A. Supplementary material

We now provide detailed proofs for all our propositions and lemmas.

Proof of Proposition 1

Proof. We show that CPNGNN, using some consistent port ordering, can distinguish some non-isomorphic graphs that LU-GNNs cannot.

We construct a pair of graphs $G$ and $\bar{G}$ such that $G$ consists of two triangles that differ in port-ordering but are otherwise identical, while $\bar{G}$ (indicated by underlined symbols) consists of a single even-length cycle. The construction ensures that each node labeled with $X \in \{B_1, C_1, D_1, B_2, C_2\}$ in $G$ has the same identical view (i.e., indistinguishable node features, and neighborhood) as the corresponding node labeled $\bar{X}$ in $\bar{G}$. However, $D_2$ and $\bar{D}_2$ have distinguishable neighborhoods due to different port-numbers: e.g., $D_2$ is connected to $B_2$ at port 2, whereas $\bar{D}_2$ is connected to $\bar{B}_1$ at port 1. Likewise, $D_2$ is connected to $C_2$ at port 1, in contrast to $\bar{D}_2$ that is connected to $\bar{C}_2$ at port 2. However, LU-GNN does not incorporate any spatial information such as ports, and fails to tell one graph from the other.

Note that since $\angle B_1C_1D_1$ differs from $\angle B_1C_1\bar{D}_1$, DimeNet can also distinguish between the two graphs.

Proof of Proposition 2

Proof. We now illustrate the importance of choosing a good consistent port numbering. Specifically, we construct a pair of graphs, and two different consistent port numberings $p$ and $q$ such that CPNGNN can distinguish the graphs with $p$ but not $q$.

We modify the consistent port numbering from the construction of Proposition 1. We consider the same pair of graphs as in the proof of Proposition 1. However, instead of having different numberings for the two components (i.e., triangles) of $G$, we now carry over the ordering from one component to the other. The two components become identical with this modification. For any node labeled $X_1$ or $\bar{X}_2$, and any neighbor labeled $Y_1$ or $\bar{Y}_2$, $X, Y \in \{B, C, D\}$, we can now simply assign the same respective local ports as the nodes labeled $X_1$ and $Y_1$ (or, equivalently, $X_2$ and $Y_2$). It is easy to verify that the two graphs become port-locally isomorphic under the new ordering, and thus cannot be separated with any permutation-invariant readout (using Proposition 3).

Proof of Proposition 3

Proof. We begin with the following definition.

Definition 3. Two nodes in a graph are locally indistinguishable if they have identical feature vectors and identical port-ordered neighborhoods.

In other words, for locally indistinguishable nodes $u$ and $v$, not only are their neighbors identical but also the respective ports that connect $u$ and $v$ to their identical neighbors are identical.

If the surjection $f : V_1 \rightarrow V_2$ in Definition 2 is also injective, then we can simply take $h = f$. Therefore, we focus on the case when $f$ is not injective. We will show that $f$ can be used to inform $h$. Since $f$ is not injective, there exist $v_1, v'_1 \in V_1$ such that $v_1 \neq v'_1$ but $f(v_1) = f(v'_1) = v_2$ for some $v_2 \in V_2$. Then, by condition (a) in Definition 2 we immediately get that the feature vector

$$x_{v_1} = x_{f(v_1)} = x_{f(v'_1)} = x_{v'_2}. \tag{6}$$

Moreover, by other conditions, there is a consistent port bijection from neighborhood of $v_1$ to that of $v_2$, and likewise another bijection from neighborhood of $v'_1$ to that of $v'_2$. Therefore, there is a consistent port bijection from neighborhood of $v_1$ to that of $v'_1$. Together with (6) and our assumption that $f(v_1) = f(v'_1) = v_2$, this implies that $v_1$ and $v'_1$ are locally indistinguishable. Note that there could be more such nodes that are indistinguishable from $v_1$ (or $v'_1$), e.g., when all such nodes map to $v_2$ as well.

