Q1: Since we put a base kernel K_e on each arc $e = (u \to v) \in E$ (i.e. $K_e : X \times X \to \mathbb{R}$), each edge (arc) is a kernel.

By Thm in Slide 15, the product of kernels is a kernel. We know that $k_p(x, z) = \prod_{i=1}^{d} K_{u_{i-1} \to u_i}(x, z)$ is the product of the arc, and the path is also a kernel.

By Thm in Slide 15, the sum of kernels is a kernel. We know that $k_G(x, z) = \sum_{p \in path(s \to t)} k_p(x, z)$ is the sum of the path, and so it is a kernel.

Therefore, $k_G(x, z)$ is a kernel. $\blacksquare$

Q2:

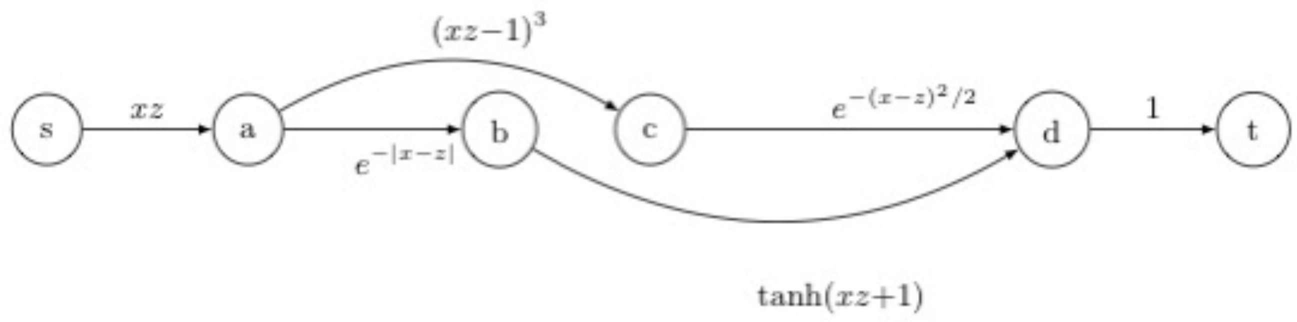


Figure 1: DAG with base kernels. The constant 1 denotes the trivial kernel $\kappa(x, z) \equiv 1$.

$$K_G(1,-1) = \sum_{P \in Path(s \to t)} K_P(1,-1).$$

From the graph above, there are 2 paths from s to t:
$P_1 = s \to a \to c \to d \to t$ and $P_2 = s \to a \to b \to d \to t$

$$K_{P_1}(1,-1) = \prod_{i=1}^{d} K_{u_{i-1} \to u_i}(1,-1)$$

$$= K_{s \to a}(1,-1) \cdot K_{a \to c}(1,-1) \cdot K_{c \to d}(1,-1) \cdot K_{d \to t}(1,-1)$$

$$= 1(-1) \cdot (1(-1)-1)^3 \cdot e^{\frac{-(1-(-1))^2}{2}} \cdot 1$$

$$= -1 \cdot (-8) \cdot e^{-2}$$

$$= 8e^{-2}$$

$$K_{P_2}(1,-1) = \prod_{i=1}^{d} K_{u_{i-1} \to u_i}(1,-1)$$

$$= K_{s \to a}(1,-1) \cdot K_{a \to b}(1,-1) \cdot K_{b \to d}(1,-1) \cdot K_{d \to t}(1,-1)$$

$$= 1(-1) \cdot e^{-|1-(-1)|} \cdot \tanh(-1+1) \cdot 1$$

$$= -e^{-2} \cdot \tanh(0)$$

$$= -e^{-2} \cdot 0$$

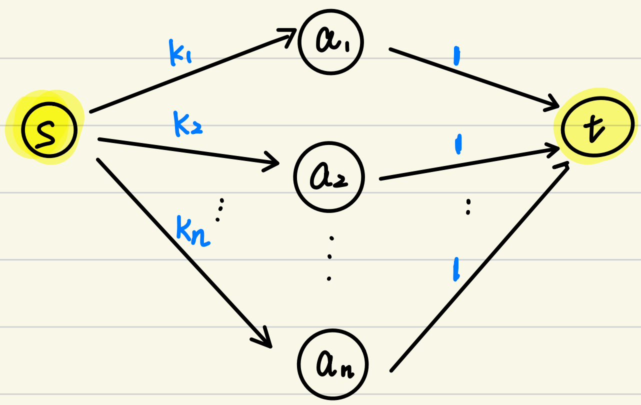
$$= 0$$

Therefore, $K_G(1,-1) = \sum_{P \in Path(s \to t)} K_P(1,-1). = K_{P_1}(1,-1) + K_{P_2}(1,-1)$

$$= 8e^{-2}.$$

Let $s, t, a_1, a_2, \dots, a_n$ be node.

