

## A Technical Appendices and Supplementary Material

### A.1 Additional details from Section 2

Here, we provide more details regarding the computation of  $\varepsilon$ -kernels. One simple algorithm to compute an  $\varepsilon$ -kernel is given in Algorithm 2. More sophisticated methods can be found in [10] and [43] but we highlight Algorithm 2 as it is most similar to the computation of the relaxed- $\varepsilon$ -kernel in Section 3.2. In summary, one may compute an  $\varepsilon$ -kernel by carefully choosing a set of directions  $\Omega$  and choose points according to the max-projection in that direction. We show the correctness of this algorithm in Theorem A.3. However, we first provide some auxiliary lemmas which we will use in the proof of Theorem A.3 as well as subsequent proofs.

**Lemma A.1.** *If  $P$  is an  $\alpha$ -fat point set, then for any  $u \in \mathbf{S}^{d-1}$ ,  $w_u(P) \geq 2\alpha$ .*

*Proof.* Suppose  $P$  is an  $\alpha$ -fat point set so  $\alpha\mathbb{C} \subseteq \text{CH}(P)$ . For any  $u \in \mathbf{S}^{d-1}$  and  $p \in \text{CH}(P)$ ,  $\langle u, p \rangle \geq \|u\| \|p\|$ . Since  $\alpha\mathbb{C} \subseteq \text{CH}(P)$ ,  $\|p\| \geq \alpha$  so  $\langle u, p \rangle \geq \alpha$  and  $\max_{p \in P} \langle u, p \rangle \geq \alpha$ . Furthermore,  $\min_{p \in P} \langle u, p \rangle = -\max_{p \in P} \langle -u, p \rangle = -\max_{p \in P} \langle -u, p \rangle \leq -\alpha$ . Then

$$w_u(P) = \max_{p \in P} \langle u, p \rangle - \min_{p \in P} \langle u, p \rangle \geq 2\alpha$$

□

**Lemma A.2.** *Let  $S \subseteq \mathbb{C}_d$ . Given two unit vectors  $u, u' \in \mathbb{C}_d$  such that  $\|u - u'\| \leq \lambda$ ,  $w_u(S) \geq w_{u'}(S) - 2\lambda\sqrt{d}$ .*

*Proof.* Given any  $x \in S$ ,

$$|\langle x, u \rangle - \langle x, u' \rangle| = |\langle u - u', x \rangle| \leq \|u - u'\| \|x\| \leq \lambda\sqrt{d}.$$

where the final  $\sqrt{d}$  is because  $x \in [-1, 1]^d$ . Then

$$\begin{aligned} w_{u'}(S) &= \max_{s \in S} \langle u', s \rangle - \min_{s \in S} \langle u', s \rangle \\ &\geq \langle u', \arg\max_{s \in S} \langle u, s \rangle \rangle - \min_{s \in S} \langle u', s \rangle \\ &\geq \langle u, \arg\max_{s \in S} \langle u, s \rangle \rangle - \lambda\sqrt{d} - \min_{s \in S} \langle u', s \rangle \\ &\geq \langle u, \arg\max_{s \in S} \langle u, s \rangle \rangle - \lambda\sqrt{d} - \langle u', \arg\min_{s \in S} \langle u, s \rangle \rangle \\ &\geq \langle u, \arg\max_{s \in S} \langle u, s \rangle \rangle - \lambda\sqrt{d} - (\langle u, \arg\min_{s \in S} \langle u, s \rangle \rangle + \lambda\sqrt{d}) \\ &\geq w_u(S) - 2\lambda\sqrt{d}. \end{aligned}$$

□

**Theorem A.3.** *Let  $P \subseteq \mathbb{C} = [-1, 1]^d$  be an  $\alpha$ -fat point set and let  $0 < \varepsilon < \frac{1}{3}$ . Suppose  $\Omega$  is an  $\frac{\alpha\varepsilon}{4\sqrt{d}}$ -net for  $\mathbf{S}^{d-1}$  so  $|\Omega| = O\left(\left(\frac{1}{\frac{\alpha\varepsilon}{4\sqrt{d}}}\right)^{d-1}\right)$ ,  $\forall u, u' \in \Omega$ ,  $\|u - u'\| \leq \frac{1}{\left(\frac{\alpha\varepsilon}{4\sqrt{d}}\right)^{d-1}}$ , and for all  $u \in \Omega$ ,  $-u \in \Omega$ . Suppose that  $\forall u \in \Omega$ ,  $-u \in \Omega$ . Then the set of points  $Q$  constructed via Algorithm 2 is an  $\varepsilon$ -kernel for  $P$ .*

---

#### Algorithm 2 Computation of $\varepsilon$ -kernels

---

**Require:**  $P \subseteq \mathbb{R}^d, k$   
 $\Omega$  is a set of  $k$  directions  
Initialize  $Q = \{\}$   
**for**  $u \in \Omega$  **do**  
     $Q.append(\arg\max_{p \in P} \langle u, p \rangle)$   
**end for**  
**return**  $Q = \{q_u : u \in \Omega\}$

---

507 **Proof of Theorem A.3.** Let  $u \in \mathbf{S}^{d-1}$ . Let  $\lambda = \frac{\alpha\varepsilon}{4\sqrt{d}}$ . Since  $\Omega$  is an  $\lambda$ -net for  $\mathbf{S}^{d-1}$ , there is a  
 508  $u' \in \Omega$  such that  $\|u - u'\| \leq \lambda$ . Therefore, by Lemma A.2

$$w_u(Q) \geq w_{u'}(Q) - 2\lambda\sqrt{d} \geq w_{u'}(P) - 2\lambda\sqrt{d} \geq w_u(P) - 4\lambda\sqrt{d}.$$

509 Therefore,

$$\frac{w_u(Q)}{w_u(P)} \geq 1 - \frac{4\lambda\sqrt{d}}{w_u(P)} \geq 1 - \frac{4\lambda\sqrt{d}}{\alpha} = 1 - \varepsilon$$

510 where the second inequality comes from Lemma A.1. Thus,  $w_u(Q) \geq (1 - \varepsilon)w_u(P)$ .

## 511 A.2 Additional details from Section 4

512 We provide an example of how relaxed- $\varepsilon$ -kernels can be used to approximate minimum width  
 513 enclosing annulus. However, we will first need show in Theorem A.5 how Theorem 4.4 can be used  
 514 to compute relaxed- $\varepsilon$ -kernels for fractional powers of polynomials. Our proof of Theorem A.5 builds  
 515 on the following lemma, originally proved by [2].

516 **Lemma A.4.** Let  $0 < \varepsilon < 1$  be a parameter,  $r \geq 2$  and let  $\delta = (\varepsilon/2(r-1))^r$ . If we have  
 517  $0 \leq a \leq A \leq B \leq b$  and  $B - A \geq (1 - \delta)(b - a)$ , then

$$B^{1/r} - A^{1/r} \geq (1 - \varepsilon)(b^{1/r} - a^{1/r})$$

518 **Theorem A.5.** Let  $\mathcal{F}$  be a family of  $(d + p)$  variate polynomials which admit a linearization of  
 519 dimension  $m$  as in Theorem 4.4. Additionally, suppose for every  $f_i$ ,  $f_i(x) \geq 0$  for all  $x \in \mathbb{R}^d$ . Let  
 520  $r \geq 2$ ,  $\delta = (\varepsilon/2(r-1))^r$ , and let  $\mathcal{G} = \{g_i\}$  be a relaxed- $\delta$ -kernel of  $\mathcal{F}$ . Then  $\mathcal{G}_{1/r} = \{g_i^{1/r}\}$  is a  
 521 relaxed- $\varepsilon$ -kernel of  $\mathcal{F}_{1/r} = \{f_i^{1/r}\}$ .

522 *Proof.* Let  $\delta = (\varepsilon/2(r-1))^r$  where  $r \geq 2$  is an integer. Let  $\mathcal{G}$  be a relaxed- $\delta$ -kernel of  $\mathcal{F}$  so  
 523  $\vec{\mathcal{E}}_{\mathcal{G}}(x) \subseteq \vec{\mathcal{E}}_{\mathcal{F}}(x)$  and  $\vec{\mathcal{E}}_{\mathcal{G}}(x)$  is within  $\varepsilon \cdot \mathcal{E}_{\mathcal{F}}(x)$ -Hausdorff distance of  $\vec{\mathcal{E}}_{\mathcal{F}}(x)$ . Since  $f \in \mathcal{F}$  is  
 524 positive, for any  $x \in \mathbb{R}^d$ , we know that

$$0 \leq \min_{f \in \mathcal{F}}(x) \leq \min_{g \in \mathcal{G}}(x) \leq \max_{g \in \mathcal{G}}(x) \leq \max_{f \in \mathcal{F}}(x).$$

525 Additionally, by the definition of relaxed- $\varepsilon$ -kernels,  $\max_{g \in \mathcal{G}} g(x) - \min_{g \in \mathcal{G}} g(x) \geq (1 -$   
 526  $\varepsilon)(\max_{f \in \mathcal{F}}(x) - \min_{f \in \mathcal{F}}(x))$ . Then, we apply Lemma A.4 to get

$$\max_{g \in \mathcal{G}_{1/r}} g(x)^{1/r} - \min_{g \in \mathcal{G}_{1/r}} g(x)^{1/r} \geq (1 - \varepsilon) \left( \max_{g \in \mathcal{F}_{1/r}} f(x)^{1/r} - \min_{f \in \mathcal{F}_{1/r}} f(x)^{1/r} \right).$$

527 □

528 Now, we are prepared to provide an example of Theorem A.5 can be used to approximate minimum  
 529 enclosing annulus for points in  $\mathbb{R}^2$ . Given  $P = \{p_1, \dots, p_N\} \subseteq \mathbb{R}^2$  and any  $x \in \mathbb{R}^2$ , finding the  
 530 width of the a minimum enclosing spherical annulus centered at  $x$  is

$$w(x, P) = \max_{p \in P} \|x - p\| - \min_{p \in P} \|x - p\|.$$

531 We define the set of functions  $\mathcal{F} = \{f_p(x) = \|x - p\| : p \in P\}$ . Notice then that  $\mathcal{E}_{\mathcal{F}}(x) = w(x, P)$   
 532 and the width of the minimum enclosing spherical shell is exactly  $\min_{x \in \mathbb{R}^d} \mathcal{E}_{\mathcal{F}}(x)$ . Similarly, the  
 533 optimal center for the minimum enclosing spherical shell is  $\operatorname{argmin}_{x \in \mathbb{R}^d} \mathcal{E}_{\mathcal{F}}(x)$ . By Theorem A.5, a  
 534 relaxed- $\varepsilon$ -kernel for  $\mathcal{F}' = \{\|x - p\|^2 : p \in P\}$  translates to relaxed- $\varepsilon$ -kernel for  $\mathcal{F}$ .

535 Suppose  $x = (x_1, x_2) \in \mathbb{R}^2$  and  $p_i = (p_{i,1}, p_{i,2}) \in \mathbb{R}^2$  where  $p_i \in P$ . Given  $f_i \in \mathcal{F}'$ ,  $f_i =$   
 536  $x_1^2 + x_2^2 - 2p_{i,1}x_1 - 2p_{i,2}x_2 + p_{i,1}^2 + p_{i,2}^2$ .  $f_i$  admits a linearization of dimension 3 as follows:

$$\begin{aligned} \psi_0(p_i) &= p_{i,1}^2 + p_{i,2}^2 & \psi_1(p_i) &= -2p_{i,1} & \psi_2(p_i) &= -2p_{i,2} & \psi_3(p_i) &= 1 \\ \varphi_1(x) &= x_1 & \varphi_2(x) &= x_2 & \varphi_3(x) &= x_1^2 + x_2^2 \end{aligned}$$

537 By Theorem 4.4, we can compute a relaxed- $\varepsilon$ -kernel for the dual space via Algorithm 1 in  $\mathbb{R}^4$  and  
 538 map back to relaxed- $\varepsilon$ -kernel for  $\mathcal{F}'$ ,  $\mathcal{Q}_{\varepsilon}$ . Then  $\mathcal{Q}_{\varepsilon, 1/r}$  is a relaxed- $\varepsilon$ -kernel for  $\mathcal{F}$  by Theorem A.5  
 539 Computing  $\min_{x \in \mathbb{R}^2} \mathcal{E}_{\mathcal{Q}_{\varepsilon, 1/r}}(x)$  as well as  $\operatorname{argmin}_{x \in \mathbb{R}^2} \mathcal{E}_{\mathcal{Q}_{\varepsilon, 1/r}}(x)$  outputs the width and center of  
 540 the minimum enclosing annulus, respectively.