Without loss of generality, let $E_1(v_1) \subseteq V_1$ denote the equivalence class of all nodes, including $v_1$, that are indistinguishable from $v_1$ in graph $G_1$. Similarly, let $E_2(v_2) \subseteq V_2$ be the class of nodes indistinguishable from $v_2$ in $G_2$. Consider $\ell_1 = |E_1(v_1)|$ and $\ell_2 = |E_2(v_2)|$. We claim that $\ell_1 = \ell_2$. Suppose not. Then if $\ell_1 < \ell_2$, we can have $h$ map each node in $E_1(v_1)$ to a separate node in $E_2(v_2)$, and use the same mapping as $f$ on the other nodes in $V_1$. Doing so does not decrease the co-domain of
$V_2$, and $h$ remains surjective. We are therefore left with $\ell_2 - \ell_1 \geq 0$ nodes from $\mathcal{E}_2(v_2)$. Therefore, these nodes must have at least one preimage in the set $V_1 - \mathcal{E}_1(v_1)$ since $f$ (and thus $h$) is a surjection by assumption (a) in Definition 1. This is clearly a contradiction since any such preimage must have either a different feature vector, or a non-isomorphic port-consistent neighborhood. By a symmetric argument, using the surjection of map from $V_2$ to $V_1$, we conclude that $\ell_1 = \ell_2$. Note that $h$ did not tinker with the nodes that were outside the class $\mathcal{E}_1(v_1)$. Recycling the procedure for other nodes in $V_1 - \mathcal{E}_1(v_1)$ that might map under $f$ to a common image in $V_2$, we note that $h$ ends up being injective. Since $h$ remains surjective throughout the procedure, we conclude that $h$ is a bijection.

We now prove by induction that the corresponding nodes in port-locally isomorphic graphs have identical embeddings for any CPNGNN. Consider any such GNN with $L + 1$ layers parameterized by the sequence $\theta_1, L + 1 \equiv (\theta_1, \ldots, \theta_L, \theta_{L + 1})$. Since there exists a bijection $h$ such that any node $v_1 \in G_1$ has an identical local view (i.e., node features, and port-numbered neighbors) as $v_2 = h(v_1) \in G_1$, the updated embeddings for $v_1$ and $v_2$ are identical after the first layer. Assume that these embeddings remain identical after update from each layer $\ell \in \{2, 3, \ldots, L\}$. Since $v_1$ and $v_2$ have identical local views and have identical embedding from the $\ell$th layer, the updates for these nodes by the $(L + 1)th$ layer are identical. Therefore, $v_1$ and $v_2$ have identical embeddings. Since $h$ is a bijection, for every $v \in V_1$ there is a corresponding $h(v) \in V_2$ with the same embedding, and thus both $G_1$ and $G_2$ produce the same output with any permutation readout function. Our choice of $\theta_1, L + 1$ was arbitrary, so the result follows.

Proof of Proposition 4

Proof. We now show that there exist consistent port orderings such that CPNGNNs with permutation-invariant readout cannot decide several important graph properties: girth, circumference, diameter, radius, conjoint cycle, total number of cycles, and $k$-clique. The same result also holds for LU-GNNs where nodes do not have access to any consistent port numbering.

We first construct a pair of graphs that have cycles of different length but produce the same output embedding via the readout function. Specifically, we show that CPNGNNs cannot decide a graph having cycles of length $n$ from a cycle of length $2n$. We construct a counterexample for $n = 4$. Our first graph consists of two cycles of length 4 (each denoted by $S_4$), while the other graph is a cycle of length 8 (denoted by $S_8$). We associate identical feature vectors with nodes that have the same color, or equivalently, that are marked with the same symbol ignoring the subscripts and the underline. For example, $A_1$, $A_2$, $A_1$, and $A_2$ are all assigned the same feature vector. Moreover, we assign identical edge feature vectors to edges that have the same pair of symbols at the nodes.

