ICML23 Rebuttal for Submission 6191

Anonymous

March 2023

In this pdf, we will answer your first (and the most important) question about proposition 4.4 and 4.5. We have proved the strictness and thus completely solved your question.

1 Proof for proposition 4.4 and 4.5

Here we address that increase k and l will indeed **strictly** increase expressive power of k, l-WL, which is a non-trivial and significant result. We will also provide our full proof to proposition 4.4 and 4.5.

As 2-WL is of the same expressivity as 1-WL, 1-WL on each labeled graph ($1, l$-WL) is equivalent to 2-WL on each labeled graph ($2, l$-WL). Therefore, $1, l$-WL and $2, l$-WL have the same expressivity. Hence, all our following proofs are for $k \geq 2$. Specifically, $1, l$-WL $\cong 2, l$-WL, $\forall l$.

Theorem 1.1. *(Proposition 4.4 in main text) $\forall k \geq 2$, $k+1, l$-WL is strictly more powerful than k, l-WL.*

Theorem 1.2. *(Proposition 4.5 in main text) $\forall k \geq 2$, $k, l+1$-WL is strictly more powerful than k, l-WL.*

Proof sketch:

1. First, we will prove a new conclusion. Proposition 4.12: k, l-WL is more powerful than $k-1, l+1$-WL. Hence we get $k+1, l$-WL $\succeq k, l+1$-WL $\succeq ... \succeq$ $2, k+l-1$-WL.

2. Then we will show that $2, k+l-1$-WL can distinguish some non-isomorphic graphs which are indistinguishable by $k+l$-WL.

3. Note that we have already proved in proposition 4.11 that k, l-WL $\preceq$ $(k+l)$-WL, $\forall k \geq 2, \forall l$. Also, we have already shown that $k+1, l$-WL $\succeq k, l$-WL and $k, l+1$-WL $\succeq k, l$-WL. Step 2 gives some non-isomorphic graphs indistinguishable by $k+l$-WL but distinguishable by $2, k+l-1$-WL, which also means they are indistinguishable by k, l-WL (proposition 4.11), and distinguishable by $k+1, l$-WL and $k, l+1$-WL (proposition 4.12). Combining all pieces together, we obtain that $k+1, l$-WL $\succ k, l$-WL and $k, l+1$-WL $\succ k, l$-WL, which proves that $k+1, l$-WL and $k, l+1$-WL are strictly more powerful than k, l-WL.

In the following text, we will provide the detailed proof.

1.1

In this step, we will propose and prove Proposition 4.12.

Proposition 1.3. *(Proposition 4.12 in rebuttal) $\forall k \geq 3$, k, l-WL is more powerful than $k-1, l+1$-WL. That is, $k-1, l+1$-WL $\preceq k, l$-WL.*

In this section of proof, we use $\tilde{C}$ to denote $k-1, l+1-WL$ and C to denote k, l-WL.

We first restate k, l-WL here. Given a graph G with a l-node tuple u labeled as color $\{1, 2, 3, ..., l\}$ (denoted as $G^{(u)}$), the initial color of a v tuple is,

$$C^0_{v,u} = C(G^{(u)}, v), \tag{1}$$

where $C(G, v)$ return the isomorphism type of the ordered subgraph induced by v in G.

The update process in the t-th iteration is,

$$C^t_{v,u} = C^{t-1}_{v,u}, (\{C^{t-1}_{N_j(v),u} | i \in V\} | j \in [k]). \tag{2}$$

We then prove that

Theorem 1.4. $\forall v \in V^{k-1}, v_0 \in V, u \in V^k$

$$C^t_{v\|v_0,u} \to \tilde{C}^t_{v,v_0\|u} \tag{3}$$

Proof. We prove it by induction on t

- If $v_0 \notin V$

  $$C^0_{v\|v_0,u} = C(G^{(u)}, v\|v_0) \to C(G^{(u)}, v) = C(G^{(v_0\|u)}, v) = \tilde{C}^0_{v|v_0\|u} \tag{4}$$

  else if $v_0 \in V$

  $$C^0_{v\|v_0,u} = C(G^{(u)}, v\|v_0) \leftrightarrow C(G^{(v_0\|u)}, v\|v_0) \to C(G^{(v_0\|u)}, v) = \tilde{C}^0_{v|v_0\|u} \tag{5}$$