Table 3: **Dataset details for approximating relaxed- $\varepsilon$ -kernels.**  $|P_{\text{train}}|$  and  $|P_{\text{test}}|$  refer to the size of the train and test point clouds, respectively. The first four columns describe synthetic datasets while the last two columns describe two real datasets. For the ‘Uniform ball’ dataset, we sample point clouds uniformly from a  $d$ -dimensional ball. Note that for ‘Uniform Ball’, we will sometimes call it ‘Uniform Disk’ for point clouds in  $\mathbb{R}^2$ . For ‘Ellipse’, we sample from a randomly scaled and rotated point cloud. For ‘Single Gaussians’, we sample point clouds from a Gaussian with a random standard deviation and ‘Gaussian Mixture’ refers to point clouds sampled from two to five randomly placed Gaussian clusters. Note that for ModelNet, we apply random scaling to the point cloud so that the bounds are between  $[-5, 5]$ .

	Synthetic				Real	
	Uniform Ball	Ellipse	Single Gaussian	Gaussian Mixture	ModelNet [40]	SQUID [25]
$ P_{\text{train}} $	500	500	500	500	200	350
$ P_{\text{test}} $	500	500	500	500	200	350
# train point clouds	3000	3000	3000	3000	2048	774
# test point clouds	750	750	750	750	2048	332
Input dim.	2,3,5	2,3,5	2,3,5	2,3,5	3	2
Bounds	$[-5, 5]$	$[-5, 5]$	$[-5, 5]$	$[-5, 5]$	$[-5, 5]$	$[0, 450]$

Table 4: **Dataset details for extent measure approximation tasks.**  $|P_{\text{train}}|$  and  $|P_{\text{test}}|$  refer to the size of the train and test point clouds, respectively. The ‘Uniform Ball’ dataset is used for the minimum enclosing ball task and consists of point clouds randomly sampled from a  $d$ -dimensional ball. The ‘Uniform Ellipse’ dataset is used for the minimum enclosing ellipse task and consists of point clouds sampled from randomly scaled and rotated ellipses. The ‘Uniform Annulus’ dataset is used for the minimum enclosing annulus task and consists of point clouds sampled from annuli with widths from 0.1 to 3.

	Synthetic			Real	
	Uniform Ball	Uniform Ellipse	Uniform Annulus	ModelNet [40]	SQUID [25]
$ P_{\text{train}} $	200	100	100	200	350
$ P_{\text{test}} $	200	100	100	200	350
# train point clouds	3000	3000	3000	2048	774
# test point clouds	3000	3000	3000	2048	332
Input dim.	2, 3	2	2	3	2
Bounds	$[-5, 5]$	$[-5, 5]$	$[-5, 5]$	$[-5, 5]$	$[0, 450]$

### 541 A.3 Additional experimental details and results

542 All models are implemented in PyTorch and train on 8 NVIDIA RTX A6000 GPUs. Additionally, all  
543 models are trained using the ADAM optimizer[19] provided in PyTorch and trained using a learning  
544 rate of 0.001. Relaxed- $\varepsilon$ -kernel networks are trained for 200 epochs while all extent-measure models  
545 are trained for 500 epochs. Additionally, for the processor for the encode-decode-process models  
546 (used for extent measure tasks), we fix the processor to take input from  $\mathbb{R}^5$ , have three blocks (either  
547 SumFormer or transformer) and output 150 points from  $\mathcal{N}_{\phi_\varepsilon}$ . We choose this configuration as these  
548 hyperparameters had the best performance on the relaxed- $\varepsilon$ -coreset task in  $\mathbb{R}^5$ . Detailed dataset  
549 specifications for the relaxed- $\varepsilon$ -kernel experiments are given in Table 3 and those for extent-measure  
550 tasks are given Table 4. All code (including models, hyperparameter settings, and synthetic dataset  
551 generation) is available.

552 **Loss functions.** Here, we list all loss functions used to train each task given an input point set  $P$ .

- 553 • **Relaxed- $\varepsilon$ -kernel:** To train a model to approximate relaxed- $\varepsilon$ -kernel, at each epoch  $t$ , we  
554 first sample a random set of 100 directions  $\Omega_t$ . We then compute the difference between  
555 the max projection of  $P$  and  $\mathcal{N}_{\phi_\varepsilon}(P)$  for  $d \in \Omega_t$  as well as the difference between the min  
556 projection of  $P$  and  $\mathcal{N}_{\phi_\varepsilon}(P)$  for  $d \in D_t$ :

$$\mathcal{L}_\varepsilon(P, \mathcal{N}_{\phi_\varepsilon}(P), D_t) = \frac{1}{|D_t|} \sum_{u \in \Omega_t} \left( \left| \max_{p \in P} \langle u, p \rangle - \max_{q \in \mathcal{N}_{\phi_\varepsilon}(P)} \langle u, q \rangle \right| + \left| \min_{p \in P} \langle u, p \rangle - \min_{q \in \mathcal{N}_{\phi_\varepsilon}(P)} \langle u, q \rangle \right| \right)$$

- 557 • **Minimum enclosing ball:** We aim to approximate the radius of the minimum enclosing ball.  
558 In order to validate that we have actually learned the ball, we also predict the center of ball.  
559 Therefore, given an input point set in  $\mathbb{R}^d$ , we configure  $\mathcal{N}_{\text{extent}}$  such that  $\mathcal{N}_{\text{extent}}(P) \in \mathbb{R}^{d+1}$   
560 where the first  $d$  coordinates,  $\mathcal{N}_{\text{extent}}(P)_{1:d}$  represent the center and the last coordinate  
561  $\mathcal{N}_{\text{extent}}(P)_{d+1}$  represents the radius. Given the ground truth center  $c \in \mathbb{R}^d$  and radius  
562  $r \in \mathbb{R}$ , the loss function we use to train  $\mathcal{N}_{\text{extent}}$  is

$$\mathcal{L}(c, r, \mathcal{N}_{\text{extent}}(P)) = \|c - \mathcal{N}_{\text{extent}}(P)_{1:d}\| + (r - \mathcal{N}_{\text{extent}}(P)_{d+1})^2$$

- 563 • **Minimum enclosing ellipse:** Recall that we aim to predict the minimum area enclosing  
564 ellipse. First, we note that we center all data at the origin for the minimum enclosing  
565 ellipse. Unlike the minimum enclosing ball, we do not predict the center for the ellipse.  
566 Given an input point set  $P \subseteq \mathbb{R}^d$ , we configure  $\mathcal{N}_{\text{extent}}$  such that  $\mathcal{N}_{\text{extent}}(P) \in \mathbb{R}^3$  where  
567  $\mathcal{N}_{\text{extent}}(P)_1$  represents the major radius,  $\mathcal{N}_{\text{extent}}(P)_2$  represents the minor radius, and  
568  $\mathcal{N}_{\text{extent}}(P)_3$  represents the angle of rotation. Given the ground truth major radius  $r_{\text{maj}} \in \mathbb{R}$ ,  
569 minor radius  $r_{\text{min}} \in \mathbb{R}$  and angle of rotation  $\theta \in \mathbb{R}$ , the loss function we use to train  $\mathcal{N}_{\text{extent}}$   
570 is

$$\begin{aligned} \mathcal{L}(r_{\text{maj}}, r_{\text{min}}, \theta, \mathcal{N}_{\text{extent}}(P)) &= (r_{\text{maj}} - \mathcal{N}_{\text{extent}}(P)_1)^2 + (r_{\text{min}} - \mathcal{N}_{\text{extent}}(P)_2)^2 \\ &\quad + (\sin(\theta) - \sin(\mathcal{N}_{\text{extent}}(P)_3))^2 + (\cos(\theta) - \cos(\mathcal{N}_{\text{extent}}(P)_3))^2 \end{aligned}$$

- 571 • **Minimum enclosing annulus:** We aim to predict the minimum width of the minimum  
572 enclosing annulus, which can be computed by predicting the inner and outer radii of the annu-  
573 lus and then taking their difference. Like the minimum enclosing ball, we predict the center  
574 of the annulus. Given an input point set  $P \subseteq \mathbb{R}^d$ ,  $\mathcal{N}_{\text{extent}}(P) \in \mathbb{R}^{d+2}$  where  $\mathcal{N}_{\text{extent}}(P)_{1:d}$   
575 represents the center,  $\mathcal{N}_{\text{extent}}(P)_{d+1}$  represents the inner radius, and  $\mathcal{N}_{\text{extent}}(P)_{d+2}$  repre-  
576 sents the outer radius. Then given the ground truth center  $c \in \mathbb{R}^d$ , inner radius  $r_{\text{inner}} \in \mathbb{R}$   
577 and outer radius  $r_{\text{outer}} \in \mathbb{R}$ , the loss function we use to train  $\mathcal{N}_{\text{extent}}$  is

$$\begin{aligned} \mathcal{L}(c, r_{\text{inner}}, r_{\text{outer}}, \mathcal{N}_{\text{extent}}(P)) &= \|c - \mathcal{N}_{\text{extent}}(P)_{1:d}\| + (r_{\text{inner}} - \mathcal{N}_{\text{extent}}(P)_{d+1})^2 \\ &\quad + (r_{\text{outer}} - \mathcal{N}_{\text{extent}}(P)_{d+2})^2 \end{aligned}$$

### 578 A.3.1 $L_1$ versus softmax normalization

579 Although our theory uses  $L_{1,\text{col}}$  normalization to produce probability distributions, we find that in  
580 practice softmax yields substantially better results. This is likely because softmax provides smooth  
581 non-zero gradients whereas  $L_{1,\text{col}}$  often creates flat regions. In fact, when using ReLU followed by  
582  $L_{1,\text{col}}$  normalization, the output of  $\phi_\varepsilon$  frequently collapses to all zeros early in training. To mitigate  
583 this, we replace ReLU with LeakyReLU and compare  $L_{1,\text{col}}$  against softmax for the relaxed- $\varepsilon$ -kernel  
584 approximating NN trained on the mixed synthetic dataset. As reported in Table 5, models using  
585 softmax consistently outperform those using  $L_{1,\text{col}}$  across all input dimensions.

### 586 A.3.2 Additional experiments for approximating relaxed- $\varepsilon$ -kernels

587 First, we note that we include all complete tables from the main text with error bars in Table 6 and  
588 Table 7. We additionally include results on datasets in higher-dimensional synthetic datasets in  $\mathbb{R}^5$  in  
589 Table 8. We see largely the same trends, where the  $\mathcal{S}_\varepsilon$  performs comparably to  $\mathcal{T}_\varepsilon$  on in-distribution  
590 data and then  $\mathcal{S}_\varepsilon$  performs much better than  $\mathcal{T}_\varepsilon$  on out of distribution data. We also provide more  
591 extensive results in Tables 9, 10, and 11, where we detail the performance of each model on different  
592 types of synthetic and real datasets.

For the relaxed- $\varepsilon$ -kernel tasks, we also provide several sensitivity analyses showing the effect of training time, input dimension, fatness of point sets, and the size of relaxed- $\varepsilon$ -kernel on the direction error ( $\mathcal{E}_{\text{dir}}$ ). For each of these sensitivity analyses (excluding training time) we also provide a baseline comparison to an implementation of the relaxed- $\varepsilon$ -kernel algorithm described in Algorithm 1 given  $\varepsilon = 0.1$ .