Q3: Let node s be the source, and node t be the sink.



Constant 1 denotes the trivial kernel $k(x,z) \equiv 1$.

There is n Paths from s to t :

$P_1 = s \to a_1 \to t$

$P_2 = s \to a_2 \to t$

$\vdots$

$P_n = s \to a_n \to t$

For path i, $k_{P_i}(x,z) = \prod_{i=1}^{d} k_{u_{i-1} \to u_i}(x,z)$

$= k_{s \to a_i}(x,z) \; k_{a_i \to t}(x,z)$

$= k_i \cdot 1$

$= k_i$

$$k_{\mathcal{G}}(x,z) = \sum_{P \in Path(s \to t)} k_P(x,z) = \sum_{i=1}^{n} k_{P_i}(x,z) = \sum_{i=1}^{n} k_i \quad \blacksquare$$

Let $s, t, a_1, a_2, \dots, a_n$ be node.

Q4: Let node s be the source, and node t be the sink.

$$s \xrightarrow{k_1} a_1 \xrightarrow{k_2} a_2 \xrightarrow{k_3} \cdots \; a_n \xrightarrow{k_n} t$$

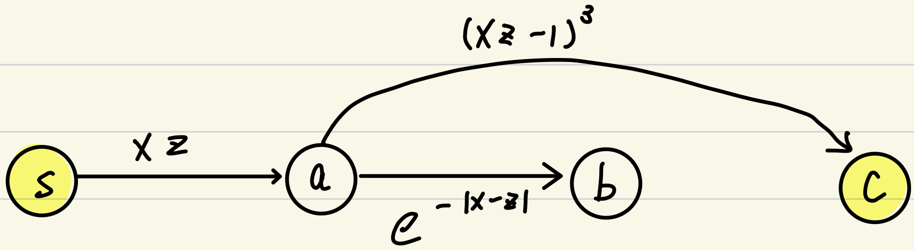
There is only one Path from s to t. $: P_1 = S \to a_1 \to a_2 \to \cdots \to a_n \to t$

$$\begin{aligned} K_{P_1}(X, Z) &= \prod_{i=1}^{d} K_{u_{i-1} \to u_i}(X, Z) \\ &= K_{s \to a_1}(X, Z) \, K_{a_1 \to a_2}(X, Z) \, K_{a_2 \to a_3}(X, Z) \cdots K_{a_n \to t}(X, Z) \\ &= k_1 k_2 k_3 \cdots k_n \\ &= \prod_{i=1}^{n} k_i \end{aligned}$$

$$K_g(X, Z) = \sum_{P \in Path(s \to t)} K_P(X, Z) = K_{P_1}(X, Z) = \prod_{i=1}^{n} k_i$$

Thus, $K_g(X, Z) = \prod_{i=1}^{n} k_i$ ∎

Q5:



From the graph above, there is only 1 path from s to c:

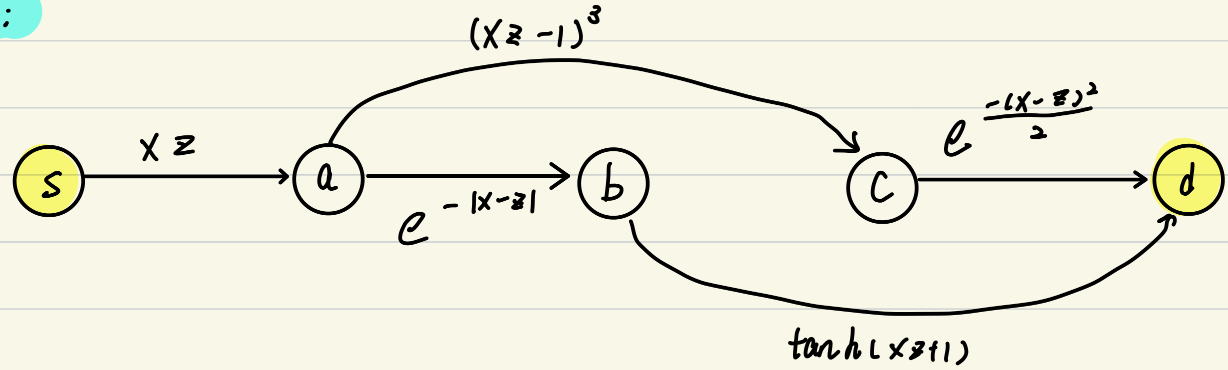
$$P_1 = s \to a \to c$$

$$K_{P_1}(x,z) = \prod_{i=1}^{d} k_{u_{i-1}\to u_i}(x,z) = K_{s\to a}(x,z)\; K_{a\to c}(x,z)$$

$$= xz\,(xz-1)^3$$

$$K_G(x,z) = \sum_{P\in Path(S\to C)} K_P(x,z) = K_{P_1}(x,z) = xz\,(xz-1)^3$$

Q6:



From the graph above, there is only 1 path from s to d:

$$P_1 = s \to a \to c \to d \text{ , and } P_2 : s \to a \to b \to d$$

$$K_{P_1}(x,z) = \prod_{i=1}^{d} k_{u_{i-1}\to u_i}(x,z) = K_{s\to a}(x,z)\; K_{a\to c}(x,z)\; K_{c\to d}(x,z)$$

$$= xz\,(xz-1)^3\, e^{-\frac{(x-z)^2}{2}}$$

$$K_{P_2}(x,z) = \prod_{i=1}^{d} k_{u_{i-1}\to u_i}(x,z) = K_{s\to a}(x,z)\; K_{a\to b}(x,z)\; K_{b\to d}(x,z)$$

$$= xz\, e^{-|x-z|}\, \tanh(xz+1)$$

$$K_G(x,z) = \sum_{P\in Path(S\to C)} K_P(x,z) = K_{P_1}(x,z) + K_{P_2}(x,z)$$

$$= xz\,(xz-1)^3\, e^{\frac{-(x-z)^2}{2}} + xz\, e^{-|x-z|}\, \tanh(xz+1)$$

$$= xz\left[(xz-1)^3\, e^{\frac{-(x-z)^2}{2}} + e^{-|x-z|}\, \tanh(xz+1)\right]$$

Q7: Input: a directed acyclic graph G

Output: graph kernel

1. Assume node s is the source, and node t is the sink.
2. Add an extra field Kernel into the DAG structure, and set it to 0 — $O(|V|)$
3. topological_sorted_nodes = topological_sort(G) — $O(|V|+|E|)$
4. for each node u in topological_sorted_nodes — $O(|V|+|E|)$ (lines 4–8)
5. outgoing_arcs = all outgoing arcs of node u
6. for each arc e in outgoing_arcs ($u \xrightarrow{e} v$)
7. ke = kernel value of arc e
8. v.kernel += u.kernel * ke
9. return t.Kernel

Runtime Justification

- Adding the extra field and initializing to 0 will take $O(|V|)$.
- Topological Sort takes $O(|V|+|E|)$.
- For the nested for loops (Line 4-8): it visits all the nodes and their outgoing arcs. Since line 7-8 only takes $O(1)$, the loop takes $O(|V|+|E|)$ time.
- Total Runtime $= O(|V|) + O(|V|+|E|) + O(|V|+|E|)$
  $= O(|V|+|E|)$

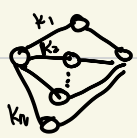
Algorithm Justification

Topological sort makes the node u comes before node v for each $u \longrightarrow v$, so we can ensure always update the kernel value of u before v.

By adding a kernel field to each node, we can keep track of K_G value.

There are 2 cases to compute graph Kernel between adjacent nodes:

Case ①:



Do the summation.

$K_G = \sum_{i=1}^{n} K_i$

Case ②:  Do the multiplication.

$$k_G = \prod_{i=1}^{n} k_i$$

Assume $u_1, u_2, u_3, \dots, u_n$ are the ancestors of node t.

i.e.

k_g from s to t $= \sum_{i=1}^{n} k_g$ from s to $u_i \cdot k_e$

times

$$= \sum_{p \in Path(s,t)} k_p(s,t) \qquad \blacksquare$$