Thus, we note that a bijection exists between the two graphs with node $X$ in the first graph corresponding to $X$ in the second graph such that both the nodes have identical features and indistinguishable port-ordered neighborhoods. Since, the two graphs have different girth, circumference, diameter, radius, and total number of cycles, it follows from Proposition 3 that CPNGNN cannot decide these properties. Note that the graph with two $S_4$ cycles is disconnected, and hence its radius (and diameter) is $\infty$.

We craft a separate construction for the remaining properties, namely, $k$-clique and conjoint cycle. The main idea is to replicate the effect of the common edge in the conjoint cycle via two identical components of another graph (that does not have any conjoint cycle) such that the components are cleverly aligned to reproduce the local port-ordered neighborhoods and thus present the same view to each node (see the adjoining figure). Specifically, each conjoint cycle is denoted by $G_1$, and the other graph that does not have any conjoint cycles by $G_2$. The graphs, being port-locally isomorphic, are indistinguishable by CPNGNN.

For the $k$-clique, we simply connect $A_1$ to $C_1$, $A_2$ to $C_2$, $A_1$ to $C_1$, and $A_2$ to $C_2$ via a new port 3 at each of these nodes. Doing so ensures that the new graphs are port-locally isomorphic as well. Adding these edges, we note that, unlike $G_2$, each conjoint cycle $G_1$ yields a 4-clique.

Proof of Proposition 5

Proof. We now demonstrate the representational limits of DimeNets. Specifically, we show two graphs that differ in several graph properties such as girth, circumference, diameter, radius, or total number of cycles. However, these graphs cannot be distinguished by DimeNets.
Note that DimeNet will be able to discriminate $S_8$ from the graph with cycles $S_4$: the graphs have not only identical local pairwise distances but also angles, so that their output embedding is same.

Our idea is to overlay the cycles $S_4$ and $S_8$ on a cube (see $G_3$ and $G_4$ - the graphs consist of only edges in bold). Doing so does not have any bearing on the graph properties. Since we orient the edges of these cycles along the sides of the cube, the local distances are identical. Moreover, by having $A_1B_1C_1D_1$ and $A_2B_2C_2D_2$ on a cube (see $G_3$ and $G_4$ - the graphs consist of only edges in bold), we ensure that each angle in $G_4$ is a right angle, exactly as in $G_3$. Thus, for each $X \in \{A, B, C, D\}$, each edge corresponds to a failure case for DimeNet, we need to construct a pair of non-isomorphic graphs that have not only identical local pairwise distances but also angles, so that their output embedding is same.

### Proof of Proposition 4

**Proof.** We now show that the complexity of the GNN may be bounded by the complexity of the computation trees. In other words, the worst case generalization bound over a set of graphs corresponds to having each graph be a single computation tree. Formally,

$$R_g = \sup_{T} \frac{1}{m} \sum_{j=1}^{m} \sigma_j \mathbb{E}_{T \sim w}(G_j; \Theta)$$

$$\leq \sup_{T} \frac{1}{m} \sum_{j=1}^{m} \sigma_j \mathbb{E}_{T \sim w}(G_j; \Theta)$$

where we invoked Jensen’s inequality to swap the expectation with supremum for our inequality (the operation is permissible since sup is a convex function). 

### Proof of Lemma 2

**Proof.** Our objective here is to bound the effect of change in weights from $(W_1, W_2)$ to $(W_1', W_2')$ on the embedding of the root node of our fixed tree (that has depth $L$). Since non-linear activation and permutation-invariant aggregation are both Lipschitz-continuous functions, and the feature vector at the root $x_L$ and the weights have bounded norm, the embedding at the root of the tree adapts to the embeddings from the subtrees.