**Effect of training time.** We record the effect of training time for  $\mathcal{N}_{\phi_\varepsilon}$  (instantiated with SumFormer and transformer) on  $\mathcal{E}_{\text{dir}}$  for output relaxed- $\varepsilon$ -kernels of size 16 and 64 in  $\mathbb{R}^2$  in Figure 9 and Figure 10. We train on uniform ball (called in uniform disk for datasets in  $\mathbb{R}^2$ ) and test on both uniform ball (disk), mixed synthetic data, and SQUID. We notice that the SumFormer reaches lower error early and if trained longer, tends to have increasing OOD error (on SQUID). First, this justifies our choice of training for 200 epochs for the relaxed- $\varepsilon$ -kernel. Second, we notice that the SumFormer achieves lower OOD error faster than the Transformer – suggesting the advantage of the alignment with the relaxed- $\varepsilon$ -kernel framework when it comes to optimization.

**Effect of input dimension.** From Theorem 3.4 we know that the accuracy of the approximation of the relaxed- $\varepsilon$ -kernel will depend on the dimension of the input point cloud. We verify this empirically in Figure 5 across different sizes of  $\varepsilon$ -kernels. For this experiment, we train and test each model on the point clouds with 500 points sampled uniformly from a ball in  $\mathbb{R}^D$  where  $D \in \{2, 3, 5\}$ . We can see in Figure 5 that  $\mathcal{E}_{\text{dir}}$  increases as the input dimension of the point cloud increases.

**Effect of  $\alpha$ -fatness.** Similar to the relationship between error and input point cloud dimension, we know from Theorem 3.4 that the accuracy of the approximation of the relaxed- $\varepsilon$ -kernel will depend on the  $\alpha$ -fatness of the input point set. To verify this, we train both the SumFormer and Transformer model on point clouds with 500 points sampled uniformly from a balls in  $\mathbb{R}^3$  and  $\mathbb{R}^2$  (disks in the case of  $\mathbb{R}^2$ ). We then test on point clouds of 100 points sampled from ellipsoids which have their minor axes scaled to simulate point clouds with a range of  $\alpha$ -fatness. The results are reported in Figure 6. We see in practice, the error is fairly stable w.r.t  $\alpha$ -fatness although in  $\mathbb{R}^3$  we do see some mild increases in error as  $\alpha$ -increases.

**Effect of output size.** In order to examine the effect of the output relaxed- $\varepsilon$ -kernel size for the quality of the approximation, we plot the normalized output size (i.e. number of relaxed- $\varepsilon$ -kernel points/total number of input points) against the directional width error. See Figure 7. As expected, we see that the error decreases as the output size increases.

**Effect of input point set size.** We also examine the out-of-distribution capabilities of the models (in terms of generalizing to larger point sets. For this experiment, we train each model on the point clouds with 500 points sampled uniformly from a ball in  $\mathbb{R}^D$  where  $D \in \{2, 3, 5\}$  and then test the models on much larger point clouds (up to 1000 points per cloud) sampled from the uniform ball. The results are reported in Figure 8 and we find that the SumFormer model generalizes especially well to out-of-distribution point set sizes (maintains low error).

### A.3.3 Additional experiments for approximation of extent measures

Here, we will include the full results from the main test (which include the percentage of points excluded from each covering object). We also include some sensitivity analysis of how the accuracy changes for each dataset as we increase the training time. The full results each extent measure task are included in Tables 12, 13, 14, and 15 for minimum enclosing ball in  $\mathbb{R}^2$ , minimum enclosing ball in  $\mathbb{R}^3$ , minimum enclosing ellipse in  $\mathbb{R}^2$ , and minimum enclosing annulus  $\mathbb{R}^2$ , respectively. Notice in those results, we also track the proportion of points excluded. We also examine how the test error for each extent measure task changes as we train past 500 epochs in Figures 11, 12 and 13 for minimum enclosing ball, minimum enclosing ellipse, and minimum enclosing annulus, respectively. Note that these tasks were all trained using the synthetic data per task (described in the main task). We see that the test error does not change much after 500 epochs, justifying our choice in training time.

## 640 A.4 Proofs

### 641 A.4.1 Proofs from Section 3

642 The following lemma will be important to show that relaxed  $\varepsilon$ -kernels for approximating well-behaved  
643 measurements of the original point set  $P$ .

644 **Lemma A.6.** Suppose  $Q$  is a relaxed- $\varepsilon$ -kernel of  $P$ . Let  $\hat{P} = P \cup Q$ . Then (a)  $Q$  is a  $\varepsilon$ -approximation  
645 of  $\hat{P}$  and (b)  $P$  is an  $\varepsilon$ -approximation of  $\hat{P}$ .

646 **Proof of Lemma A.6** Let  $u \in \mathbb{S}^{d-1}$ . To show that  $Q$  is an  $\varepsilon$ -approximation of  $\hat{P}$ , we must show  
647 that

$$\max_{q \in Q} \langle q, u \rangle - \varepsilon w_u(Q) \leq \max_{\hat{p} \in \hat{P}} \langle \hat{p}, u \rangle \leq \max_{q \in Q} \langle q, u \rangle + \varepsilon w_u(Q)$$

648 and

$$\min_{q \in Q} \langle q, u \rangle - \varepsilon w_u(Q) \leq \min_{\hat{p} \in \hat{P}} \langle \hat{p}, u \rangle \leq \min_{q \in Q} \langle q, u \rangle + \varepsilon w_u(Q)$$

649 We know that  $\max_{\hat{p} \in \hat{P}} \langle \hat{p}, u \rangle = \max\{\max_{p \in P} \langle p, u \rangle, \max_{q \in Q} \langle q, u \rangle\}$ . We know that  
650  $\max_{q \in Q} \langle q, u \rangle - \varepsilon w_u(Q) \leq \max_{p \in P} \langle u, p \rangle \leq \max_{q \in Q} \langle u, q \rangle + \varepsilon w_u(Q)$  because  $Q$  is an  $\varepsilon$ -relaxed-  
651 kernel of  $P$ . Clearly,  $\max_{q \in Q} \langle q, u \rangle - \varepsilon w_u(Q) \leq \max_{q \in Q} \langle q, u \rangle \leq \max_{q \in Q} \langle q, u \rangle + \varepsilon w_u(Q)$  so

$$\max_{q \in Q} \langle q, u \rangle - \varepsilon w_u(Q) \leq \max\{\max_{p \in P} \langle p, u \rangle, \max_{q \in Q} \langle q, u \rangle\} = \max_{\hat{p} \in \hat{P}} \langle \hat{p}, u \rangle \leq \max_{q \in Q} \langle q, u \rangle + \varepsilon w_u(Q).$$

652 We can do something similar for  $\min_{q \in Q} \langle q, u \rangle$  and  $\min_{\hat{p} \in \hat{P}} \langle \hat{p}, u \rangle$ .

653 The same argument will hold for (b).

**Proof of Theorem 3.2.** Let  $\hat{P} = P \cup Q$ . Let  $\mu$  be a faithful measure and let  $c$  be the constant in for  
the faithful measure,  $\mu$ . By Lemma A.6 (a),  $Q$  is  $\varepsilon$ -kernel of  $\hat{P}$ , meaning  $Q$  is a  $c\varepsilon$ -coreset of  $\hat{P}$ . By  
Lemma A.6 (b),  $P$  is  $\varepsilon$ -kernel of  $\hat{P}$ , meaning  $P$  is a  $c\varepsilon$ -coreset of  $\hat{P}$ . It then follows that

$$\frac{1 - c\varepsilon}{1 + c\varepsilon} \mu(Q) \leq \mu(P) \leq \frac{1 + c\varepsilon}{1 - c\varepsilon} \mu(Q).$$

For  $c\varepsilon < 1/3$ , it is easy to verify that this implies

$$(1 - 3c\varepsilon) \mu(Q) \leq \mu(P) \leq (1 + 3c\varepsilon) \mu(Q),$$

654 hence the theorem holds with  $c' = 3c$ .

655 **Proof of Theorem 3.3** First, we make two important observations regarding the points in the  
656 output relaxed- $\varepsilon$ -kernels,  $Q$ , computed by Algorithm 1 and their relationship to the set of chosen  
657 directions  $\Omega$ . First, suppose we are given  $\varepsilon > 0$  and  $A = \{a_1, \dots, a_N\} \subseteq \mathbb{R}$  such that  $a_i = \langle u, p_i \rangle$ ,  
658 let  $\mathbf{a} = [a_1, \dots, a_N]^T$ . Recall from Algorithm 1,  $\rho_\varepsilon$  is defined as

$$\rho_\varepsilon(\mathbf{a}) = \begin{pmatrix} \text{ReLU}(a_1 + \varepsilon \cdot w(A) - M_A) \\ \text{ReLU}(a_2 + \varepsilon \cdot w(A) - M_A) \\ \vdots \\ \text{ReLU}(a_n + \varepsilon \cdot w(A) - M_A) \end{pmatrix}$$

659 where  $M_A = \max(A)$  and  $w(A) = \max(A) - \min(A)$ . Note  $\text{ReLU}(a_i + \varepsilon \cdot w(A) - M_A) > 0$  only  
660 when  $a_i$  is within  $\varepsilon \cdot w(A)$  of the maximum value,  $M_A$ . Additionally, we can see  $\rho_\varepsilon(\mathbf{a}) / \|\rho_\varepsilon(\mathbf{a})\|_1$  as  
661 a probability vector over  $\mathbf{a}$  which only has non-zero values corresponding to those  $a_i$  that are within  
662  $\varepsilon \cdot w(A)$  of  $M_A$ . If we take  $(\rho_\varepsilon(\mathbf{a}) / \|\rho_\varepsilon(\mathbf{a})\|_1)^T \mathbf{a}$ , we observe that  $\text{softmax}(\rho_\varepsilon(\mathbf{a}))^T \mathbf{a}$  is a convex  
663 combination of the points in  $A$ .

664 Thus, given  $u \in \Omega$  and  $q_u \in Q$ , we make the following important observations about  $q_u$ .

665 **Observation A.7.** Given  $u \in \Omega$ ,  $q_u \in Q$  is the convex combination of a subset of points  $p_{I_1}, \dots, p_{I_{r_u}}$ .  
666 For each such point  $p_{I_i}$ ,  $\langle u, p_{I_i} \rangle \in [\max_{p \in P} \langle u, p \rangle - \varepsilon w_u(P), \max_{p \in P} \langle u, p \rangle]$ .

667 **Observation A.8.** Given  $u \in \Omega$ ,  $\langle u, q_u \rangle \in [\max_{p \in P} \langle u, p \rangle - \varepsilon w_u(P), \max_{p \in P} \langle u, p \rangle]$ .

