Appendix

This appendix provides both theoretical and experimental material and is organized as follows: Appendix A presents a classical result, allowing us to characterize the RKHS of the graph kernels we introduce. Appendix B provides additional experimental details that are useful to reproduce our results and additional experimental results. Then, Appendix C explains how to accelerate the computation of GCKN when using walks instead of paths (at the cost of lower expressiveness), and Appendix D presents a proof of Theorem 1 on the expressiveness of WL and walk kernels.

A. Useful Result about RKHSs

The following result characterizes the RKHS of a kernel function when an explicit mapping to a Hilbert space is available. It may be found in classical textbooks (see, *e.g.*, Saitoh, 1997, §2.1).

Theorem 2. *Let $\Phi : \mathcal{X} \to \mathcal{F}$ be a mapping from a data space $\mathcal{X}$ to a Hilbert space $\mathcal{F}$, and let $K(x, x') := \langle \Phi(x), \psi(x') \rangle_{\mathcal{F}}$ for x, x' in $\mathcal{X}$. Consider the Hilbert space*

$$\mathcal{H} := \{ f_z \,;\, z \in \mathcal{F} \} \quad \textit{s.t.} \quad f_z : x \mapsto \langle z, \Phi(x) \rangle_{\mathcal{F}},$$

endowed with the norm

$$\|f\|_{\mathcal{H}}^2 := \inf_{z \in \mathcal{F}} \left\{ \|z\|_{\mathcal{F}}^2 \quad \textit{s.t.} \quad f = f_z \right\}.$$

Then, $\mathcal{H}$ is the reproducing kernel Hilbert space associated to kernel K.

B. Details on Experimental Setup and Additional Experiments

In this section, we provide additional details and more experimental results. In Section B.1, we provide additional experimental details; in Section B.2, we perform a hyperparameter study for unsupervised GCKN on three datasets, showing that our approach is relatively robust to the choice of hyperparameters. In particular, the number of filters controls the quality of Nyström's kernel approximation: more filters means a better approximation and better results, at the cost of more computation. This is in contrast with a traditional (supervised) GNN, where more filters may lead to overfitting. Finally, Section B.3 provides motif discovery results.

B.1. Experimental Setup and Reproducibility

Hyperparameter search grids. In our experiments for supervised models, we use an Adam optimizer (Kingma & Ba, 2015) for at most 350 epochs with an initial learning rate equal to 0.01 and halved every 50 epochs with a batch size fixed to 32 throughout all datasets; the number of epochs is selected using cross validation following Xu et al. (2019). The full hyperparameter search range is given in Table 3 for both unsupervised and supervised models on all tasks. Note that we include some large values (1.5 and 2.0) for σ to simulate the linear kernel as we discussed in Section 3.3. In fact, the function $\sigma_1(x) = e^{\alpha(x-1)}$ defined in (12) is upper bounded by $e^{-\alpha} + (1 - e^{-\alpha})x$ and lower bounded by $1 + \alpha(x - 1)$ by its convexity at 0 and 1. Their difference is increasing with α and converges to zero when α tends to 0. Hence, when α is small, σ_1 behaves as an affine kernel with a small slope.

Computing infrastructure. Experiments for unsupervised models were conducted by using a shared CPU cluster composed of 2 Intel Xeon E5-2470v2 @2.4GHz CPUs with 16 cores and 192GB of RAM. Supervised models were trained by using a shared GPU cluster, in large parts built with Nvidia gamer cards (Titan X, GTX1080TI). About 20 of these CPUs and 10 of these GPUs were used simultaneously to perform the experiments of this paper.

B.2. Hyperparameter Study

We show here that both unsupervised and supervised models are generally robust to different hyperparameters, including path size k_1, bandwidth parameter σ, regularization parameter λ and their performance grows increasingly with the number

Table 3. Hyperparameter search range

Hyperparameter	Search range
σ ($\alpha = 1/\sigma^2$)	$[0.3; 0.4; 0.5; 0.6; 1.0; 1.5; 2.0]$
local/global pooling	[sum, mean, max]
path size k_1	integers between 2 and 12
number of filters (unsup)	[32; 128; 512; 1024]
number of filters (sup)	[32; 64] and 256 for ENZYMES
λ (unsup)	$1/n \times$ np.logspace(-3, 4, 60)
λ (sup)	[0.01; 0.001; 0.0001; 1e-05; 1e-06; 1e-07]

of filters q. The accuracies for NCI1, PROTEINS and IMDBMULTI are given in Figure 4, by varying respectively the number of filters, the path size, the bandwidth parameter and regularization parameter when fixing other parameters which give the best accuracy. Supervised models generally require fewer number of filters to achieve similar performance to its unsupervised counterpart. In particular on the NCI1 dataset, the supervised GCKN outperforms its unsupervised counterpart by a significant margin when using a small number of filters.

