theorem 1) and the related rapidly decreasing variance (Theorem 2).

Proof of Theorem 1. Use the definition of (3) and the linearity of the expectation. We get that $\mathbb{E}_{n,\mathbf{W}}\left[\langle \Phi(\mathbf{X}), \Phi(\mathbf{Y})\rangle\right]$ equals to

$$\mathbb{E}_n\left[\frac{1}{\sigma^{4n}}\frac{\exp(-\sigma^{-2})}{\rho(n)n!}\mathbb{E}_{\mathbf{W}}\left[\operatorname{tr}\left(\mathbf{W}^\top \log(\mathbf{X})^{\otimes n}\right)\operatorname{tr}\left(\mathbf{W}^\top \log(\mathbf{Y})^{\otimes n}\right)\right]\right], \tag{8}$$

by simply noticing that the dependence with respect to $\mathbf{W}$ involves the terms inside the trace operators only. Let us focus on the term $\operatorname{tr}\left(\mathbf{W}^\top \log(\mathbf{X})^{\otimes n}\right)$. We can expand

$$\operatorname{tr}\left(\mathbf{W}^\top \log(\mathbf{X})^{\otimes n}\right) = \sum_{i_1,\dots,i_{2n}=1}^{d} w_{i_1,\dots,i_{2n}} \log(\mathbf{X})_{i_1,i_2}\cdots\log(\mathbf{X})_{i_{2n-1},i_{2n}} \tag{9}$$

by using the definition of $\log(\mathbf{X})^{\otimes n}$ and the properties of the trace operator. In equation (9), we replace the random coefficient $w_{i_1,\ldots,i_{2n}}$ with $u^{(1)}_{i_1,i_2},\ldots,u^{(n)}_{i_{2n-1},i_{2n}}$ independent and identically distributed (i.i.d.) according to a $\mathcal{N}(0,\sigma^2)$ distribution. We can notice that (9) can be rewritten as

$$\operatorname{tr}\left(\mathbf{W}^\top \log(\mathbf{X})^{\otimes n}\right) = \prod_{\alpha=1}^{n} \sum_{i,j=1}^{d} u^{(\alpha)}_{i,j} \log(\mathbf{X})_{ij}. \tag{10}$$

Making use of (10) in (8), we can rewrite $\mathbb{E}_{n,\mathbf{W}}\left[K_\Phi(\mathbf{X},\mathbf{Y})\right]$ as

$$\mathbb{E}_n\left[\frac{1}{\sigma^{4n}}\frac{\exp(-\sigma^{-2})}{\rho(n)n!}\mathbb{E}_{\mathbf{W}}\left[\left(\sum_{i,j=1}^{d} u^{(1)}_{i,j}\log(\mathbf{X})_{ij}\right)\left(\sum_{h,k=1}^{d} u^{(1)}_{h,k}\log(\mathbf{Y})_{hk}\right)\right]^n\right] \tag{11}$$

by also considering the independence of $u^{(\alpha)}_{i,j}$ are independent. By furthermore using the fact that $\mathbb{E}_{\mathbf{W}}\left[u^{(1)}_{i,j}u^{(1)}_{h,k}\right] = 0$ if $i \neq h$ and $j \neq k$ and the formula for the variance of a Gaussian distribution, we get

$$\mathbb{E}_{n,\mathbf{W}}\left[K_\Phi(\mathbf{X},\mathbf{Y})\right] = \mathbb{E}_n\left[\frac{1}{\sigma^{4n}}\frac{\exp(-\sigma^{-2})}{\rho(n)n!}\sigma^{2n}\left(\langle\log(\mathbf{X}),\log(\mathbf{Y})\rangle_F\right)^n\right], \tag{12}$$

by introducing the Frobenius inner product $\langle\mathbf{A},\mathbf{B}\rangle_F = \sum_{i,j=1}^d \mathbf{A}_{ij}\mathbf{B}_{ij}$ between matrices $\mathbf{A}$ and $\mathbf{B}$. By expanding the expectation over ρ, (12) becomes

$$\begin{aligned}\mathbb{E}_{n,\mathbf{W}}\left[K_\Phi(\mathbf{X},\mathbf{Y})\right] &= \sum_{n=0}^{\infty}\rho(n)\frac{1}{\sigma^{2n}}\frac{\exp(-\sigma^{-2})}{\rho(n)n!}\left(\langle\log(\mathbf{X}),\log(\mathbf{Y})\rangle_F\right)^n \\ &= \exp\left(-\frac{1}{\sigma^2}\right)\sum_{n=0}^{\infty}\left(\frac{\langle\log(\mathbf{X}),\log(\mathbf{Y})\rangle_F}{\sigma^2}\right)^n\frac{1}{n!}.\end{aligned} \tag{13}$$