Specifically, we note that the $l_2$-norm of difference of embedding vectors produced by $(W_1, W_2)$ and $(W_1', W_2')$ is

$$\Delta_L = \|T_L(W_1, W_2) - T_L(W_1', W_2')\|_2$$

$$= \left\| \phi\left(W_1 x_L + W_2 \rho\left( \sum_{j \in C(x_L)} g(T_{L-1,j}(W_1, W_2)) \right) \right) \right\|_2$$

$$\leq C_\phi \| (W_1 - W_1') x_L \|_2$$

$$+ C_\phi \| W_2 R(W_1, W_2, x_L) - W_2 R(W_1', W_2', x_L) \|_2. \quad (7)$$

Therefore, in order to find an upper bound for $\Delta_L$, we will bound the two terms in the last inequality separately. We first bound the second term using the sum of $\| W_2 R(W_1, W_2, x_L) - W_2 R(W_1', W_2', x_L) \|_2$ and $\| W_2 R(W_1, W_2, x_L) - W_2 R(W_1', W_2', x_L) \|_2$. Note that

$$\| W_2 R(W_1, W_2, x_L) - W_2 R(W_1', W_2', x_L) \|_2 \leq \| W_2 \|_2 \| R(W_1, W_2, x_L) - R(W_1', W_2', x_L) \|_2. \quad (8)$$

Since $g$ is $C_g$-Lipschitz, the branching factor of tree is $d$, and $\rho$ is $C_\rho$-Lipschitz, therefore, $R$ is $dC_gC_\rho$-Lipschitz. We will use this fact to bound (8). Specifically,

$$\| R(W_1, W_2, x_L) - R(W_1', W_2', x_L) \|_2 \leq C_\rho \| \sum_{j \in C(x_L)} \left( g(T_{L-1,j}(W_1, W_2)) - g(T_{L-1,j}(W_1', W_2')) \right) \|_2$$

$$\leq C_\rho \| \sum_{j \in C(x_L)} \left( g(T_{L-1,j}(W_1, W_2)) - g(T_{L-1,j}(W_1', W_2')) \right) \|_2$$

where we invoked Jensen’s inequality to swap the expectation with supremum for our inequality (the operation is permissible since sup is a convex function). 

$\square$
Using this with $\|W'_2\|_2 \leq B_2$ in (8), we immediately get
\[
\|W'_2R(W_1, W_2, x_L) - W'_2R(W'_1, W'_2, x_L)\|_2 \\
\leq B_2 C_\rho C_g \sum_{j \in C(x_L)} \Delta_{L-1,j} \\
\leq B_2 C_\rho C_g d \max_{j \in C(x_L)} \Delta_{L-1,j}.
\]
In other words, we bound the effect on each subtree of the root by the maximum effect across these subtrees. Combining this with $\|x_L\|_2 \leq B_x$, we note from (7) that
\[
\Delta_L \leq C_\rho B_x \|(W_1 - W'_1)\|_2 \\
+ C_\rho B_2 C_\rho C_g d \max_{j \in C(x_L)} \Delta_{L-1,j} \\
+ C_\rho \|(W_2 - W'_2)R(W_1, W_2, x_L)\|_2.
\]