B.3. Model Interpretation

Implementation details. We use a similar experimental setting as Ying et al. (2019) to train a supervised GCKN-subtree model on Mutagenicity dataset, consisting of 4337 molecule graphs labeled according to their mutagenic effect. Specifically, we use the same split for train and validation set and train a GCKN-subtree model with $k_1 = 3$, which is similar to a 3-layer GNN model. The number of filters is fixed to 20, the same as Ying et al. (2019). The bandwidth parameter σ is fixed to 0.4, local and global pooling are fixed to mean pooling, the regularization parameter λ is fixed to 1e-05. We use an Adam optimizer with initial learning equal to 0.01 and halved every 50 epochs, the same as previously. The accuracy of the trained model is assured to be more than 80% on the test set as Ying et al. (2019). Then we use the procedure described in Section 4 to interpret our trained model. We use an LBFGS optimizer and fixed μ to 0.01. The final subgraph for each given graph is obtained by extracting the maximal connected component formed by the selected paths. A contribution score for each edge can also be obtained by gathering the weights M of all the selected paths that pass through this edge.

More results. More motifs extracted by GCKN are shown in Figure 5 for the Mutagenicity dataset. We recovered some benzene ring or polycyclic aromatic groups which are known to be mutagenic. We also found some groups whose mutagenicity is not known, such as polyphenylene sulfide in the fourth subgraph and 2-chloroethyl- in the last subgraph.

C. Fast Computation of GCKN with Walks

Here we discuss an efficient computational variant using walk kernel instead of path kernel, at the cost of losing some expressive power. Let us consider a relaxed walk kernel by analogy to (8) with

$$\kappa_{\text{base}}^{(k)}(u, u') = \sum_{p \in \mathcal{W}_k(G,u)} \sum_{p' \in \mathcal{W}_k(G',u')} \kappa_1(\varphi_0(p), \varphi_0'(p')), \tag{19}$$

using walks instead of paths and with κ_1 the Gaussian kernel defined in (9). As Gaussian kernel can be decomposed as a product of the Gaussian kernel on pair of nodes at each position

$$\kappa_1(\varphi_0(p), \varphi_0'(p')) = \prod_{j=1}^{k} \kappa_1(\varphi_0(p_j), \varphi_0'(p_j')),$$

We can obtain similar recursive relation as for the original walk kernel in Lemma 2

$$\kappa_{\text{base}}^{(k)}(u, u') = \kappa_1(\varphi_0(u), \varphi_0'(u')) \sum_{v \in \mathcal{N}(u)} \sum_{v' \in \mathcal{N}(u')} \kappa_{\text{base}}^{(k-1)}(v, v'). \tag{20}$$