The thesis easily comes from (13) by using the Taylor expansion for the exponential function and the assumption $\|\log(\mathbf{X})\|_F = \|\log(\mathbf{Y})\|_F = 1$. □

Proof of Theorem 2. Due to the independence of the components in Φ, by definition of inner product we get $\mathbb{V}_{n,\mathbf{W}}\left[\langle\Phi(\mathbf{X}),\Phi(\mathbf{Y})\rangle\right] = \nu\mathbb{V}_{n,\mathbf{W}}\left[\Phi_1(\mathbf{X})\Phi_1(\mathbf{Y})\right]$. But then $\mathbb{V}_{n,\mathbf{W}}\left[\langle\Phi(\mathbf{X}),\Phi(\mathbf{Y})\rangle\right] \leq \nu\mathbb{E}_{n,\mathbf{W}}\left[\Phi_1(\mathbf{X})^2\Phi_1(\mathbf{Y})^2\right]$ by definition of variance. Taking advantage of (3), yields to the equality between $\mathbb{V}_{n,\mathbf{W}}\left[K_\Phi(\mathbf{X},\mathbf{Y})\right]$ and

$$\frac{1}{\nu^3}\mathbb{E}_{n,\mathbf{U}}\left[\frac{1}{\sigma^{8n}}\frac{\exp(-2\sigma^{-2})}{(\rho(n)n!)^2}\prod_{\alpha=1}^{n}\left(\sum_{i,j=1}^{d}u^{(\alpha)}_{i,j}\log(\mathbf{X})_{ij}\right)^2\left(\sum_{h,k=1}^{d}u^{(\alpha)}_{h,k}\log(\mathbf{Y})_{hk}\right)^2\right], \tag{14}$$

where $u^{(1)}_{i_1,i_2},\ldots,u^{(n)}_{i_{2n-1},i_{2n}}$ are i.i.d. from $\mathcal{N}(0,\sigma^2)$ distribution used to re-parametrize the original weights $\mathbf{W}$. Exploit the independence of $u^{(\alpha)}_{ij}$ to rewrite (14) as

$$\frac{1}{\nu^3}\mathbb{E}_n\left[\frac{1}{\sigma^{8n}}\frac{\exp(-2\sigma^{-2})}{(\rho(n)n!)^2}\mathbb{E}_{\mathbf{U}}\left[\left(\sum_{i,j=1}^{d}u^{(1)}_{i,j}\log(\mathbf{X})_{ij}\right)^2\left(\sum_{h,k=1}^{d}u^{(1)}_{h,k}\log(\mathbf{Y})_{hk}\right)^2\right]^n\right]. \tag{15}$$

By exploiting the zero correlation of the weights in $\mathbf{U}$ and the formula $\mathbb{E}[(\mathcal{N}(0,\sigma^2))^4] = 3\sigma^4$ [4]. Thus,

$$\mathbb{V}_{n,\mathbf{W}}\left[K_{\Phi}(\mathbf{X},\mathbf{Y})\right] \leq \frac{1}{\nu^3}\mathbb{E}_n\left[\frac{1}{\sigma^{8n}}\frac{\exp(-2\sigma^{-2})}{(\rho(n)n!)^2}3^n\sigma^{4n}\left(\sum_{i,j=1}^{d}\log(\mathbf{X})_{ij}^2\log(\mathbf{Y})_{ij}^2\right)^n\right]. \tag{16}$$

Since $\sum_{i,j=1}^{d}\log(\mathbf{X})_{ij}^2\log(\mathbf{Y})_{ij}^2 \leq \left(\sum_{i,j=1}^{d}\log(\mathbf{X})_{ij}^2\right)\left(\sum_{i,j=1}^{d}\log(\mathbf{Y})_{ij}^2\right) = 1$ due to the assumption of unitary Frobenius norm for both $\log\mathbf{X}$ and $\log\mathbf{Y}$, we get

$$\mathbb{V}_{n,\mathbf{W}}\left[K_{\Phi}(\mathbf{X},\mathbf{Y})\right] \leq \frac{1}{\nu^3}\mathbb{E}_n\left[\frac{1}{\sigma^{8n}}\frac{\exp(-2\sigma^{-2})}{(\rho(n)n!)^2}3^n\sigma^{4n}\right]. \tag{17}$$