**Proof of Lemma 3**

**Proof.** Note from (9) that in order for the change in embedding of the root (due to a small change in weights) to be small, we require that the last term in (9) is small. Toward that goal we bound the norm of permutation-invariant aggregation at the root node. Specifically, we note that
\[
\|R(W_1, W_2, x_L)\|_2 \\
= \rho \left( \sum_{j \in C(x_L)} g(T_{L-1,j}(W_1, W_2)) \right) \\
= \rho \left( \sum_{j \in C(x_L)} g(T_{L-1,j}(W_1, W_2)) - \rho(0) \right) \\
\leq C_\rho \left\| \sum_{j \in C(x_L)} g(T_{L-1,j}(W_1, W_2)) \right\|_2 \\
\leq C_\rho \sum_{j \in C(x_L)} \left\| g(T_{L-1,j}(W_1, W_2)) - g(0) \right\|_2 \\
\leq C_\rho C_g \sum_{j \in C(x_L)} \left\| T_{L-1,j}(W_1, W_2) \right\|_2 \\
\leq C_\rho C_g d \max_{j \in C(x_L)} \left\| T_{L-1,j}(W_1, W_2) \right\|_2,
\]
where the norm of the embedding produced by children $j$ of the root using weights $W_1$ and $W_2$ is given by
\[
\|T_{L-1,j}(W_1, W_2)\|_2 \\
= \left\| \phi(W_1 x_{L-1,j} + W_2 R(W_1, W_2, x_{L-1,j})) \right\|_2 \\
= \left\| \phi(W_1 x_{L-1,j} + W_2 R(W_1, W_2, x_{L-1,j})) - \phi(0) \right\|_2 \\
\leq C_\phi \left\| W_1 x_{L-1,j} + W_2 R(W_1, W_2, x_{L-1,j}) \right\|_2 \\
\leq C_\phi \left\| W_2 R(W_1, W_2, x_{L-1,j}) \right\|_2 \\
\leq C_\phi B_1 B_x + C_\phi B_2 \left\| R(W_1, W_2, x_{L-1,j}) \right\|_2.
\]

Also, since $\|\phi(x)\|_\infty \leq b$ for all $x \in \mathbb{R}^r$ (by our assumption), and $\|\phi(0)\|_2 \leq \sqrt{r}$, we obtain
\[
\left\| T_{L-1,j}(W_1, W_2) \right\|_2 \leq b \sqrt{r}. \tag{12}
\]
Combining (10) and (11), we get the recursive relationship
\[
\|R(W_1, W_2, x_L)\|_2 \\
\leq C_\rho C_g B_1 B_x d \\
+ C_\rho C_g B_2 d \max_{j \in C(x_L)} \left\| R(W_1, W_2, x_{L-1,j}) \right\|_2 \\
\leq C_\rho C_g B_1 B_x d \sum_{\ell=0}^{L-1} \left( C_\rho C_g B_2 d \right)^\ell \\
= C_\rho C_g B_1 B_x d \left( \frac{(Cd)^L - 1}{Cd - 1} \right). \tag{13}
\]
On the other hand, combining (10) and (11), we get
\[
\|R(W_1, W_2, x_L)\|_2 \leq bdC_\rho C_g \sqrt{r}. \tag{14}
\]
Taken together, (13) and (14) yield
\[
\|R(W_1, W_2, x_L)\|_2 \leq C_\rho C_g d \min \left\{ b \sqrt{r}, C_\rho B_1 B_x \left( \frac{(Cd)^L - 1}{Cd - 1} \right) \right\}. \tag{15}
\]

**Proof of Lemma 4**

**Proof.** Using the results from Lemma 2 and 3 we will simplify the bound on $\Delta_L$, i.e., the change in embedding due to a change in weights. We will then bound the change in probability (that the tree label is 1) $\Lambda_L$ in terms of $\Delta_L$, when we change not only the weights from $(W_1, W_2)$ to $(W'_1, W'_2)$ but also the local classifier parameters from $\beta$ to $\beta'$ (where $\beta$ and $\beta'$ are chosen from a bounded norm family). We show these steps below.
Plugging the bound on \( R \) we get
\[
\Delta_L \leq C_\phi B_x ||W_1 - W'_1||_2 + C d \max_{\beta \in C(x_L)} \Delta_{L-1,j} + C_\phi ||W_2 - W'_2||_2 R.
\]
Expanding the recursion, we note that
\[
\Delta_L \leq MB_x ||W_1 - W'_1||_2 + M\bar{R}||W_2 - W'_2||_2.
\]
where
\[
M = C_\phi \frac{(Cd)^L - 1}{Cd - 1}.
\]
Since \( ||A||_2 \leq ||A||_F \) for every matrix \( A \), we have
\[
\Delta_L \leq MB_x ||W_1 - W'_1||_F + M\bar{R}||W_2 - W'_2||_F.
\]
Now since sigmoid is 1-Lipschitz, we have
\[
\Lambda_L = |\psi(\beta^T T_L(W_1, W_2)) - \psi(\beta'^T T_L(W'_1, W'_2))| \leq ||\beta - \beta'||_2 \|T_L(W_1, W_2)\|_2 + B_\beta \Delta_L
\]
\[
\leq ||\beta - \beta'||_2 (C_\phi B_1 B_x + C_\phi B_2 \bar{R}) + B_\beta \Delta_L
\]
(19)