After applying the Nyström method, the approximate feature map in (13) becomes

$$\psi_1(u) = \sigma_1(Z^\top Z)^{-\frac{1}{2}} c_k(u),$$

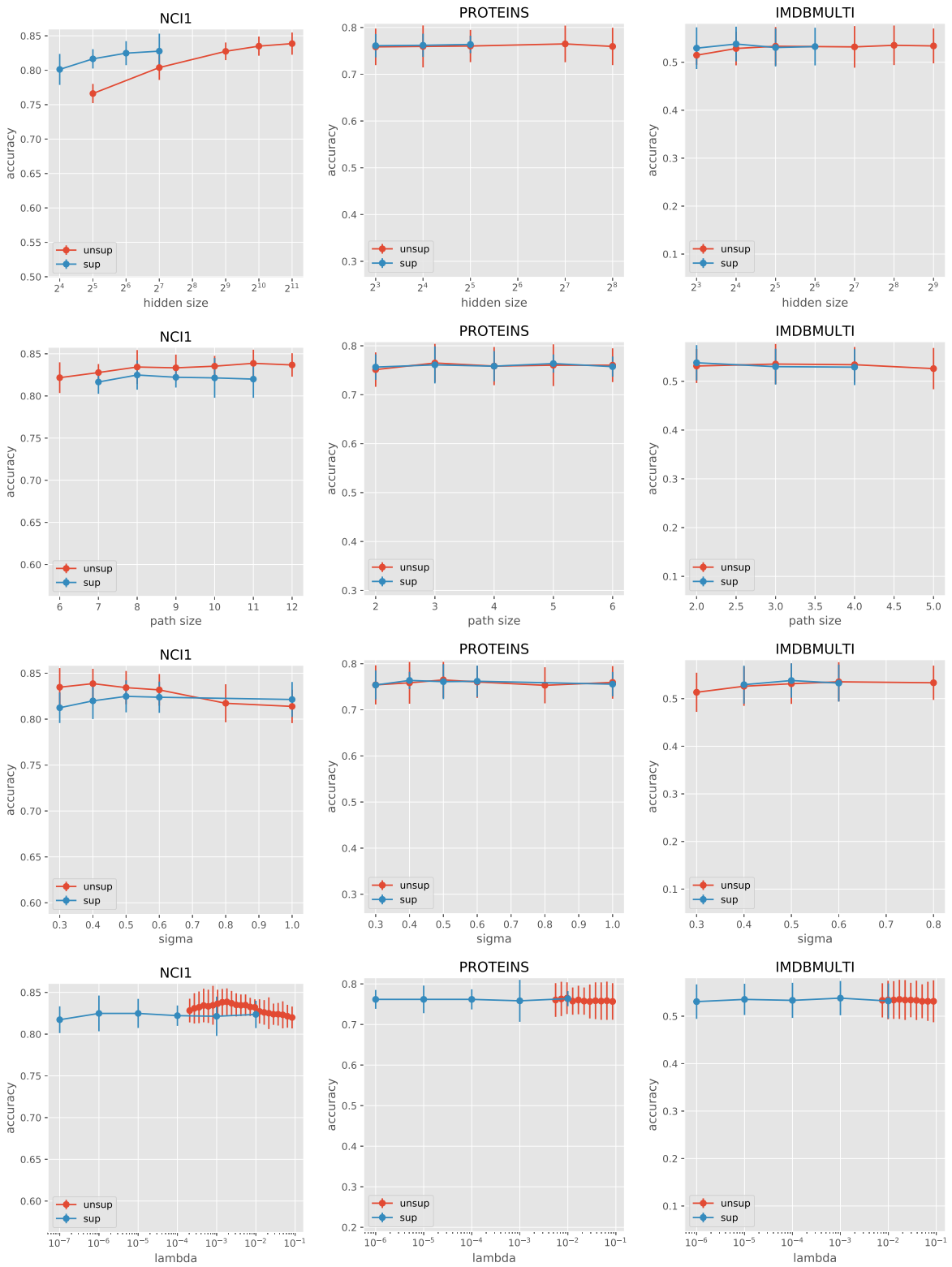

Figure 4. Hyperparamter study: sensibility to different hyperparameters for unsupervised and supervised GCKN-subtree models. The row from top to bottom respectively corresponds to number of filters q_1, path size k_1, bandwidth parameter σ and regularization parameter λ. The column from left to right corresponds to different datasets: NC11, PROTEINS and IMDBMULTI.