We can now expand the expectation over ρ in (17), achieving

$$\mathbb{V}_{n,\mathbf{W}}\left[K_{\Phi}(\mathbf{X},\mathbf{Y})\right] \leq \frac{\exp(-2\sigma^{-2})}{\nu^3}\sum_{n=0}^{\infty}\left(\frac{3}{\sigma^4}\right)^n\frac{1}{n!}\sum_{n=0}^{\infty}\frac{1}{\rho(n)n!}, \tag{18}$$

since the series of the products is less than the product of the series, provided that both converge. This is actually true since, by exploiting the McLaurin expansion for the exponential function, we easily get $\sum_{n=0}^{\infty}\left(\frac{3}{\sigma^4}\right)^n\frac{1}{n!} = \exp\left(\frac{3}{\sigma^4}\right)$. On the other hand, since ρ is a probability distribution, it must be $\lim_{n\to\infty}\frac{\rho(n+1)}{\rho(n)} = L$ where $0 < L \leq 1$, being $\mathbb{N}$ the support of ρ and due to $\sum_{n=0}^{\infty}\rho(n) = 1$. Then, since $\lim_{n\to\infty}\frac{\rho(n)}{\rho(n+1)} = \frac{1}{L} < \infty$ and $\lim_{n\to\infty}\frac{1}{n+1} = 0$, by the ration criterion for positive-terms series [18], there must exist a constant $\mathcal{C}_\rho > 0$ such that

$$\sum_{n=0}^{\infty}\frac{1}{\rho(n)n!} = \mathcal{C}_\rho. \tag{19}$$

Therefore, by combining (19) in (18), we obtain

$$\mathbb{V}_{n,\mathbf{W}}\left[K_{\Phi}(\mathbf{X},\mathbf{Y})\right] \leq \frac{\exp(-2\sigma^{-2})}{\nu^3}\exp\left(\frac{3}{\sigma^4}\right)\mathcal{C}_\rho = \frac{\mathcal{C}_\rho}{\nu^3}\exp\left(\frac{3-2\sigma^2}{\sigma^4}\right),$$

which is the thesis. □

References

1. Bach, F.R., Jordan, M.I.: Predictive low-rank decomposition for kernel methods. In: ICML (2005)
2. Bishop, C.M.: Pattern Recognition and Machine Learning - Information Science and Statistics. Springer-Verlag New York, Inc. (2006)
3. Bloom, V., Makris, D., Argyriou, V.: G3D: A gaming action dataset and real time action recognition evaluation framework. In: CVPR (2012)

4. Casella, G., Berger, R.: Statistical Inference. Duxbury advanced series in statistics and decision sciences, Thomson Learning (2002)
5. Cavazza, J., Zunino, A., San Biagio, M., Murino, V.: Kernelized covariance for action recognition. In: ICPR (2016)
6. Cho, K., Chen, X.: Classifying and visualizing motion capture sequences using deep neural networks. CoRR 1306.3874 (2014)
7. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: LIBLINEAR: A library for large linear classification. JMLR 9, 1871–1874 (2008)
8. Fothergill, S., Mentis, H.M., Kohli, P., Nowozin, S.: Instructing people for training gestural interactive systems. In: ACM-CHI (2012)
9. Harandi, M., Salzmann, M., Porikli, F.: Bregman divergences for infinite dimensional covariance matrices. In: CVPR (2014)
10. Hussein, M., Torki, M., Gowayyed, M., El-Saban., M.: Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. IJCAI (2013)
11. Kar, P., Karnick, H.: Random feature maps for dot product kernels. In: AISTATS (2012)
12. Koniusz, P., Cherian, A., Porikli, F.: Tensor representation via kernel linearization for action recognition from 3d skeletons. In: ECCV (2016)
13. Le, Q., Sarlos, T., Smola, A.: Fastfood - approximating kernel expansion in loglinear time. In: ICML (2013)
14. Li, W., Zhang, Z., Liu, Z.: Action recognition based on a bag of 3d points. In: CVPR workshop (2010)
15. Müller, M., Röder, T., Clausen, M., Eberhardt, B., Krüger, B., Weber, A.: HDM-05 doc. In: Tech. Rep. (2007)
16. Oreifej, O., Liu., Z.: HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences. In: CVPR (2013)
17. Rahimi, A., Recth, B.: Random features for large-scale kernel machines. In: NIPS (2007)
18. Rudin, W.: Real and Complex Analysis, 3rd Ed. McGraw-Hill, Inc., New York, NY, USA (1987)
19. Seidenari, L., Varano, V., Berretti, S., Bimbo, A.D., Pala, P.: Recognizing actions from depth cameras as weakly aligned multi-part bag-of-poses. In: CVPR workshops (2013)
20. Vemulapalli, R., Arrate, F., Chellappa, R.: Human action recognition by representing 3d skeletons as points in a lie group. In: CVPR (June 2014)
21. Vrigkas, M., Nikou, C., Kakadiaris, I.A.: A review of human activity recognition methods. Front. robot. AI 2, 28 (2015)
22. Wang, L., Zhang, J., Zhou, L., Tang, C., Li, W.: Beyond covariance: Feature representation with nonlinear kernel matrices. In: ICCV (2015)
23. Xia, L., Chen, C.C., Aggarwal, J.: View invariant human action recognition using histograms of 3D joints. In: CVPR workshops (2012)
24. Zhang, K., Tsang, I.W., Kwok, J.T.: Improved Nyström low-rank approximation. In: ICML (2008)