- When $t > 0$, assume that $\forall t' < t, C^{t'}_{v\|v_0,u} \to \tilde{C}^{t'}_{v,v_0\|u}$.

$$\begin{aligned}
C^t_{v\|v_0,u} &= C^{t-1}_{v\|v_0,u}, (\{C^{t-1}_{N_j(v\|v_0,a),u} | a \in V\} | j \in [k]) && (6)\\
&\to C^{t-1}_{v\|v_0,u}, (\{C^{t-1}_{N_j(v,a)\|v_0,u} | a \in V\} | j \in [k-1]) && (7)\\
&\to \tilde{C}^{t-1}_{v,v_0\|u}, (\{\tilde{C}^{t-1}_{N_j(v,a),v_0\|u} | a \in V\} | j \in [k-1]) && (8)\\
&= C^t_{v\|v_0,u}. && (9)
\end{aligned}$$

□

So far we have shown that $k-1, l+1$-WL $\preceq k, l$-WL. Using induction, we can easily obtain that:

Corollary 1.5. *$k+1, l$-WL $\succeq k, l+1$-WL $\succeq ... \succeq 2, k+l-1$-WL.*

This is an important and non-trivial property of k, l-WL. Also, we will use this property in step 3 for our final proof.

1.2

In this step, We will show that $2, k+l-1$-WL can distinguish some non-isomorphic graphs which are indistinguishable by $k+l$-WL.

We can construct CFI-gadget graphs [2] G_{k+l} and H_{k+l}, which can be distinguished by $2, k+l-1$-WL, but cannot be distinguished by $k+l$-WL.

Construction of CFI-Gadgets For simplicity, we will use $k \in \mathbb{N}$ to describe the construction process. CFI-Gadgets are an infinite family of graphs $(G_k, H_k), k \in \mathbb{N}$, such that (1) k-WL cannot distinguish G_k and H_k, but (2) $k+1$-WL can distinguish G_k and H_k. We will restate the construction process, but one can refer to [2] for more details.

Let K_{k+1} denote the complete graph on $k+1$ nodes. The nodes are indexed from 0 to k. Let $E(v)$ denote the set of edges incident to v in K_{k+1}, then $|E(v)| = k, \forall v \in V(K_{k+1})$. We define the graph G_k as follows.

- For the node set $V(G_k)$, add: (a) (v, S) for each v in $V(K_{k+1})$ and for each *even* subset S of $E(v)$; (b) two nodes e^1 and e^0 for each edge e in $E(K_{k+1})$.
- For the edge set $E(G_k)$, add: (a) an edge (e^0, e^1) for each e in $E(G_k)$; (b) an edge between (v, S) and e^1 if v in e and e in S; (c) an edge between (v, S) and e^0 if v in e and e not in S.

H_k are constructed in a similar manner, with the following exceptions: in step 1(a), for node indexed 0 in $V(K_{k+1})$, we choose all *odd* subsets of $E(0)$. Following the above construction, both graphs have $k \cdot 2^k + \binom{k+2}{2} \cdot 2$ nodes. This is a typical variation of Cai-Fürer-Immerman construction.

Note that for $v \in V(K_{k+1})$, vertices of the form (v, S) defined in Item 1 are assigned a common color C_v. These vertices form a *vertex-cloud* corresponding to the vertex v.

Distance-two cliques and properties of (G_k, H_k) A set S of nodes is said to form a *distance-two-clique* if the distance between any two nodes in S is exactly two. A distance-two-clique S is colorful if (a) every vertex of S is of vertex-cloud kind, and (b) no two vertices in S belong to the same vertex cloud. It's easy to verify that each vertex in a colorful distance-two-clique has a unique initial color.