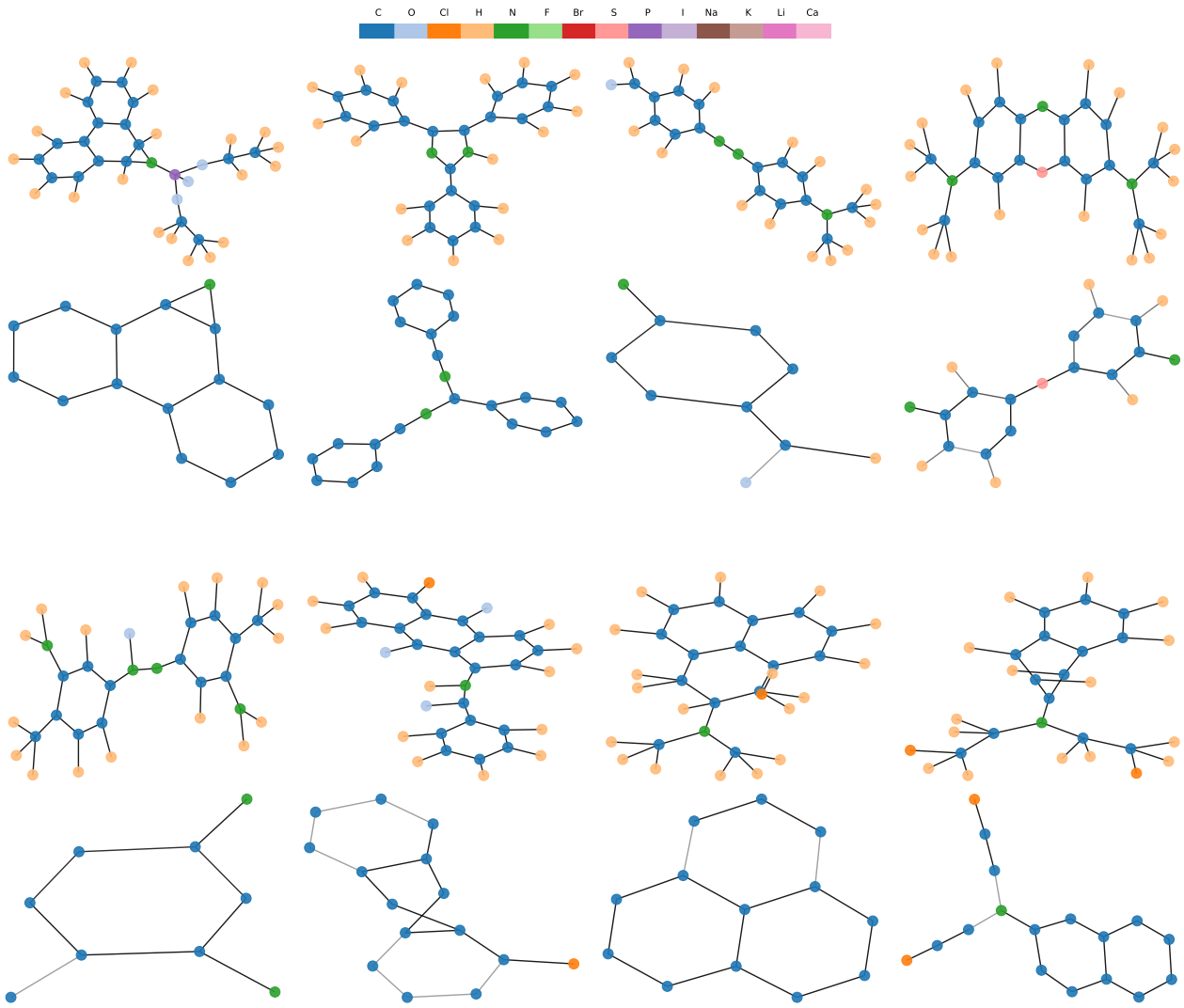

Figure 5. More motifs extracted by GCKN on Mutagenicity dataset. First and third rows are original graphs; second and fourth rows are corresponding motifs. Some benzene ring or polycyclic aromatic groups are identified, which are known to be mutagenic. In addition, Some chemical groups whose mutagenicity is not known are also identified, such as polyphenylene sulfide in the fourth subgraph and 2-chloroethyl- in the last subgraph.

where for any $0 \leq j \leq k$, $c_j(u) := \sum_{p \in \mathcal{W}_j(G,u)} \sigma_1(Z_j^\top \psi_0(p))$ and Z_j in $\mathbb{R}^{q_0(j+1) \times q_1}$ denotes the matrix consisting of the $j+1$ last columns of q_1 anchor points. Using the above recursive relation (20) and similar arguments in *e.g.* (Chen et al., 2019b), we can show c_j obeys the following recursive relation

$$c_j(u) = b_j(u) \odot \sum_{v \in \mathcal{N}(u)} c_{j-1}(v), \ 1 \leq j \leq k, \tag{21}$$

where $\odot$ denotes the element-wise product and $b_j(u)$ is a vector in $\mathbb{R}^{q_1}$ whose entry i in $\{1, \dots, q_1\}$ is $\kappa_1(u, z_i^{(k+1-j)})$ and $z_i^{(k+1-j)}$ denotes the $k+1-j$-th column vector of z_i in $\mathbb{R}^{q_0}$. In practice, $\sum_{v \in \mathcal{N}(u)} c_{j-1}(v)$ can be computed efficiently by multiplying the adjacency matrix with the $|\mathcal{V}|$-dimensional vector with entries $c_{j-1}(v)$ for $v \in \mathcal{V}$.