Our goal is to show that for any  $u \in \mathbf{S}^{d-1}$ , the  $u$ -projection  $\vec{w}_u(Q)$  is an  $\varepsilon$ -approximation of the interval ( $u$ -projection)  $\vec{w}_u(P)$ . First, note that by construction, each point in  $Q$  is a convex combination of a subset of points in  $P$  so  $Q \subseteq \text{CH}(P)$ . This implies that  $\vec{w}_u(Q) \subseteq \vec{w}_u(P)$ . What remains is to show that  $w_u(Q) \geq (1 - 3\varepsilon)w_u(P)$ . If  $u \in \Omega$ , then this holds as

$$\begin{aligned} \max_{q \in Q} \langle u, q \rangle - \min_{q \in Q} \langle u, q \rangle &\geq \langle u, q_u \rangle - \langle -u, q_{-u} \rangle \\ &\geq \max_{p \in P} \langle u, p \rangle - \varepsilon w_u(P) - \min_{p \in P} \langle u, p \rangle - \varepsilon w_u(P) \quad \text{due to Observation 2} \\ &\geq w_u(P) - 2\varepsilon w_u(P) \\ &\geq (1 - 3\varepsilon)w_u(P) \end{aligned}$$

Consider when  $u \notin \Omega$  and let  $\lambda = \frac{\alpha\varepsilon}{4\sqrt{d}}$ . Then there is a  $u' \in \Omega$  such that  $\|u - u'\| \leq \lambda$  due to the construction of  $\Omega$  and  $-u' \in \Omega$ . Then

$$\begin{aligned} w_{u'}(Q) &= \max_{q \in Q} \langle u', q \rangle - \min_{q \in Q} \langle u', q \rangle \\ &\geq \langle u', q_{u'} \rangle - \min_{q \in Q} \langle u', q \rangle \\ &\geq \langle u', \arg\max_{p \in P} \langle u', p \rangle \rangle - \varepsilon w_{u'}(P) - \min_{q \in Q} \langle u', q \rangle \quad (\text{Observation 2}) \\ &= \langle u', \arg\max_{p \in P} \langle u', p \rangle \rangle - \varepsilon w_{u'}(P) - \max_{q \in Q} \langle -u', q \rangle \\ &= \max_{p \in P} \langle u', p \rangle - \varepsilon w_{u'}(P) - \langle -u', q_{-u'} \rangle \\ &= \max_{p \in P} \langle u', p \rangle - \varepsilon w_{u'}(P) - (\langle -u', \arg\max_{p \in P} \langle -u', p \rangle \rangle - \varepsilon w_{-u'}(P)) \\ &= \max_{p \in P} \langle u', p \rangle - \min_{p \in P} \langle u', p \rangle - 2\varepsilon w_{u'}(P) = (1 - 2\varepsilon)w_{u'}(P) \end{aligned}$$

By Lemma A.2

$$\begin{aligned} w_u(Q) &\geq w_{u'}(Q) - 2\lambda\sqrt{d} \\ &\geq (1 - 2\varepsilon)w_{u'}(P) \\ &\geq (1 - 2\varepsilon)(w_u(P) - 2\lambda\sqrt{d}) - 2\lambda\sqrt{d} \\ &= (1 - 2\varepsilon)w_u(P) - 2\lambda\sqrt{d}(1 - 2\varepsilon + 1) \\ &= (1 - 2\varepsilon)w_u(P) - 4\lambda\sqrt{d}(1 - \varepsilon) \end{aligned}$$

Therefore, by Lemma A.1

$$\begin{aligned} \frac{w_u(Q)}{w_u(P)} &\geq 1 - 2\varepsilon - \frac{4\lambda\sqrt{d}(1 - \varepsilon)}{w_u(P)} \\ &\geq 1 - 2\varepsilon - \frac{4\lambda\sqrt{d}(1 - \varepsilon)}{\alpha} \quad \text{due to } \alpha\text{-fatness of } P \\ &\geq 1 - 2\varepsilon - \varepsilon(1 - \varepsilon) \\ &\geq 1 - 3\varepsilon \end{aligned}$$

and we get our desired result i.e.  $w_u(Q) \geq (1 - 3\varepsilon)w_u(P)$

**Proof of Theorem 3.4.** Let  $k(\varepsilon) \geq O\left(\frac{1}{(\frac{\alpha\varepsilon/3}{4\sqrt{d}})^{d-1}}\right)$  and let  $\Omega$  be a fixed set of directions such that  $|\Omega| = k(\varepsilon)$ . Notice that here, since  $k(\varepsilon) \geq O\left(\frac{1}{(\frac{\alpha\varepsilon/3}{4\sqrt{d}})^{d-1}}\right)$ , by Theorem 3.3, the output of Algorithm 1 would be a relaxed- $\varepsilon$ -kernel. Let  $F_{\text{inner}} : \mathcal{X} \rightarrow \mathbb{R}^{k(\varepsilon)}$  such that

$$F_{\text{inner}}(S) = \begin{pmatrix} \max_{x \in S} \langle u_1, x \rangle \\ \vdots \\ \max_{x \in S} \langle u_{k(\varepsilon)}, x \rangle \end{pmatrix}$$

680 Additionally, let  $F_{\text{outer},1} : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^{k(\varepsilon)}$  such

$$F_{\text{outer},1}(x) = \begin{pmatrix} \langle x, u_1 \rangle \\ \langle x, u_2 \rangle \\ \vdots \\ \langle x, u_{k(\varepsilon)} \rangle \end{pmatrix}$$

681 and  $F_{\text{outer},2} : \mathbb{R}^{k(\varepsilon)} \times \mathbb{R}^{k(\varepsilon)} \rightarrow \mathbb{R}^{k(\varepsilon)}$  be defined as  $F_{\text{outer},2}(x, y) = \text{ReLU}(x + y)$ . Given  $x \in \mathbb{R}^d$   
 682 and  $y \in \mathbb{R}^{k(\varepsilon)}$ , let  $F_{\text{outer}}(x, y) = F_{\text{outer},2}(F_{\text{outer},1}(x) + y)$ .

683 First, we note that  $\ell_p$ -norms converge to the  $\ell_\infty$ -norm so there is a  $p$  such that for any  $x \in \mathbb{R}^N$ ,  
 684  $|||x|||_p - |||x|||_\infty| < \frac{\varepsilon\alpha}{2}$ .  $F_{\text{inner}}$  can be approximated to within  $\frac{\varepsilon\alpha}{2}$  via a sum-decomposition by first  
 685 mapping each  $s \in S$  to  $\phi_1(s) = [\langle u_1, s \rangle^p, \dots, \langle u_{k(\varepsilon)}, s \rangle^p]^T \in \mathbb{R}^{k(\varepsilon)}$  and then taking  $\sum (\phi_1(s))^{1/p}$   
 686 where the  $1/p$ -power is taken elementwise. Thus, by the universal approximation property of MLPs,  
 687 there are multilayer perceptrons  $\text{MLP}_1$  and  $\text{MLP}_2$  such that

$$\left\| F_{\text{inner}}(S) - \text{MLP}_1 \left( \sum_{s \in S} \text{MLP}_2(s) \right) \right\| < \varepsilon\alpha.$$

688 This means that  $F_{\text{inner}}(S)_i$  is within  $\varepsilon\alpha$  for  $\max_{s \in S} \langle u_i, s \rangle$  for any  $S \in \mathcal{X}_\alpha$ . Because each set in  $\mathcal{X}_\alpha$   
 689 is  $\alpha$ -fat,  $\forall u \in S^{d-1}$ ,  $F_{\text{inner}}(S)_i$  is within  $\varepsilon w_u(S)$  of the maximum value. In other words,

$$\left| \text{MLP}_1 \left( \sum_{s \in S} \text{MLP}_2(s) \right)_i - F_{\text{inner}}(S)_i \right| \leq \varepsilon \cdot \alpha \leq \varepsilon \cdot w_{u_i}(S).$$

690 We can implement  $F_{\text{outer},1}$  with  $\text{MLP}_3$  i.e.  $F_{\text{outer},1}(x) = \text{MLP}_3(x)$  where  $\text{MLP}_3$  can realize  
 691  $F_{\text{outer},1}$  by choosing its weight matrix to be diagonal with entries  $[u_1, \dots, u_{k(\varepsilon)}]$  along the diagonal.  
 692 Similarly,  $F_{\text{outer},2}$  can be exactly represented by

$$F_{\text{outer},2}(x, y) = \text{MLP}_4(x + y)$$

693 where  $\text{MLP}_4$  represents the identity function followed by a ReLU activation.  $F_{\text{outer}}$  can be rep-  
 694 resented by a neural network  $\psi$  such that  $\psi(x, y) = \text{MLP}_4(\text{MLP}_3(x) + y)$ . Given  $P \in \mathcal{X}_\alpha$  and  
 695  $p_i \in P$ ,

$$\psi \left( p_i + \text{MLP}_1 \left( \sum_{s \in S} \text{MLP}_2(s) \right) \right) = \begin{pmatrix} \text{ReLU}(\langle u_1, p_i \rangle + \varepsilon w_{u_1}(P) - \max_{p \in P} \langle u_1, p \rangle) \\ \vdots \\ \text{ReLU}(\langle u_{k(\varepsilon)}, p_i \rangle + \varepsilon w_{u_{k(\varepsilon)}}(P) - \max_{p \in P} \langle u_{k(\varepsilon)}, p \rangle) \end{pmatrix}$$

696 This is exactly the point-wise update computed in a SumFormer architecture so if we instantiate a  
 697 SumFormer  $\mathcal{S}_\varepsilon$  with the MLPs described above and each MLP in  $\mathcal{S}_\varepsilon$  mapping to an intermediate  
 698 dimension of  $k(\varepsilon)$ ,  $Q_{\mathcal{S}_\varepsilon} = L_{1,\text{col}}(\mathcal{S}_\varepsilon(P))^T P$  is a relaxed- $\varepsilon$ -kernel by Theorem 3.3. Additionally,  
 699 note that each MLP used to approximate  $Q_{\mathcal{S}_\varepsilon}$  maps to  $\mathbb{R}^{k(\varepsilon)}$  where  $k$  is independent of the size of  
 700 the input set and only dependent on the input dimension  $d$ , the fatness of the point set  $\alpha$ , and the  
 701 desired approximation error  $\varepsilon$ .

702 **Proof of Corollary 3.5** From Theorem 3.4, we know that  $Q_\varepsilon = L_{1,\text{col}}(\mathcal{S}_\varepsilon(P))^T P$  is a relaxed  
 703  $\varepsilon$ -kernel for  $P$ . By Theorem 3.2, since  $Q_\varepsilon$  is a relaxed  $\varepsilon$ -kernel, it is  $3c\varepsilon$ -coreset for  $P$ . Then  
 704  $(1 - 3c\varepsilon)\mu(P) \leq \mu(Q_\varepsilon) \leq (1 + 3c\varepsilon)\mu(P)$ . By the universality of DeepSets,  $\mathcal{N}_{\text{deepset}}$  can approximate  
 705  $\mu$  to arbitrary accuracy for any point set in  $\mathcal{X}_\alpha$ . Thus, there is a  $\mathcal{N}_{\text{deepset}}$  architecture such that

$$|\mathcal{N}_{\text{deepset}}(Q_\varepsilon) - \mu(Q_\varepsilon)| < \varepsilon$$

706 and we get the desired result.

#### 707 A.4.2 Proofs from Section 4

708 **Proof of Theorem 4.3** Note that the non-relaxed version is also given in [2]. For completeness, we  
 709 prove this theorem for the relaxed case which essentially just follows from the definitions of duality.

710 First for any point  $x \in \mathbb{R}^{d-1}$ , let  $\hat{x} = [x, 1]$  denote the point in  $\mathbb{R}^d$  with the last coordinate being 1.  
 711 Let  $\phi(x) := \hat{x}/\|\hat{x}\| \in \mathbb{S}^{d-1}$  be the unit vector in direction  $x$ . Let  $\mathbb{S}_+^{d-1}$  be the positive semi-sphere.

712 It is easy to see that  $\phi$  is bijective from  $\mathbb{R}^{d-1}$  to  $\mathbf{S}_+^{d-1}$ . Note that for any  $x \in \mathbb{R}^{d-1}$ ,  $\langle \hat{x}, f^* \rangle$  and for  
 713 any set of functions  $\mathcal{H}$ ,

$$\begin{aligned} L_{\mathcal{H}}(x) &= \min_{h \in \mathcal{H}} h(x) = \|x\| \min_{h \in \mathcal{H}} \langle \phi(x), h^* \rangle \quad \text{and} \\ U_{\mathcal{H}}(x) &= \max_{h \in \mathcal{H}} h(x) = \|x\| \max_{h \in \mathcal{H}} \langle \phi(x), h^* \rangle. \end{aligned}$$

714 Then

$$\mathcal{E}_{\mathcal{H}}(x) = \|x\| \cdot \mathbf{w}_{\phi(x)}(\mathcal{H}^*)$$

715 where  $\mathcal{H}^*$  is the dual point set of  $\mathcal{H}$ .