Note that G_k and H_k are non-isomorphic, which has been proved in [3].

Lemma 1.6. *There exists a set of $k+1$ vertex-cloud vertices in G_k such that they form a colorful distance-two-clique of size $k+1$, but there does not exist a set of $k+1$ vertex-cloud vertices in H_k such that they form a colorful distance-two-clique of size $k+1$.*

Further, in [3] the authors proved that:

Lemma 1.7. *k-WL cannot distinguish G_k and H_k, but k+1-WL can distinguish G_k and H_k.*

Now we will show that $2, k-1$-WL can distinguish G_k and H_k. It's remarkable that here k is used to parameterize number of labels (as l does in the main text) for simplicity. A similar proof can be found in [4], where their k-OSWL is equivalent to our $2, k$-WL (as well as $1, k$-WL, note that they all use FWL descriptions).

Lemma 1.8. *$2, k-1$-WL can distinguish G_k and H_k.*

Proof. It only remains to show that $2, k-1$-WL can distinguish the colorful distance-two clique Q of size $k+1$, which appears in G_k but not in H_k. We follow a paradigm of pebble games (see [1]), where we place $k-1$ fixed pebbles (the role of our explicit labels) on $k-1$ vertices of Q, while remaining two vertices $x, y \in Q$. Obviously, after two rounds of color refinement, x and y will detect all individualized colors corresponding to the $k-1$ fixed pebbles. Meanwhile, the individualized pebbles also detect all other pebbles' individualized colors. However, same procedure will not obtain same colors in H_k, otherwise it indicates a colorful distance-two clique. Hence, for all $x' \in V(H_k)$ and $v \in V(H_k)^{k-1}$, the color refinement on node x and x' will lead to different colors:

$$\{\{C_{x||u} | u \in V(G_k)^{k-1}\}\} \neq \{\{C_{x'||v \in V(H_k)^{k-1}}\}\} \tag{10}$$

Since the colors are different for all labeling orders (ordered subgraphs), the final colors of graphs (pooled from set of subgraph colors) are also different. Above all, $2, k-1$-WL can distinguish G_k an H_k. □

Now back to our main theorem, we use $k+l-1$ to replace $k-1$ in the above conclusions (to align with notations), then we will get:

Lemma 1.9. *$2, k+l-1$-WL distinguish CFI-Gadgets G_{k+l} and H_{k+l}, but $k+l$-WL cannot distinguish them.*

Using this lemma, combined with proposition 4.11, 4.12 and conclusions in main texts, we can finally prove that $k+1, l$-WL and $k, l+1$-WL are strictly more powerful than k, l-WL. See below (step 3) for detailed logic.

1.3

In this step, we combine all pieces together and finally prove the strictness.

From step 1, we obtain that both $k+1, l$-WL and $k, l+1$-WL are at least as powerful as $2, k+l-1$-WL. From step 2, we know that $2, k+l-1$-WL can distinguish non-isomorphic CFI-graphs G_{k+l} and H_{k+l} which are indistinguishable by $k+l$-WL. Therefore, both $k+1, l$-WL and $k, l+1$-WL can distinguish these non-isomorphic graphs (CFI-Gadget graphs), but $k+l$-WL cannot. Notice that we have already shown in proposition 4.11 that $k+l$-WL upper bounds k, l-WL, hence k, l-WL cannot distinguish G_{k+l} and H_{k+l} either. Recall that we have shown that both $k+1, l$-WL and $k, l+1$-WL are at least as powerful as k, l-WL, i.e. $k+1, l\text{-WL} \succeq k, l\text{-WL}$, $k, l+1\text{-WL} \succeq k, l\text{-WL}$. Now since we obtain CFI graphs G_{k+l} and H_{k+l} that can distinguished by both $k+1, l$-WL and $k, l+1$-WL, but not by k, l-WL, the strictness holds. Above all, we have proved proposition 4.4 and 4.5 at the same time:

Proposition 1.10. *(Proposition 4.4 and 4.5 in main text)* $k+1, l\text{-}WL \succeq k, l+1\text{-}WL \succ k, l\text{-}WL$.