D. Proof of Theorem 1

Before presenting and proving the link between the WL subtree kernel and the walk kernel, we start by reminding and showing some useful results about the WL subtree kernel and the walk kernel.

D.1. Useful results for the WL subtree kernel

We first recall a recursive relation of the WL subtree kernel, given in the Theorem 8 of Shervashidze et al. (2011). Let us denote by $\mathcal{M}(u, u')$ the set of exact matchings of subsets of the neighbors of u and u', formally given by

$$\mathcal{M}(u,u') = \Big\{ R \subseteq \mathcal{N}(u) \times \mathcal{N}(u') \Big| \, |R| = |\mathcal{N}(u)| = |\mathcal{N}(u')| \wedge \\ (\forall (v,v'), (w,w') \in R : u = w \Leftrightarrow u' = w') \wedge (\forall (u,u') \in R : a(u) = a'(u')) \Big\}. \tag{22}$$

Then we have the following recursive relation for $\kappa_{\text{subtree}}^{(k)}(u,u') := \delta(a_k(u), a'_k(u'))$

$$\kappa_{\text{subtree}}^{(k+1)}(u,u') = \begin{cases} \kappa_{\text{subtree}}^{(k)}(u,u') \max\limits_{R \in \mathcal{M}(u,u')} \prod\limits_{(v,v') \in R} \kappa_{\text{subtree}}^{(k)}(v,v'), & \text{if } \mathcal{M}(u,u') \neq \emptyset, \\ 0, & \text{otherwise.} \end{cases} \tag{23}$$

We can further simply the above recursion using the following Lemma

Lemma 1. *If $\mathcal{M}(u,u') \neq \emptyset$, we have*

$$\kappa_{\textit{subtree}}^{(k+1)}(u,u') = \delta(a(u), a'(u')) \max_{R \in \mathcal{M}(u,u')} \prod_{(v,v') \in R} \kappa_{\textit{subtree}}^{(k)}(v,v').$$

Proof. We prove this by induction on $k \geq 0$. For $k = 0$, this is true by the definition of $\kappa_{\text{subtree}}^{(0)}$. For $k \geq 1$, we suppose that $\kappa_{\text{subtree}}^{(k)}(u,u') = \delta(a(u), a'(u')) \max_{R \in \mathcal{M}(u,u')} \prod_{(v,v') \in R} \kappa_{\text{subtree}}^{(k-1)}(v,v')$. We have

$$\begin{aligned} \kappa_{\text{subtree}}^{(k+1)}(u,u') &= \kappa_{\text{subtree}}^{(k)}(u,u') \max_{R \in \mathcal{M}(u,u')} \prod_{(v,v') \in R} \kappa_{\text{subtree}}^{(k)}(v,v') \\ &= \delta(a(u), a'(u')) \max_{R \in \mathcal{M}(u,u')} \prod_{(v,v') \in R} \kappa_{\text{subtree}}^{(k-1)}(v,v') \max_{R \in \mathcal{M}(u,u')} \prod_{(v,v') \in R} \kappa_{\text{subtree}}^{(k)}(v,v'). \end{aligned}$$

It suffices to show

$$\max_{R \in \mathcal{M}(u,u')} \prod_{(v,v') \in R} \kappa_{\text{subtree}}^{(k-1)}(v,v') \max_{R \in \mathcal{M}(u,u')} \prod_{(v,v') \in R} \kappa_{\text{subtree}}^{(k)}(v,v') = \max_{R \in \mathcal{M}(u,u')} \prod_{(v,v') \in R} \kappa_{\text{subtree}}^{(k)}(v,v').$$

Since the only values can take for $\kappa_{\text{subtree}}^{(k-1)}$ is 0 and 1, the only values that $\max_{R \in \mathcal{M}(u,u')} \prod_{(v,v') \in R} \kappa_{\text{subtree}}^{(k-1)}(v,v')$ can take is also 0 and 1. Then we can split the proof on these two conditions. It is obvious if this term is equal to 1. If this term is equal to 0, then

$$\max_{R \in \mathcal{M}(u,u')} \prod_{(v,v') \in R} \kappa_{\text{subtree}}^{(k)}(v,v') \leq \max_{R \in \mathcal{M}(u,u')} \prod_{(v,v') \in R} \kappa_{\text{subtree}}^{(k-1)}(v,v') = 0,$$

as all terms are not negative and $\kappa^{(k)}_{\text{subtree}}(v, v')$ is not creasing on k. Then $\max_{R \in \mathcal{M}(u,u')} \prod_{(v,v') \in R} \kappa^{(k)}_{\text{subtree}}(v, v') = 0$ and we have 0 for both sides. □