Proof of Lemma 5

Proof. Building on results from Lemmas 2-4, we will now show that the change in probability \( \Lambda_L \) can be bounded by \( \epsilon \), using a covering of size \( P \), where \( P \) depends on \( \epsilon \). Moreover, we show that \( \log P \) grows as \( \mathcal{O} \left( \log \left( \frac{1}{\epsilon} \right) \right) \) for sufficiently small values of \( \epsilon \). That is, we can ensure \( \Lambda_L \) is small by using a small covering.

We begin by noting that we can find a covering \( C \left( \beta, \frac{\epsilon}{3Z_\epsilon}, ||\cdot||_2 \right) \) of size
\[
\mathcal{N} \left( \beta, \frac{\epsilon}{3Z_\epsilon}, ||\cdot||_2 \right) \leq \left( 1 + \frac{6 Z B_\beta}{\epsilon} \right)^{\frac{1}{e}}.
\]
Thus, for any specified \( \epsilon \), we can ensure that \( \Lambda_L \) is at most \( \epsilon \) by finding matrix coverings \( C \left( W_1, \frac{\epsilon}{3MB_x B_\beta}, ||\cdot||_F \right) \) and \( C \left( W_2, \frac{\epsilon}{3MRB_\beta}, ||\cdot||_F \right) \). Using Lemma 8 from (Chen et al. 2020a), we obtain the corresponding bounds on their covering number. Specifically,
\[
N \left( W_1, \frac{\epsilon}{3MB_x B_\beta}, ||\cdot||_F \right) \leq \left( 1 + \frac{6 MB_x B_1 \sqrt{\tau}}{\epsilon} \right)^{\frac{1}{e}}
\]
\[
N \left( W_2, \frac{\epsilon}{3MRB_\beta}, ||\cdot||_F \right) \leq \left( 1 + \frac{6 MR B_2 \sqrt{\tau}}{\epsilon} \right)^{\frac{1}{e}}
\]
The product of all the covering numbers is bounded by
\[
P = \left( 1 + \frac{6B_\beta \max \{Z, M \sqrt{\tau} \max \{B_x B_1, RB_2\}\}}{\epsilon} \right)^{2 \epsilon + \frac{r}{2}}.
\]
Therefore, the class \( B(L, d, r, \beta, B_1, B_2, B_2) \) that maps a tree-structured input to the probability that the corresponding tree label is 1 can be approximated to within \( \epsilon \) by a covering of size \( P \). Moreover, when
\[
\epsilon < 6B_\beta \max \{Z, M \sqrt{\tau} \max \{B_x B_1, RB_2\}\},
\]
we obtain that \( \log P \) is at most
\[
3r^2 \log \left( \frac{12B_\beta \max \{Z, M \sqrt{\tau} \max \{B_x B_1, RB_2\}\}}{\epsilon} \right).
\]

Proof of Proposition 7

Proof. We are now ready to prove our generalization bound. Specifically, we invoke a specific form of Dudley’s entropy integral to bound the empirical Rademacher complexity \( \mathcal{R}_T(J_T) \) via our result on covering from Lemma 5 where recall that \( J_T \) maps each tree-label pair \( (t, y) \) to margin loss loss\(_y\)\( (-\tau(f_{\epsilon}(t; \Theta), y)) \).

