**Design Document**

**Abstract:**

**Section 1: Introduction**

**1.1: Preliminaries:**

**- Translations group**

**- Maybe (S\_nxS\_n)**

**- Universal Approximation via NN’s**

**- Invariant Networks for Group actions, example**

**1.2: Actual Introduction**

* **Problem definitions**
* **Naïve and non-naïve approaches**
* **Wasserstein Permutations Algorithm**
* **Wasserstein Hungarian Algorithm**
* **Primal and dual bounds**
* **Overview of our approach with networks**

**Section 2: Mathematical basis**

* **Primal and dual problems**
* **Explaining the Mathematical origins of each bound**
* **Discussing the strong duality and what does it mean**
* **Discussing loss and convexity**
* **Discussing algorithm variations and**

**Correction projections (Proof in appendix)**

**Section 3: Implementation**

* **Data generating process (Literally random)**
* **Architecture selection for base network and n^2log(n) network.**
* **Evaluation and initial benchmarks**
* **Maybe try O(n^2)\O(nlog(n)) dual solution estimation network.(Custom dual loss)**

**Section 4: Aggregation of multiple networks**

* **Explaining the Ensemble Method of bootstrap aggregation.**
* **Explain Aggregation, and discuss our specific interaction.**
* **Benchmarks**

**Section 5: Trying basic invariant networks (maybe)**

* **Considering batch translation invariance by training on distance matrices instead of actual vectors, or correcting mean of vectors to be the origin (more efficient?)**
* **invariant networks with input lexicographical Pre-Sorting.**
* **Benchmarks**

**Section 6:**

* **Conclusions**

**Section 7:**

* **Appendix & references + Useless claim proof** 😊