D.2. Recursive relation for the walk kernel

We recall that the k-walk kernel is defined as

$$K(G, G') = \sum_{u \in \mathcal{V}} \sum_{u' \in \mathcal{V}'} \kappa^{(k)}_{\text{walk}}(u, u'),$$

where

$$\kappa^{(k)}_{\text{walk}}(u, u') = \sum_{p \in \mathcal{W}_k(G,u)} \sum_{p' \in \mathcal{W}_k(G',u')} \delta(a(p), a'(p')).$$

The feature map of this kernel is given by

$$\varphi^{(k)}_{\text{walk}}(u) = \sum_{p \in \mathcal{W}_k(G,u)} \varphi_\delta(a(p)),$$

where φ_δ is the feature map associated with δ. We give here a recursive relation for the walk kernel on the size of walks, thanks to its allowance of nodes to repeat.

Lemma 2. *For any $k \geq 0$, we have*

$$\kappa^{(k+1)}_{walk}(u, u') = \delta(a(u), a'(u')) \sum_{v \in \mathcal{N}(u)} \sum_{v' \in \mathcal{N}(u')} \kappa^{(k)}_{walk}(v, v'). \tag{24}$$

Proof. Noticing that we can always decompose a path $p \in \mathcal{W}_{k+1}(G, u)$, with (u, v) the first edge that it passes and $v \in \mathcal{N}(u)$, into (u, q) with $q \in \mathcal{W}_k(G, v)$, then we have

$$\begin{aligned}
\kappa^{(k+1)}_{\text{walk}}(u, u') &= \sum_{p \in \mathcal{W}_{k+1}(G,u)} \sum_{p' \in \mathcal{W}_{k+1}(G',u')} \delta(a(p), a'(p')) \\
&= \sum_{v \in \mathcal{N}(u)} \sum_{p \in \mathcal{W}_k(G,v)} \sum_{v' \in \mathcal{N}(u')} \sum_{p' \in \mathcal{W}_k(G,v')} \delta(a(u), a'(u')) \delta(a(p), a'(p')) \\
&= \delta(a(u), a'(u')) \sum_{v \in \mathcal{N}(u)} \sum_{v' \in \mathcal{N}(u')} \sum_{p \in \mathcal{W}_k(G,v)} \sum_{p' \in \mathcal{W}_k(G',v')} \delta(a(p), a'(p')) \\
&= \delta(a(u), a'(u')) \sum_{v \in \mathcal{N}(u)} \sum_{v' \in \mathcal{N}(u')} \kappa^{(k)}_{\text{walk}}(v, v').
\end{aligned}$$

□

This relation also provides us a recursive relation for the feature maps of the walk kernel

$$\varphi^{(k+1)}_{\text{walk}}(u) = \varphi_\delta(a(u)) \otimes \sum_{v \in \mathcal{N}(u)} \varphi^{(k)}_{\text{walk}}(v),$$

where $\otimes$ denotes the tensor product.

D.3. Discriminative power between walk kernel and WL subtree kernel

Before proving the Theorem 1, let us first show that the WL subtree kernel is always more discriminative than the walk kernel.

Proposition 1. *For any node u in graph G and u' in graph G' and any $k \geq 0$, then $d_{\kappa^{(k)}_{subtree}}(u, u') = 0 \implies d_{\kappa^{(k)}_{walk}}(u, u') = 0$.*

This proposition suggests that though both of their feature maps are not injective (see e.g. Kriege et al. (2018)), the feature map of $\kappa_{\text{subtree}}^{(k)}$ is more injective in the sense that for a node u, its collision set $\{u' \in \mathcal{V} \,|\, \varphi(u') = \varphi(u)\}$ for $\kappa_{\text{subtree}}^{(k)}$, with φ the corresponding feature map, is included in that for $\kappa_{\text{walk}}^{(k)}$. Furthermore, if we denote by $\hat{\kappa}$ the normalized kernel of κ such that $\hat{\kappa}(u,u') = \kappa(u,u')/\sqrt{\kappa(u,u)\kappa(u',u')}$, then we have