So far, we have proved all theorems and propositions in main texts (including the strictness of increasing power), and solved all your problems. We also proposed some useful, non-trivial conclusions (proposition 4.11, 4.12), which are also important contributions aside of the main theorems.

Reference

[1] Huang, Y., Peng, X., Ma, J., Zhang, M. (2022). Boosting the Cycle Counting Power of Graph Neural Networks with I2-GNNs. ArXiv, abs/2210.13978

[2] Cai, J., Fürer, M., Immerman, N. (1989). An optimal lower bound on the number of variables for graph identification. Combinatorica, 12, 389-410.

[3] Morris, C., Mutzel, P. (2019). Towards a practical k-dimensional Weisfeiler-Leman algorithm. ArXiv, abs/1904.01543.

[4] Qian, C., Rattan, G., Geerts, F., Morris, C., Niepert, M. (2022). Ordered Subgraph Aggregation Networks. ArXiv, abs/2206.11168.

2 Detailed proof of $C(G, \mathbf{v}||\mathbf{u}) \to C(G^{(u)}, \mathbf{v})$

$C(G, \mathbf{v})$ is the isomorphism tuple $\mathbf{v}$ in graph G. Given graphs G^1, G^2 and n-tuples $\mathbf{v}^1, \mathbf{v}^2$, $C(G^1, \mathbf{v}^1) = C(G^2, \mathbf{v}^2)$ iff (see [5])

1. $\forall i_1, i_2 \in \{1, 2, ..., n\}$, $\mathbf{v}^1_{i_1} = \mathbf{v}^1_{i_2} \leftrightarrow \mathbf{v}^2_{i_1} = \mathbf{v}^2_{i_2}$

2. $\forall i \in \{1, 2, ..., n\}, d(\mathbf{v}^1_i) = d(\mathbf{v}^2_i)$, where d means node degree and initial node feature.

3. $\forall i_1, i_2 \in \{1, 2, ..., n\}$, $E(\mathbf{v}^1_{i_1}, \mathbf{v}^1_{i_2}, G^1) = E(\mathbf{v}^2_{i_1}, \mathbf{v}^2_{i_2}, G^2)$, where

$$E(v_1, v_2, G) = \begin{cases} 1 & \text{edge } (v_1, v_2) \text{ in graph } G \\ 0 & \text{otherwise} \end{cases}$$

Mathematically:

Given two graph G^1, G^2 two n-tuples $\mathbf{v}^1, \mathbf{v}^2$, two l-tuples $\mathbf{u}^1, \mathbf{u}^2$

$$C(G^1, \mathbf{v}^1||\mathbf{u}^1) = C(G^2, \mathbf{v}^2||\mathbf{u}^2) \Rightarrow \forall i_1, i_2 \in [n+l], (\mathbf{v}^1||\mathbf{u}^1)_{i_1} = (\mathbf{v}^1||\mathbf{u}^1)_{i_2} \leftrightarrow (\mathbf{v}^2||\mathbf{u}^2)_{i_1} = (\mathbf{v}^2||\mathbf{u}^2)_{i_2} \tag{11}$$

$$\forall i \in [n+l], d(G^1, (\mathbf{v}^1||\mathbf{u}^1)_i) = d(G^2, (\mathbf{v}^2||\mathbf{u}^2)_i) \tag{12}$$

$$\forall i_1, i_2 \in [n+l], E((\mathbf{v}^1||\mathbf{u}^1)_{i_1}, (\mathbf{v}^1||\mathbf{u}^1)_{i_2}, G^1) \leftrightarrow E((\mathbf{v}^2||\mathbf{u}^2)_{i_1}, (\mathbf{v}^2||\mathbf{u}^2)_{i_2}, G^2) \tag{13}$$