It is straightforward to show that \( p \) is 2-Lipschitz in its first argument, and loss\(_y\)\( \leq \frac{1}{2} \)-Lipschitz. Therefore, we can approximate the class \( I \) that maps \( (t, y) \) to \( \tau(f_{\epsilon}(t; \Theta), y) \) by finding an \( \epsilon \)-cover of \( B \). Now, note that \( I \) takes values in the interval \([ -e, e ]\), where
\[
e = ||u||_2 ||T_L(W_1, W_2)||_2 \leq B_\beta Z.
\]
Using Lemma A.I. in (Bartlett et al. 2017), we obtain that
\[
\mathcal{R}_T(I) \leq \inf \alpha>0 \left( \frac{4\alpha}{\sqrt{m}} + \frac{12}{m} \int_{m}^{2e \sqrt{m}} \sqrt{\log N(I, \epsilon, ||\cdot||)} \right)\, \text{d}c
\]
where, using Lemma 5 we have
\[
\int_{m}^{2e \sqrt{m}} \sqrt{\log N(I, \epsilon, ||\cdot||)} \leq \sqrt{2e \sqrt{m}} \sqrt{\log N(B, \frac{\epsilon}{2}, \text{dist}(\cdot, \cdot))} \, \text{d}c
\]
\[
\leq \int_{m}^{2e \sqrt{m}} \sqrt{\log U} \leq 2e \sqrt{m} \sqrt{\log U} = 2\beta Z \sqrt{m \log U}
\]

Generalization and Representational Limits of Graph Neural Networks
with $\text{dist}$ being the combination of $\| \cdot \|_2$ and $\| \cdot \|_F$ norms used to obtain covering of size $P$ in Lemma 5 and $\log U$ is
\[3r^2\log \left( \frac{24B_\beta \max\{Z, M\sqrt{r} \max\{B_x B_1, B B_2\}\}}{\alpha} \right) .\]

Setting $\alpha = \sqrt{\frac{1}{m}}$, we immediately get
\[\hat{R}_T(I) \leq \frac{4}{m} + \frac{24B_\beta Z}{\sqrt{m}} \sqrt{3r^2 \log Q} ,\]
where
\[Q = 24B_\beta \sqrt{m} \max\{Z, M\sqrt{r} \max\{B_x B_1, B B_2\}\} .\]

We finally bound the complexity of class $\hat{R}_T(J_\gamma)$ by noting that $\text{loss}_\gamma$ is $\frac{1}{\gamma}$-Lipschitz, and invoking Talagrand’s lemma (Mohri et al., 2012):
\[\hat{R}_T(J_\gamma) \leq \frac{\hat{R}_T(I)}{\gamma} \leq \frac{4}{\gamma m} + \frac{24rB_\beta Z}{\gamma \sqrt{m}} \sqrt{3 \log Q} .\]

**Proof of Proposition 8**

**Proof.** We first convey some intuition. Suppose $|X| < 8$, and we assign a distinct index $z(x) \in \{1, 2, \ldots, 8\}$ to each message $x \in X$. Then, we can map each $x$ to $10^{-z(x)}$, i.e., obtain a decimal expansion which may be viewed as a one-hot vector representation of at most 10 digits. We would reserve a separate block of 10 digits for each port. This would allow us to disentangle the coupling between messages and their corresponding ports. Specifically, since the ports are all distinct, we can shift the digits in expansion of $x$ to the right by dividing by $10^p$, where $p$ is the port number of $x$. This allows us to represent each $(x, p)$ pair uniquely.

Formally, since $X$ is countable, there exists a mapping $Z : X \mapsto \mathbb{N}$ from $x \in X$ to natural numbers. Since $X$ has bounded cardinality, we know the existence of some $N \in \mathbb{N}$ such that $|X| < N$ for all $X$. Define $k = 10^{\left\lceil \log_{10} N \right\rceil}$. We define function $f$ in the proposition as $f(x) = k^{-Z(x)}$. We also take function $g$ in proposition to be $g(p) = 10^{-kN(p-1)}$. That is, we express the function $h$ as $h((x_1, p_1), \ldots, (x|P|, p|P|)) = \sum_{i=1}^{P} g(p_i) f(x_i).$