Corolary 1. *For any node u in graph G and u' in graph G' and any $k \geq 0$, $d_{\kappa_{subtree}^{(k)}}(u,u') \geq d_{\hat{\kappa}_{walk}^{(k)}}(u,u')$.*

Proof. We prove by induction on k. It is clear for $k = 0$ as both kernels are equal to the Dirac kernel on the node attributes. Let us suppose this is true for $k \geq 0$, we will show this is also true for $k+1$. We suppose $d_{\kappa_{\text{subtree}}^{(k+1)}}(u,u') = 0$. Since $\kappa_{\text{subtree}}^{(k+1)}(u,u) = 1$, by equality (23) we have

$$1 = \kappa_{\text{subtree}}^{(k+1)}(u,u') = \kappa_{\text{subtree}}^{(k)}(u,u') \max_{R \in \mathcal{M}(u,u')} \prod_{(v,v') \in R} \kappa_{\text{subtree}}^{(k)}(v,v'),$$

which implies that $\kappa_{\text{subtree}}^{(k)}(u,u') = 1$ and $\max_{R \in \mathcal{M}(u,u')} \prod_{(v,v') \in R} \kappa_{\text{subtree}}^{(k)}(v,v') = 1$. Then $\delta(a(u), a'(u)) = 1$ by the non-growth of $\kappa_{\text{subtree}}^{(k)}(u,u')$ on k and it exists an exact matching $R^\star \in \mathcal{M}(u,u')$ such that $|\mathcal{N}(u)| = |\mathcal{N}(u')| = |R^\star|$ and $\forall (v,v') \in R^\star$, $\kappa_{\text{subtree}}^{(k)}(v,v') = 1$. Therefore, we have $d_{\kappa_{\text{walk}}^{(k)}}(v,v') = 0$ for all $(v,v') \in R^\star$ by the induction hypothesis.

On the other hand, by Lemma 2 we have

$$\begin{aligned}
\kappa_{\text{walk}}^{(k+1)}(u,u') &= \delta(a(u), a'(u')) \sum_{v \in \mathcal{N}(u)} \sum_{v' \in \mathcal{N}(u')} \kappa_{\text{walk}}^{(k)}(v,v') \\
&= \sum_{v \in \mathcal{N}(u)} \sum_{v' \in \mathcal{N}(u')} \kappa_{\text{walk}}^{(k)}(v,v'),
\end{aligned}$$

which suggest that the feature map of $\kappa_{\text{walk}}^{(k+1)}$ can be written as $\varphi_{\text{walk}}^{(k+1)}(u) = \sum_{v \in \mathcal{N}(u)} \varphi_{\text{walk}}^{(k)}(v)$. Then we have

$$\begin{aligned}
d_{\kappa_{\text{walk}}^{(k+1)}}(u,u') &= \left\| \sum_{v \in \mathcal{N}(u)} \varphi_{\text{walk}}^{(k)}(v) - \sum_{v' \in \mathcal{N}(u')} \varphi_{\text{walk}}^{(k)}(v') \right\| \\
&= \left\| \sum_{(v,v') \in R^\star} \varphi_{\text{walk}}^{(k)}(v) - \varphi_{\text{walk}}^{(k)}(v') \right\| \\
&\leq \sum_{(v,v') \in R^\star} \| \varphi_{\text{walk}}^{(k)}(v) - \varphi_{\text{walk}}^{(k)}(v') \| \\
&= \sum_{(v,v') \in R^\star} d_{\kappa_{\text{walk}}^{(k)}}(v,v') = 0.
\end{aligned}$$

We conclude that $d_{\kappa_{\text{walk}}^{(k+1)}}(u,u') = 0$.