1.

$$\forall i_1, i_2 \in [n+l], (\mathbf{v}^1||\mathbf{u}^1)_{i_1} = (\mathbf{v}^1||\mathbf{u}^1)_{i_2} \leftrightarrow (\mathbf{v}^2||\mathbf{u}^2)_{i_1} = (\mathbf{v}^2||\mathbf{u}^2)_{i_2} \tag{14}$$

$$\Rightarrow \forall i_1, i_2 \in [n], \mathbf{v}^1_{i_1} = \mathbf{v}^1_{i_2} \leftrightarrow \mathbf{v}^2_{i_1} = \mathbf{v}^2_{i_2} \tag{15}$$

2.

$$\forall i \in [n+l], d(G^1, (\mathbf{v}^1||\mathbf{u}^1)_i) = d(G^2, (\mathbf{v}^2||\mathbf{u}^2)_i) \Rightarrow \forall i \in [n], d(G^1, \mathbf{v}^1_i) = d(G^2, \mathbf{v}^2_i) \tag{16}$$

Moreover,

$$d(G^{(\mathbf{u})}, v) = \begin{cases} d(G, v) & v \notin \mathbf{u} \\ (d(G, v), \{i \in [l] | v = \mathbf{u}_i\}) & v \in \mathbf{u} \end{cases}$$

As

$$\forall i_1, i_2 \in [n+l], (\mathbf{v}^1||\mathbf{u}^1)_{i_1} = (\mathbf{v}^1||\mathbf{u}^1)_{i_2} \leftrightarrow (\mathbf{v}^2||\mathbf{u}^2)_{i_1} = (\mathbf{v}^2||\mathbf{u}^2)_{i_2} \tag{17}$$

$$\Rightarrow \forall i \in [n], \{j \in [l] | \mathbf{v}^1_i = \mathbf{u}^1_j\} = \{j \in [l] | \mathbf{v}^2_i = \mathbf{u}^2_j\} \tag{18}$$

Therefore,

$$\forall i \in [n], d(G^{1,(\mathbf{u}^1)}, \mathbf{v}^1_i) = d(G^{2,(\mathbf{u}^2)}, \mathbf{v}^2_i)$$

3.

$$\forall i_1, i_2 \in [n+l], E((\mathbf{v}^1||\mathbf{u}^1)_{i_1}, (\mathbf{v}^1||\mathbf{u}^1)_{i_2}, G^1) \leftrightarrow E((\mathbf{v}^2||\mathbf{u}^2)_{i_1}, (\mathbf{v}^2||\mathbf{u}^2)_{i_2}, G^2) \tag{19}$$

$$\Rightarrow \forall i_1, i_2 \in [n], E(\mathbf{v}^1_{i_1}, \mathbf{v}^1_{i_2}, G^1) \leftrightarrow E(\mathbf{v}^2_{i_1}, \mathbf{v}^2_{i_2}, G^2) \tag{20}$$

$$\Rightarrow \forall i_1, i_2 \in [n], E(\mathbf{v}^1_{i_1}, \mathbf{v}^1_{i_2}, G^{1,(u^1)}) \leftrightarrow E(\mathbf{v}^2_{i_1}, \mathbf{v}^2_{i_2}, G^{2,(u^2)}) \tag{21}$$

$$\tag{22}$$

Combining 1,2,3, $C(G^1, \mathbf{v}^1||\mathbf{u}^1) = C(G^2, \mathbf{v}^2||\mathbf{u}^2) \Rightarrow C(G^{1,(\mathbf{u}_1)}, \mathbf{v}^1) = C(G^{2,(\mathbf{u}_2)}, \mathbf{v}^2)$. Therefore, $C(G, \mathbf{v}||\mathbf{u}) \to C(G^{(u)}, \mathbf{v})$.

[5] Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, Yaron Lipman. Provably Powerful Graph Networks. NeurIPS 2019.