716 If  $\mathcal{G} = \{g_1, \dots, g_k\}$  is a relaxed  $\varepsilon$ -kernel for  $\mathcal{F}$ . Then for any  $x \in \mathbb{R}^d$ , we have that  $L_{\mathcal{F}}(x) \geq$   
 717  $L_{\mathcal{G}}(x) - \varepsilon \mathcal{E}_{\mathcal{F}}(x)$  and  $U_{\mathcal{F}}(x) \leq U_{\mathcal{G}}(x) + \varepsilon \mathcal{E}_{\mathcal{F}}(x)$ . It then follows that for any  $x \in \mathbb{R}^{d-1}$ ,

$$\begin{aligned} \min_{f^* \in \mathcal{F}^*} \langle \phi(x), f^* \rangle &\geq \min_{g^* \in \mathcal{G}^*} \langle \phi(x), g^* \rangle - \varepsilon \mathbf{w}_{\phi(x)}(\mathcal{F}^*) \quad \text{and} \\ \max_{f^* \in \mathcal{F}^*} \langle \phi(x), f^* \rangle &\leq \max_{g^* \in \mathcal{G}^*} \langle \phi(x), g^* \rangle + \varepsilon \mathbf{w}_{\phi(x)}(\mathcal{F}^*) \end{aligned}$$

718 That is, for any  $u \in \mathbf{S}_+^{d-1}$ ,  $\vec{w}_u(\mathcal{G}^*)$   $\varepsilon$ -approximates  $\vec{w}_u(\mathcal{F}^*)$ . Given that  $\langle -u, p \rangle = -\langle u, p \rangle$  for any  
 719 point  $p$ , this means that  $\vec{w}_u(\mathcal{G}^*)$   $\varepsilon$ -approximates  $\vec{w}_u(\mathcal{F}^*)$  holds for any  $u \in \mathbf{S}_+^{d-1} \cup \mathbf{S}_-^{d-1}$ . Hence  
 720 the only directions left are  $u \in \mathbf{S}^{d-1} \cap \{x \in \mathbb{R}^d, x_d = 0\}$ . However, given that each term above is  
 721 continuous, this holds due to continuity.

722 The other direction follows easily from the fact that  $\phi$  being bijective.

723 **Proof of Theorem 4.4** Suppose  $\mathcal{F}$  admits a linearization of dimension  $m$ . For any  $x \in \mathbb{R}^d$ , we  
 724 will show first that  $\mathcal{E}_{\mathcal{Q}_{\varepsilon}}(x) \subseteq \mathcal{E}_{\mathcal{F}}(x)$ . Then we will show that  $\mathcal{E}_{\mathcal{Q}_{\varepsilon}}(x) \geq (1 - \varepsilon) \mathcal{E}_{\mathcal{F}}(x)$ . If we use  
 725 Algorithm 1 to compute a relaxed- $\varepsilon$ -kernel of the dual  $\mathcal{F}^*$ , we know that each  $q_i^* \in \mathcal{Q}_{\varepsilon}^*$  is

$$q_i^* = \alpha_1^i \Psi(a_1) + \dots + \alpha_N^i \Psi(a_N)$$

726 where  $\sum_{\ell=1}^N \alpha_{\ell}^i = 1$ . Additionally, we will let  $k = |\mathcal{Q}_{\varepsilon}^*|$ . Then we know there is a  $q_i \in \mathcal{Q}_{\varepsilon}$  where

$$q_i(x) = (\alpha_1^i \dots \alpha_N^i) \begin{pmatrix} \psi_0(a_1) \\ \vdots \\ \psi_0(a_N) \end{pmatrix} + (\alpha_1^i \dots \alpha_N^i) \begin{pmatrix} \psi_1(a_1) \\ \vdots \\ \psi_1(a_N) \end{pmatrix} \varphi_1(x) + \dots + (\alpha_1^i \dots \alpha_N^i) \begin{pmatrix} \psi_m(a_1) \\ \vdots \\ \psi_m(a_N) \end{pmatrix} \varphi_m(x)$$

727 To ease notation, for any  $j \in \{0, \dots, m\}$  and  $i \in [k]$ , we write

$$\tilde{\Psi}_j = \begin{pmatrix} \psi_j(a_1) \\ \vdots \\ \psi_j(a_N) \end{pmatrix} \in \mathbb{R}^N \quad \alpha^i = \begin{pmatrix} \alpha_1^i \\ \vdots \\ \alpha_N^i \end{pmatrix} \in \mathbb{R}^N$$

728 so  $q_i(x) = (\alpha^i)^T \tilde{\Psi}_0 + (\alpha^i)^T \tilde{\Psi}_1 \varphi_1(x) + \dots + (\alpha^i)^T \tilde{\Psi}_m \varphi_m(x)$ . We can re-write  $q_i$  as a convex  
 729 combination of elements in  $\mathcal{F}$ :

$$\begin{aligned} q_i(x) &= \alpha_1^i \Psi(a_1)^T \Phi(x) + \dots + \alpha_N^i \Psi(a_N)^T \Phi(x) \\ &= \alpha_1^i f_1(x) + \dots + \alpha_N^i f_N(x) \end{aligned}$$

730 Let  $x \in \mathbb{R}^d$  and let  $f^* = \operatorname{argmin}_{f \in \mathcal{F}} f(x)$ . Then for any  $q_i \in \mathcal{Q}_{\varepsilon}$ ,

$$q_i(x) = \alpha_1^i f_1(x) + \dots + \alpha_N^i f_N(x) \geq \alpha_1^i f^*(x) + \dots + \alpha_N^i f^*(x) = f^*(x)$$

731 Therefore,  $L_{\mathcal{Q}_{\varepsilon}} = \min_{q \in \mathcal{Q}_{\varepsilon}} q(x) \geq L_{\mathcal{F}}(x) = \min_{f \in \mathcal{F}} f(x)$ . A similar argument shows that  
 732  $U_{\mathcal{Q}_{\varepsilon}}(x) \leq U_{\mathcal{F}}(x)$ .

733 Now we will show that  $\mathcal{E}_{\mathcal{Q}_\varepsilon}(x) \geq (1 - \varepsilon)\mathcal{E}_{\mathcal{F}}(x)$  for any  $x \in \mathbb{R}^d$ . Given  $x \in \mathbb{R}^d$ , we know that  
 734  $\Phi(x) = (1, \varphi_1(x), \dots, \varphi_m(x)) \in \mathbb{R}^{m+1}$ . Define  $\mathcal{F}^{\text{Lin}}$  and  $\mathcal{Q}_\varepsilon^{\text{Lin}}$  to be a set of linear functions

$$\mathcal{F}^{\text{Lin}} = \left\{ f_i^{\text{Lin}}(x_1, \dots, x_m) = \Psi(a_j)^T \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_m \end{pmatrix} : a_j \in A \right\}$$

$$\mathcal{Q}_\varepsilon^{\text{Lin}} = \left\{ ((\alpha^i)^T \tilde{\Psi}_0 \dots (\alpha^i)^T \tilde{\Psi}_m) \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_m \end{pmatrix} : i \in [m] \right\}.$$

735 First, notice that  $(\mathcal{F}^{\text{Lin}})^* = \mathcal{F}^*$  and  $(\mathcal{Q}_\varepsilon^{\text{Lin}})^* = \mathcal{Q}_\varepsilon^*$ . We know that  $\mathcal{Q}_\varepsilon^*$  is a relaxed- $\varepsilon$ -kernel of  
 736  $\mathcal{F}^*$  so by Theorem 4.3,  $\mathcal{Q}_\varepsilon^{\text{Lin}}$  is a relaxed- $\varepsilon$ -kernel for  $\mathcal{F}^*$ . Given any  $x \in \mathbb{R}^d$ , we know that  
 737  $f_i(x) = f_i^{\text{Lin}}(\Phi(x))$  so

$$\mathcal{E}_{\mathcal{F}}(x) = \mathcal{E}_{\mathcal{F}^{\text{Lin}}}(\Phi(x)).$$

738 Similarly, we know that  $q_j(x) = q_j^{\text{Lin}}(x)$  so

$$\mathcal{E}_{\mathcal{Q}_\varepsilon}(x) = \mathcal{E}_{\mathcal{Q}_\varepsilon^{\text{Lin}}}(\Phi(x)).$$

739 Since  $\mathcal{Q}_\varepsilon^{\text{Lin}}$  is a relaxed- $\varepsilon$ -kernel for  $\mathcal{F}^*$ ,

$$\mathcal{E}_{\mathcal{Q}_\varepsilon}(x) \geq (1 - \varepsilon)\mathcal{E}_{\mathcal{F}}(x).$$

Table 5: Comparison of  $\mathcal{E}_{\text{dir}}$  between models trained with  $L_{1,\text{col}}$ -normalization vs. softmax normalization. All models were trained on mixed synthetic data in  $\mathbb{R}^d$  where  $d \in \{2, 3, 5\}$ . We report  $\mathcal{E}_{\text{dir}}$  for both in-distribution data (point clouds sampled from uniform balls) as well as out-of-distribution real data (SQUID for  $\mathbb{R}^2$  and ModelNet  $\mathbb{R}^3$ ).

Dimension	Method	Normalization	In-dist.	OOD (Real)
2D	$\mathcal{S}_\varepsilon$	softmax	$0.027 \pm 0.018$	$0.065 \pm 0.071$
	$\mathcal{S}_\varepsilon$	$L_1$	$0.160 \pm 0.124$	$0.492 \pm 0.223$
	$\mathcal{T}_\varepsilon$	softmax	$0.032 \pm 0.024$	$0.296 \pm 0.113$
	$\mathcal{T}_\varepsilon$	$L_1$	$0.531 \pm 0.426$	$1.020 \pm 0.250$
3D	$\mathcal{S}_\varepsilon$	softmax	$0.035 \pm 0.029$	$0.055 \pm 0.480$
	$\mathcal{S}_\varepsilon$	$L_1$	$0.396 \pm 0.222$	$0.612 \pm 0.170$
	$\mathcal{T}_\varepsilon$	softmax	$0.042 \pm 0.038$	$0.117 \pm 0.071$
	$\mathcal{T}_\varepsilon$	$L_1$	$0.538 \pm 1.213$	$0.970 \pm 1.115$
5D	$\mathcal{S}_\varepsilon$	softmax	$0.075 \pm 0.025$	–
	$\mathcal{S}_\varepsilon$	$L_1$	$0.668 \pm 0.86$	–
	$\mathcal{T}_\varepsilon$	softmax	$0.087 \pm 0.030$	–
	$\mathcal{T}_\varepsilon$	$L_1$	$1.02 \pm 0.650$	–

Table 6:  $\mathcal{E}_{\text{dir}}$  on datasets in  $\mathbb{R}^2$  for an output relaxed- $\varepsilon$ -kernel of 16 points. We compare SumFormer and Transformer approaches across three different test sets for each train set. Note that ‘Mixed Synthetic’ has no ‘OOD, Synthetic’ because the train set contains all types of synthetic data. Boldface indicates the lower error per row

Train Set	Method	Test Sets		
		In-dist.	OOD	
			Synthetic	SQUID
Ellipse	$\mathcal{S}_\varepsilon$	$0.060 \pm 0.026$	<b><math>0.044 \pm 0.042</math></b>	<b><math>0.127 \pm 0.111</math></b>
	$\mathcal{T}_\varepsilon$	<b><math>0.029 \pm 0.013</math></b>	$0.098 \pm 0.056$	$0.275 \pm 0.112$
Gaussian Mixture	$\mathcal{S}_\varepsilon$	<b><math>0.014 \pm 0.013</math></b>	<b><math>0.043 \pm 0.040</math></b>	<b><math>0.028 \pm 0.043</math></b>
	$\mathcal{T}_\varepsilon$	$0.027 \pm 0.024$	$0.097 \pm 0.049$	$0.218 \pm 0.101$
SQUID	$\mathcal{S}_\varepsilon$	<b><math>0.028 \pm 0.022</math></b>	$0.504 \pm 0.114$	<b><math>0.028 \pm 0.022</math></b>
	$\mathcal{T}_\varepsilon$	$0.061 \pm 0.052$	<b><math>0.40 \pm 0.109</math></b>	$0.061 \pm 0.052$
Mixed Synthetic	$\mathcal{S}_\varepsilon$	<b><math>0.027 \pm 0.018</math></b>	–	<b><math>0.065 \pm 0.071</math></b>
	$\mathcal{T}_\varepsilon$	<b><math>0.032 \pm 0.024</math></b>	–	$0.296 \pm 0.113$

Table 7:  $\mathcal{E}_{\text{dir}}$  for predicted relaxed- $\varepsilon$ -kernels of 64 points in  $\mathbb{R}^3$ . We compare SumFormer and Transformer approaches across three different test sets for each train set. Note that ‘Mixed Synthetic’ has no ‘OOD, Synthetic’ because the train set contains all types of synthetic data. Boldface indicates the lower error per row.