Now let us prove the Corollary 1. The only values that $d_{\kappa_{\text{subtree}}^{(k)}}(u,u')$ can take are 0 and 1. Since $d_{\hat{\kappa}_{\text{walk}}^{(k)}}(u,u')$ is always not larger than 1, we only need to prove $d_{\kappa_{\text{subtree}}^{(k)}}(u,u') = 0 \implies d_{\hat{\kappa}_{\text{walk}}^{(k)}}(u,u') = 0$, which has been shown above. □

D.4. Proof of Theorem 1

Note that using our notation here, $\varphi_1 = \varphi_{\text{walk}}^{(k)}$

Proof. We prove by induction on k. For $k = 0$, we have for any $u \in \mathcal{V}$ and $u' \in \mathcal{V}'$

$$\kappa_{\text{subtree}}^{(0)}(u,u') = \delta(a(u), a'(u')) = \delta(\varphi_{\text{walk}}^{(0)}(u), \varphi_{\text{walk}}^{(0)}(u')).$$

Assume that (16) is true for $k \geq 0$. We want to show this is also true for $k+1$. As the only values that the δ kernel can take is 0 and 1, it suffices to show the equality between $\kappa_{\text{subtree}}^{(k+1)}(u,u')$ and $\delta(\varphi_{\text{walk}}^{(k+1)}(u), \varphi_{\text{walk}}^{(k+1)}(u'))$ in these two situations.

- If $\kappa_{\text{subtree}}^{(k+1)}(u, u') = 1$, by Proposition 1 we have $\varphi_{\text{walk}}^{(k+1)}(u) = \varphi_{\text{walk}}^{(k+1)}(u')$, and thus $\delta(\varphi_{\text{walk}}^{(k+1)}(u), \varphi_{\text{walk}}^{(k+1)}(u')) = 1$.
- If $\kappa_{\text{subtree}}^{(k+1)}(u, u') = 0$, by the recursive relation of the feature maps in Lemma 2, we have

$$\delta(\varphi_{\text{walk}}^{(k+1)}(u), \varphi_{\text{walk}}^{(k+1)}(u')) = \delta(a(u), a'(u'))\delta\left(\sum_{v\in\mathcal{N}(u)} \varphi_{\text{walk}}^{(k)}(v), \sum_{v'\in\mathcal{N}(u')} \varphi_{\text{walk}}^{(k)}(v')\right).$$

By Lemma 1, it suffices to show that

$$\max_{R\in\mathcal{M}(u,u')} \prod_{(v,v')\in R} \kappa_{\text{subtree}}^{(k)}(u, u') = 0 \implies \delta\left(\sum_{v\in\mathcal{N}(u)} \varphi_{\text{walk}}^{(k)}(v), \sum_{v'\in\mathcal{N}(u')} \varphi_{\text{walk}}^{(k)}(v')\right) = 0.$$

The condition $|\mathcal{M}(u, u')| = 1$ suggests that there exists exactly one matching of the neighbors of u and u'. Let us denote this matching by R. The left equality implies that there exists a non-empty subset of neighbor pairs $S \subseteq R$ such that $\kappa_{\text{subtree}}^{(k)}(v, v') = 0$ for any $(v, v') \in S$ and $\kappa_{\text{subtree}}^{(k)}(v, v') = 1$ for all $(v, v') \notin S$. Then by the induction hypothesis, $\varphi_{\text{walk}}^{(k)}(v) = \varphi_{\text{walk}}^{(k)}(v')$ for all $(v, v') \notin S$ and $\varphi_{\text{walk}}^{(k)}(v) \neq \varphi_{\text{walk}}^{(k)}(v')$ for all $(v, v') \in S$. Consequently, $\sum_{(v,v')\notin S} \varphi_{\text{walk}}^{(k)}(v) - \varphi_{\text{walk}}^{(k)}(v') = 0$. Now we will show $\sum_{(v,v')\in S} \varphi_{\text{walk}}^{(k)}(v) - \varphi_{\text{walk}}^{(k)}(v') \neq 0$ since all neighbors of either u or u' have distinct attributes. Then

$$\begin{aligned}
&\| \sum_{v\in\mathcal{N}(u)} \varphi_{\text{walk}}^{(k)}(v) - \sum_{v'\in\mathcal{N}(u')} \varphi_{\text{walk}}^{(k)}(v')\| \\
=&\| \sum_{(v,v')\in R} \varphi_{\text{walk}}^{(k)}(v) - \varphi_{\text{walk}}^{(k)}(v')\| \\
=&\| \sum_{(v,v')\in S} \varphi_{\text{walk}}^{(k)}(v) - \varphi_{\text{walk}}^{(i)}(v')\| > 0.
\end{aligned}$$

Therefore, $\delta\left(\sum_{v\in\mathcal{N}(u)} \varphi_{\text{walk}}^{(k)}(v), \sum_{v'\in\mathcal{N}(u')} \varphi_{\text{walk}}^{(k)}(v')\right) = 0$.

□