Train Set	Method	Test Sets		
		In-Dist.	OOD	
			Synthetic	ModelNet
Ellipse	$\mathcal{S}_\varepsilon$	<b><math>0.049 \pm 0.019</math></b>	<b><math>0.032 \pm 0.024</math></b>	<b><math>0.066 \pm 0.053</math></b>
	$\mathcal{T}_\varepsilon$	$0.055 \pm 0.021$	$0.106 \pm 0.076$	$0.147 \pm 0.077$
Gaussian Mixture	$\mathcal{S}_\varepsilon$	<b><math>0.047 \pm 0.027</math></b>	<b><math>0.039 \pm 0.026</math></b>	<b><math>0.064 \pm 0.050</math></b>
	$\mathcal{T}_\varepsilon$	$0.054 \pm 0.027$	$0.066 \pm 0.033$	$0.250 \pm 0.125$
ModelNet	$\mathcal{S}_\varepsilon$	<b><math>0.041 \pm 0.037</math></b>	<b><math>0.099 \pm 0.077</math></b>	<b><math>0.041 \pm 0.037</math></b>
	$\mathcal{T}_\varepsilon$	$0.049 \pm 0.044$	$0.265 \pm 0.118$	$0.049 \pm 0.044$
Mixed Synthetic	$\mathcal{S}_\varepsilon$	<b><math>0.035 \pm 0.029</math></b>	–	<b><math>0.055 \pm 0.048</math></b>
	$\mathcal{T}_\varepsilon$	$0.042 \pm 0.038$	–	$0.117 \pm 0.071$

Table 8:  $\mathcal{E}_{\text{dir}}$  for predicted relaxed- $\varepsilon$ -kernels of 200 points in  $\mathbb{R}^5$ . We compare SumFormer and Transformer approaches for in-distribution and out-of-distribution synthetic test sets. Note that there are no real datasets in  $\mathbb{R}^5$  so the only test sets here are synthetic.

Train Set	Method	In-Distribution	OOD, Synthetic
Uniform Ball	$\mathcal{S}_\varepsilon$	$0.084 \pm 0.016$	$0.091 \pm 0.041$
	$\mathcal{T}_\varepsilon$	$0.118 \pm 0.022$	$0.29 \pm 0.117$
Ellipse	$\mathcal{S}_\varepsilon$	$0.085 \pm 0.025$	$0.077 \pm 0.027$
	$\mathcal{T}_\varepsilon$	$0.084 \pm 0.024$	$0.196 \pm 0.133$
Single Gaussian	$\mathcal{S}_\varepsilon$	$0.065 \pm 0.018$	$0.10 \pm 0.033$
	$\mathcal{T}_\varepsilon$	$0.065 \pm 0.022$	$0.183 \pm 0.053$
Gaussian Mixture	$\mathcal{S}_\varepsilon$	$0.084 \pm 0.028$	$0.087 \pm 0.027$
	$\mathcal{T}_\varepsilon$	$0.091 \pm 0.025$	$0.109 \pm 0.032$
Mixed Synthetic	$\mathcal{S}_\varepsilon$	$0.075 \pm 0.025$	–
	$\mathcal{T}_\varepsilon$	$0.087 \pm 0.032$	–

Table 9: Directional error across 2D datasets for various  $\epsilon$ -kernel sizes. Note that we already report performance for the mixed synthetic data in the tables in the main text so we do not report the performance here. Additionally, note that the ‘Uniform Disk’ data is the same as the ‘Uniform Ball’ dataset described in Table 3.

Train Set	$\epsilon$	Method	Uniform Disk	Ellipse	Single Gaussian	Gaussian Mixture	SQUID
Uniform Ball	16	$\mathcal{S}_\epsilon$	$0.040 \pm 0.013$	$0.109 \pm 0.081$	$0.125 \pm 0.057$	$0.074 \pm 0.059$	$0.101 \pm 0.065$
		$\mathcal{T}_\epsilon$	$0.040 \pm 0.013$	$0.200 \pm 0.094$	$0.370 \pm 0.078$	$0.346 \pm 0.144$	$0.297 \pm 0.110$
	64	$\mathcal{S}_\epsilon$	$0.009 \pm 0.004$	$0.065 \pm 0.051$	$0.214 \pm 0.057$	$0.061 \pm 0.053$	$0.264 \pm 0.093$
		$\mathcal{T}_\epsilon$	$0.007 \pm 0.004$	$0.158 \pm 0.079$	$0.385 \pm 0.058$	$0.287 \pm 0.123$	$0.236 \pm 0.099$
	200	$\mathcal{S}_\epsilon$	$0.006 \pm 0.003$	$0.069 \pm 0.057$	$0.128 \pm 0.062$	$0.097 \pm 0.099$	$0.174 \pm 0.085$
		$\mathcal{T}_\epsilon$	$0.005 \pm 0.003$	$0.199 \pm 0.110$	$0.465 \pm 0.041$	$0.262 \pm 0.110$	$0.427 \pm 0.109$
Ellipse	16	$\mathcal{S}_\epsilon$	$0.045 \pm 0.013$	$0.058 \pm 0.026$	$0.042 \pm 0.039$	$0.067 \pm 0.057$	$0.130 \pm 0.109$
		$\mathcal{T}_\epsilon$	$0.031 \pm 0.010$	$0.029 \pm 0.013$	$0.093 \pm 0.056$	$0.050 \pm 0.041$	$0.272 \pm 0.108$
	64	$\mathcal{S}_\epsilon$	$0.011 \pm 0.005$	$0.015 \pm 0.008$	$0.034 \pm 0.050$	$0.078 \pm 0.086$	$0.088 \pm 0.070$
		$\mathcal{T}_\epsilon$	$0.013 \pm 0.005$	$0.012 \pm 0.006$	$0.116 \pm 0.057$	$0.033 \pm 0.032$	$0.241 \pm 0.087$
	200	$\mathcal{S}_\epsilon$	$0.004 \pm 0.002$	$0.005 \pm 0.003$	$0.033 \pm 0.046$	$0.057 \pm 0.077$	$0.118 \pm 0.073$
		$\mathcal{T}_\epsilon$	$0.010 \pm 0.004$	$0.008 \pm 0.005$	$0.030 \pm 0.027$	$0.034 \pm 0.033$	$0.197 \pm 0.079$
Gaussian	16	$\mathcal{S}_\epsilon$	$0.056 \pm 0.015$	$0.097 \pm 0.052$	$0.032 \pm 0.020$	$0.071 \pm 0.049$	$0.091 \pm 0.069$
		$\mathcal{T}_\epsilon$	$0.062 \pm 0.017$	$0.069 \pm 0.029$	$0.019 \pm 0.014$	$0.080 \pm 0.052$	$0.206 \pm 0.095$
	64	$\mathcal{S}_\epsilon$	$0.029 \pm 0.009$	$0.050 \pm 0.029$	$0.011 \pm 0.010$	$0.039 \pm 0.032$	$0.078 \pm 0.068$
		$\mathcal{T}_\epsilon$	$0.052 \pm 0.015$	$0.056 \pm 0.027$	$0.005 \pm 0.007$	$0.083 \pm 0.058$	$0.241 \pm 0.100$
	200	$\mathcal{S}_\epsilon$	$0.017 \pm 0.007$	$0.057 \pm 0.060$	$0.003 \pm 0.004$	$0.056 \pm 0.048$	$0.067 \pm 0.058$
		$\mathcal{T}_\epsilon$	$0.047 \pm 0.015$	$0.049 \pm 0.020$	$0.004 \pm 0.006$	$0.071 \pm 0.049$	$0.220 \pm 0.099$
Gaussian Mixture	16	$\mathcal{S}_\epsilon$	$0.024 \pm 0.008$	$0.033 \pm 0.015$	$0.071 \pm 0.059$	$0.013 \pm 0.013$	$0.028 \pm 0.037$
		$\mathcal{T}_\epsilon$	$0.040 \pm 0.011$	$0.034 \pm 0.011$	$0.063 \pm 0.050$	$0.027 \pm 0.024$	$0.214 \pm 0.100$
	64	$\mathcal{S}_\epsilon$	$0.015 \pm 0.006$	$0.018 \pm 0.010$	$0.107 \pm 0.067$	$0.006 \pm 0.006$	$0.015 \pm 0.013$
		$\mathcal{T}_\epsilon$	$0.032 \pm 0.011$	$0.017 \pm 0.008$	$0.026 \pm 0.033$	$0.009 \pm 0.010$	$0.156 \pm 0.082$
	200	$\mathcal{S}_\epsilon$	$0.005 \pm 0.003$	$0.009 \pm 0.007$	$0.100 \pm 0.047$	$0.003 \pm 0.004$	$0.012 \pm 0.018$
		$\mathcal{T}_\epsilon$	$0.030 \pm 0.012$	$0.011 \pm 0.006$	$0.032 \pm 0.033$	$0.006 \pm 0.007$	$0.149 \pm 0.084$
SQUID	16	$\mathcal{S}_\epsilon$	$0.548 \pm 0.012$	$0.528 \pm 0.027$	$0.547 \pm 0.040$	$0.395 \pm 0.183$	$0.029 \pm 0.021$
		$\mathcal{T}_\epsilon$	$0.434 \pm 0.041$	$0.403 \pm 0.060$	$0.390 \pm 0.081$	$0.368 \pm 0.163$	$0.066 \pm 0.051$
	64	$\mathcal{S}_\epsilon$	$0.287 \pm 0.014$	$0.289 \pm 0.032$	$0.411 \pm 0.047$	$0.274 \pm 0.173$	$0.009 \pm 0.008$
		$\mathcal{T}_\epsilon$	$0.373 \pm 0.025$	$0.376 \pm 0.039$	$0.409 \pm 0.057$	$0.350 \pm 0.149$	$0.062 \pm 0.042$
	200	$\mathcal{S}_\epsilon$	$0.368 \pm 0.013$	$0.339 \pm 0.034$	$0.378 \pm 0.048$	$0.255 \pm 0.190$	$0.009 \pm 0.009$
		$\mathcal{T}_\epsilon$	$0.322 \pm 0.059$	$0.346 \pm 0.069$	$0.463 \pm 0.081$	$0.379 \pm 0.140$	$0.047 \pm 0.036$
Mixed	16	$\mathcal{S}_\epsilon$	$0.038 \pm 0.012$	$0.052 \pm 0.024$	$0.019 \pm 0.015$	$0.031 \pm 0.026$	$0.064 \pm 0.073$
		$\mathcal{T}_\epsilon$	$0.035 \pm 0.011$	$0.046 \pm 0.023$	$0.017 \pm 0.014$	$0.040 \pm 0.033$	$0.307 \pm 0.112$
	64	$\mathcal{S}_\epsilon$	$0.010 \pm 0.004$	$0.013 \pm 0.008$	$0.002 \pm 0.004$	$0.008 \pm 0.008$	$0.023 \pm 0.030$
		$\mathcal{T}_\epsilon$	$0.016 \pm 0.007$	$0.014 \pm 0.006$	$0.001 \pm 0.003$	$0.014 \pm 0.017$	$0.183 \pm 0.080$
	200	$\mathcal{S}_\epsilon$	$0.006 \pm 0.003$	$0.008 \pm 0.005$	$0.001 \pm 0.002$	$0.005 \pm 0.006$	$0.014 \pm 0.022$
		$\mathcal{T}_\epsilon$	$0.014 \pm 0.008$	$0.012 \pm 0.008$	$0.001 \pm 0.003$	$0.013 \pm 0.014$	$0.144 \pm 0.062$

Table 10:  $\mathcal{E}_{\text{dir}}$  across all 3D datasets for various  $\epsilon$ -kernel sizes. Note that we already report performance for the mixed synthetic data in the tables in the main text so we do not report the performance here.

Dataset	$\epsilon$	Architecture	Uniform Ball	Ellipse	Single Gaussian	Gaussian Mixture	ModelNet
Uniform Ball	16	$\mathcal{S}_\epsilon$	$0.180 \pm 0.030$	$0.239 \pm 0.069$	$0.159 \pm 0.045$	$0.180 \pm 0.072$	$0.161 \pm 0.094$
		$\mathcal{T}_\epsilon$	$0.180 \pm 0.032$	$0.358 \pm 0.085$	$0.553 \pm 0.030$	$0.421 \pm 0.123$	$0.426 \pm 0.130$
	64	$\mathcal{S}_\epsilon$	$0.046 \pm 0.012$	$0.051 \pm 0.018$	$0.029 \pm 0.014$	$0.050 \pm 0.028$	$0.067 \pm 0.057$
		$\mathcal{T}_\epsilon$	$0.041 \pm 0.010$	$0.198 \pm 0.079$	$0.389 \pm 0.047$	$0.257 \pm 0.093$	$0.361 \pm 0.111$
	200	$\mathcal{S}_\epsilon$	$0.021 \pm 0.007$	$0.020 \pm 0.008$	$0.007 \pm 0.007$	$0.029 \pm 0.021$	$0.053 \pm 0.047$
		$\mathcal{T}_\epsilon$	$0.014 \pm 0.005$	$0.158 \pm 0.065$	$0.332 \pm 0.055$	$0.237 \pm 0.097$	$0.361 \pm 0.111$
Ellipse	16	$\mathcal{S}_\epsilon$	$0.178 \pm 0.029$	$0.244 \pm 0.074$	$0.159 \pm 0.045$	$0.187 \pm 0.076$	$0.149 \pm 0.093$
		$\mathcal{T}_\epsilon$	$0.176 \pm 0.027$	$0.244 \pm 0.078$	$0.235 \pm 0.063$	$0.235 \pm 0.111$	$0.232 \pm 0.114$
	64	$\mathcal{S}_\epsilon$	$0.043 \pm 0.010$	$0.049 \pm 0.019$	$0.028 \pm 0.013$	$0.051 \pm 0.029$	$0.065 \pm 0.053$
		$\mathcal{T}_\epsilon$	$0.047 \pm 0.011$	$0.050 \pm 0.018$	$0.196 \pm 0.068$	$0.075 \pm 0.051$	$0.163 \pm 0.089$
	200	$\mathcal{S}_\epsilon$	$0.021 \pm 0.007$	$0.021 \pm 0.009$	$0.008 \pm 0.007$	$0.030 \pm 0.021$	$0.049 \pm 0.045$
		$\mathcal{T}_\epsilon$	$0.019 \pm 0.006$	$0.019 \pm 0.009$	$0.039 \pm 0.050$	$0.087 \pm 0.074$	$0.250 \pm 0.121$
Single Gaussian	16	$\mathcal{S}_\epsilon$	$0.180 \pm 0.030$	$0.252 \pm 0.076$	$0.156 \pm 0.043$	$0.183 \pm 0.073$	$0.200 \pm 0.087$
		$\mathcal{T}_\epsilon$	$0.239 \pm 0.037$	$0.232 \pm 0.051$	$0.146 \pm 0.040$	$0.227 \pm 0.082$	$0.363 \pm 0.121$
	64	$\mathcal{S}_\epsilon$	$0.057 \pm 0.013$	$0.080 \pm 0.034$	$0.033 \pm 0.016$	$0.063 \pm 0.032$	$0.075 \pm 0.052$
		$\mathcal{T}_\epsilon$	$0.078 \pm 0.016$	$0.110 \pm 0.044$	$0.037 \pm 0.015$	$0.123 \pm 0.054$	$0.257 \pm 0.124$
	200	$\mathcal{S}_\epsilon$	$0.034 \pm 0.009$	$0.039 \pm 0.017$	$0.013 \pm 0.010$	$0.040 \pm 0.024$	$0.059 \pm 0.047$
		$\mathcal{T}_\epsilon$	$0.069 \pm 0.016$	$0.075 \pm 0.026$	$0.014 \pm 0.010$	$0.088 \pm 0.044$	$0.219 \pm 0.100$
Gaussian Mixture	16	$\mathcal{S}_\epsilon$	$0.183 \pm 0.031$	$0.231 \pm 0.062$	$0.163 \pm 0.046$	$0.179 \pm 0.072$	$0.154 \pm 0.092$
		$\mathcal{T}_\epsilon$	$0.176 \pm 0.027$	$0.204 \pm 0.047$	$0.176 \pm 0.049$	$0.178 \pm 0.069$	$0.209 \pm 0.099$
	64	$\mathcal{S}_\epsilon$	$0.054 \pm 0.013$	$0.061 \pm 0.020$	$0.032 \pm 0.019$	$0.047 \pm 0.027$	$0.065 \pm 0.053$
		$\mathcal{T}_\epsilon$	$0.060 \pm 0.013$	$0.059 \pm 0.016$	$0.054 \pm 0.027$	$0.054 \pm 0.027$	$0.250 \pm 0.127$
	200	$\mathcal{S}_\epsilon$	$0.027 \pm 0.008$	$0.030 \pm 0.013$	$0.010 \pm 0.009$	$0.023 \pm 0.014$	$0.042 \pm 0.037$
		$\mathcal{T}_\epsilon$	$0.032 \pm 0.010$	$0.029 \pm 0.012$	$0.021 \pm 0.019$	$0.030 \pm 0.019$	$0.232 \pm 0.098$
ModelNet	16	$\mathcal{S}_\epsilon$	$0.209 \pm 0.035$	$0.268 \pm 0.075$	$0.199 \pm 0.058$	$0.206 \pm 0.082$	$0.149 \pm 0.098$
		$\mathcal{T}_\epsilon$	$0.448 \pm 0.029$	$0.408 \pm 0.073$	$0.683 \pm 0.053$	$0.486 \pm 0.126$	$0.150 \pm 0.096$
	64	$\mathcal{S}_\epsilon$	$0.113 \pm 0.022$	$0.148 \pm 0.043$	$0.182 \pm 0.052$	$0.106 \pm 0.084$	$0.041 \pm 0.038$
		$\mathcal{T}_\epsilon$	$0.303 \pm 0.027$	$0.217 \pm 0.051$	$0.332 \pm 0.060$	$0.274 \pm 0.124$	$0.048 \pm 0.040$
	200	$\mathcal{S}_\epsilon$	$0.037 \pm 0.015$	$0.047 \pm 0.029$	$0.021 \pm 0.015$	$0.043 \pm 0.033$	$0.024 \pm 0.030$
		$\mathcal{T}_\epsilon$	$0.169 \pm 0.022$	$0.183 \pm 0.040$	$0.466 \pm 0.046$	$0.286 \pm 0.105$	$0.030 \pm 0.025$
Mixed	16	$\mathcal{S}_\epsilon$	$0.179 \pm 0.030$	$0.238 \pm 0.069$	$0.157 \pm 0.043$	$0.180 \pm 0.075$	$0.150 \pm 0.093$
		$\mathcal{T}_\epsilon$	$0.189 \pm 0.031$	$0.245 \pm 0.071$	$0.162 \pm 0.043$	$0.189 \pm 0.075$	$0.235 \pm 0.106$
	64	$\mathcal{S}_\epsilon$	$0.040 \pm 0.010$	$0.047 \pm 0.019$	$0.020 \pm 0.012$	$0.042 \pm 0.025$	$0.055 \pm 0.049$
		$\mathcal{T}_\epsilon$	$0.044 \pm 0.010$	$0.055 \pm 0.024$	$0.025 \pm 0.014$	$0.066 \pm 0.042$	$0.117 \pm 0.068$
	200	$\mathcal{S}_\epsilon$	$0.017 \pm 0.006$	$0.017 \pm 0.009$	$0.004 \pm 0.005$	$0.022 \pm 0.016$	$0.032 \pm 0.032$
		$\mathcal{T}_\epsilon$	$0.023 \pm 0.007$	$0.023 \pm 0.012$	$0.005 \pm 0.006$	$0.035 \pm 0.029$	$0.112 \pm 0.063$

Table 11:  $\mathcal{E}_{\text{dir}}$  across all 5D datasets for various  $\epsilon$ -kernel sizes. Note that we already report performance for the mixed synthetic data in the tables in the main text so we do not report the performance here.

Train dataset	$\epsilon$	Method	Uniform Ball	Ellipse	Single Gaussian	Gaussian Mixture
Uniform Ball	16	$\mathcal{S}_\epsilon$	$0.431 \pm 0.043$	$0.562 \pm 0.086$	$0.426 \pm 0.067$	$0.524 \pm 0.102$
		$\mathcal{T}_\epsilon$	$0.464 \pm 0.048$	$0.514 \pm 0.069$	$0.676 \pm 0.038$	$0.599 \pm 0.092$
	64	$\mathcal{S}_\epsilon$	$0.188 \pm 0.025$	$0.247 \pm 0.057$	$0.159 \pm 0.033$	$0.232 \pm 0.058$
		$\mathcal{T}_\epsilon$	$0.197 \pm 0.027$	$0.366 \pm 0.066$	$0.538 \pm 0.033$	$0.362 \pm 0.092$
	200	$\mathcal{S}_\epsilon$	$0.084 \pm 0.016$	$0.112 \pm 0.038$	$0.058 \pm 0.023$	$0.103 \pm 0.038$
		$\mathcal{T}_\epsilon$	$0.118 \pm 0.023$	$0.202 \pm 0.052$	$0.369 \pm 0.047$	$0.356 \pm 0.113$
Ellipse	16	$\mathcal{S}_\epsilon$	$0.547 \pm 0.066$	$0.549 \pm 0.072$	$0.554 \pm 0.089$	$0.609 \pm 0.107$
		$\mathcal{T}_\epsilon$	$0.534 \pm 0.066$	$0.534 \pm 0.072$	$0.577 \pm 0.054$	$0.553 \pm 0.102$
	64	$\mathcal{S}_\epsilon$	$0.193 \pm 0.024$	$0.213 \pm 0.043$	$0.168 \pm 0.033$	$0.187 \pm 0.048$
		$\mathcal{T}_\epsilon$	$0.313 \pm 0.051$	$0.302 \pm 0.063$	$0.543 \pm 0.034$	$0.417 \pm 0.095$
	200	$\mathcal{S}_\epsilon$	$0.091 \pm 0.017$	$0.085 \pm 0.024$	$0.069 \pm 0.017$	$0.094 \pm 0.031$
		$\mathcal{T}_\epsilon$	$0.092 \pm 0.018$	$0.084 \pm 0.024$	$0.369 \pm 0.042$	$0.239 \pm 0.091$
Gaussian	16	$\mathcal{S}_\epsilon$	$0.488 \pm 0.066$	$0.529 \pm 0.072$	$0.448 \pm 0.075$	$0.565 \pm 0.142$
		$\mathcal{T}_\epsilon$	$0.728 \pm 0.061$	$0.722 \pm 0.068$	$0.467 \pm 0.073$	$0.631 \pm 0.095$
	64	$\mathcal{S}_\epsilon$	$0.204 \pm 0.027$	$0.250 \pm 0.050$	$0.163 \pm 0.033$	$0.196 \pm 0.049$
		$\mathcal{T}_\epsilon$	$0.471 \pm 0.055$	$0.452 \pm 0.064$	$0.205 \pm 0.045$	$0.411 \pm 0.099$
	200	$\mathcal{S}_\epsilon$	$0.095 \pm 0.017$	$0.116 \pm 0.038$	$0.065 \pm 0.019$	$0.102 \pm 0.032$
		$\mathcal{T}_\epsilon$	$0.177 \pm 0.026$	$0.170 \pm 0.041$	$0.065 \pm 0.022$	$0.165 \pm 0.056$
Gaussian Mixture	16	$\mathcal{S}_\epsilon$	$0.546 \pm 0.066$	$0.539 \pm 0.074$	$0.460 \pm 0.079$	$0.501 \pm 0.098$
		$\mathcal{T}_\epsilon$	$0.622 \pm 0.060$	$0.571 \pm 0.071$	$0.516 \pm 0.081$	$0.504 \pm 0.098$
	64	$\mathcal{S}_\epsilon$	$0.193 \pm 0.025$	$0.213 \pm 0.041$	$0.164 \pm 0.032$	$0.170 \pm 0.045$
		$\mathcal{T}_\epsilon$	$0.197 \pm 0.025$	$0.205 \pm 0.039$	$0.182 \pm 0.036$	$0.174 \pm 0.046$
	200	$\mathcal{S}_\epsilon$	$0.093 \pm 0.017$	$0.103 \pm 0.030$	$0.073 \pm 0.019$	$0.084 \pm 0.028$
		$\mathcal{T}_\epsilon$	$0.097 \pm 0.018$	$0.093 \pm 0.025$	$0.084 \pm 0.025$	$0.091 \pm 0.032$
Mixed	16	$\mathcal{S}_\epsilon$	$0.562 \pm 0.065$	$0.544 \pm 0.073$	$0.485 \pm 0.085$	$0.512 \pm 0.101$
		$\mathcal{T}_\epsilon$	$0.602 \pm 0.063$	$0.559 \pm 0.071$	$0.477 \pm 0.078$	$0.524 \pm 0.095$
	64	$\mathcal{S}_\epsilon$	$0.194 \pm 0.026$	$0.220 \pm 0.045$	$0.162 \pm 0.032$	$0.177 \pm 0.047$
		$\mathcal{T}_\epsilon$	$0.202 \pm 0.025$	$0.228 \pm 0.046$	$0.168 \pm 0.034$	$0.188 \pm 0.051$
	200	$\mathcal{S}_\epsilon$	$0.085 \pm 0.017$	$0.076 \pm 0.022$	$0.060 \pm 0.020$	$0.079 \pm 0.031$
		$\mathcal{T}_\epsilon$	$0.094 \pm 0.018$	$0.094 \pm 0.028$	$0.064 \pm 0.020$	$0.096 \pm 0.047$

Table 12: Results for minimum enclosing ball (2D). We report the relative error for the predicted radius (w.r.t to ground truth radius) ( $\mathcal{E}_r$ ) and percentage of points excluded from the predicted enclosing ball ( $\mathcal{E}_p$ ). All processors are configured to output a coarsened set of 16 points and frozen processors are trained on mixed synthetic data.

Train Set	Architecture	$\mathcal{E}_r$	$\mathcal{E}_p(\%)$
Uniform Disk	$\mathcal{S}_{\text{extent}}$ (Frozen)	<b><math>0.020 \pm 0.010</math></b>	$0.2 \pm 0.5$
	$\mathcal{S}_{\text{extent}}$ (E2E)	$0.023 \pm 0.019$	$2.3 \pm 1.9$
	$\mathcal{T}_{\text{extent}}$ (Frozen)	$0.099 \pm 0.050$	<b><math>0.1 \pm 0.3</math></b>
	$\mathcal{T}_{\text{extent}}$ (E2E)	$0.024 \pm 0.020$	$1.4 \pm 1.8$
	$\mathcal{S}_{\text{Baseline}}$ (E2E)	$0.048 \pm 0.028$	$0.5 \pm 1.0$
	$\mathcal{T}_{\text{Baseline}}$ (E2E)	$0.030 \pm 0.024$	$0.3 \pm 3.0$
SQUID	$\mathcal{S}_{\text{extent}}$ (Frozen)	$0.020 \pm 0.200$	$2.0 \pm 3.0$
	$\mathcal{S}_{\text{extent}}$ (E2E)	<b><math>0.010 \pm 0.010</math></b>	$2.0 \pm 3.0$
	$\mathcal{T}_{\text{extent}}$ (Frozen)	$0.068 \pm 0.045$	<b><math>1.6 \pm 3.2</math></b>
	$\mathcal{T}_{\text{extent}}$ (E2E)	$0.190 \pm 0.090$	$4.0 \pm 5.0$
	$\mathcal{S}_{\text{Baseline}}$ (E2E)	$0.027 \pm 0.025$	$2.8 \pm 2.5$
	$\mathcal{T}_{\text{Baseline}}$ (E2E)	$0.039 \pm 0.034$	$4.5 \pm 3.8$

Table 13: Results for minimum enclosing ball (3D). We report the relative error for the predicted radius ( $\mathcal{E}_r$ ) and the percentage of points which excluded from the estimated ball ( $\mathcal{E}_p(\%)$ ). Hidden dimension of the SumFormer and Transformer blocks is five. All frozen processors are trained on mixed synthetic data.

Train Set	Architecture	$\mathcal{E}_r$	$\mathcal{E}_p(\%)$
Uniform Ball	$\mathcal{S}_{\text{extent}}$ (Frozen)	$0.030 \pm 0.020$	<b><math>0.7 \pm 1.0</math></b>
	$\mathcal{S}_{\text{extent}}$ (E2E)	$0.035 \pm 0.028$	$2.8 \pm 3.4$
	$\mathcal{T}_{\text{extent}}$ (Frozen)	<b><math>0.020 \pm 0.018</math></b>	$4.6 \pm 3.8$
	$\mathcal{T}_{\text{extent}}$ (E2E)	$0.076 \pm 0.051$	$2.0 \pm 4.0$
	$\mathcal{S}_{\text{Baseline}}$ (E2E)	$0.056 \pm 0.031$	$1.8 \pm 2.5$
	$\mathcal{T}_{\text{Baseline}}$ (E2E)	$0.035 \pm 0.032$	$8.6 \pm 6.5$
ModelNet	$\mathcal{S}_{\text{extent}}$ (Frozen)	$0.100 \pm 0.070$	<b><math>0.2 \pm 0.7</math></b>
	$\mathcal{S}_{\text{extent}}$ (E2E)	<b><math>0.060 \pm 0.050</math></b>	<b><math>0.2 \pm 0.9</math></b>
	$\mathcal{T}_{\text{extent}}$ (Frozen)	$0.077 \pm 0.072$	$0.6 \pm 1.3$
	$\mathcal{T}_{\text{extent}}$ (E2E)	<b><math>0.029 \pm 0.032</math></b>	$3.3 \pm 4.4$
	$\mathcal{S}_{\text{Baseline}}$ (E2E)	$0.053 \pm 0.058$	$1.6 \pm 2.5$
	$\mathcal{T}_{\text{Baseline}}$ (E2E)	$0.858 \pm 0.289$	$57.3 \pm 39.6$

Table 14: Minimum enclosing ellipse error for point clouds in  $\mathbb{R}^2$  on in-distribution test data, comparing frozen and end-to-end (E2E) training procedures. All frozen processors are trained on mixed synthetic data. We report the relative error in major ( $\mathcal{E}_{r,\text{maj}}$ ) and minor radii ( $\mathcal{E}_{r,\text{min}}$ ) and the percentage of points excluded ( $\mathcal{E}_p(\%)$ ).

Train Set	Architecture	$\mathcal{E}_{r,\text{min}}$	$\mathcal{E}_{r,\text{maj}}$	$\mathcal{E}_p(\%)$
Synthetic Ellipses	$\mathcal{S}_{\text{extent}}$ (Frozen)	<b><math>0.037 \pm 0.037</math></b>	<b><math>0.022 \pm 0.018</math></b>	<b><math>5.1 \pm 5.2</math></b>
	$\mathcal{S}_{\text{extent}}$ (E2E)	$0.056 \pm 0.047$	$0.047 \pm 0.035$	$9.1 \pm 5.4$
	$\mathcal{T}_{\text{extent}}$ (Frozen)	$0.041 \pm 0.040$	$0.027 \pm 0.020$	$7.5 \pm 5.4$
	$\mathcal{T}_{\text{extent}}$ (E2E)	$0.038 \pm 0.033$	$0.022 \pm 0.019$	$6.5 \pm 5.3$
	$\mathcal{S}_{\text{Baseline}}$ (E2E)	$0.378 \pm 0.234$	$0.426 \pm 0.387$	$27.7 \pm 25.1$
	$\mathcal{T}_{\text{Baseline}}$ (E2E)	$0.071 \pm 0.660$	$0.047 \pm 0.038$	$8.8 \pm 7.2$
SQUID	$\mathcal{S}_{\text{extent}}$ (Frozen)	<b><math>0.056 \pm 0.047</math></b>	$0.047 \pm 0.035$	<b><math>6.6 \pm 5.4</math></b>
	$\mathcal{S}_{\text{extent}}$ (E2E)	$0.078 \pm 0.065$	$0.050 \pm 0.038$	$13.9 \pm 7.2$
	$\mathcal{T}_{\text{extent}}$ (Frozen)	$0.058 \pm 0.051$	<b><math>0.032 \pm 0.026</math></b>	$12.5 \pm 6.6$
	$\mathcal{T}_{\text{extent}}$ (E2E)	$0.093 \pm 0.074$	$0.045 \pm 0.034$	$14.4 \pm 7.7$
	$\mathcal{S}_{\text{Baseline}}$ (E2E)	$0.322 \pm 0.365$	$0.250 \pm 0.262$	$32.0 \pm 16.6$
	$\mathcal{T}_{\text{Baseline}}$ (E2E)	$0.357 \pm 0.550$	$0.049 \pm 0.041$	$16.8 \pm 18.4$

Table 15: Minimum enclosing annulus error on in-distribution test data, comparing frozen and end-to-end (E2E) training procedures. All frozen processors are trained on mixed synthetic data. We report the relative error in the width of the annuli and the proportion of points excluded.

Train Set	Architecture	$\mathcal{E}_w$	$\mathcal{E}_p(\%)$
Synthetic Annuli	$\mathcal{S}_{\text{extent}}$ (Frozen)	$0.028 \pm 0.042$	$2.9 \pm 3.5$
	$\mathcal{S}_{\text{extent}}$ (E2E)	<b><math>0.022 \pm 0.028</math></b>	$4.9 \pm 3.0$
	$\mathcal{T}_{\text{extent}}$ (Frozen)	$0.454 \pm 0.577$	$24.2 \pm 21.2$
	$\mathcal{T}_{\text{extent}}$ (E2E)	$0.026 \pm 0.032$	<b><math>1.6 \pm 2.2</math></b>
	$\mathcal{S}_{\text{Baseline}}$ (E2E)	$1.010 \pm 1.650$	$36.7 \pm 28.7$
	$\mathcal{T}_{\text{Baseline}}$ (E2E)	$0.066 \pm 0.087$	$5.7 \pm 5.0$
SQUID	$\mathcal{S}_{\text{extent}}$ (Frozen)	<b><math>0.076 \pm 0.100</math></b>	$8.9 \pm 8.1$
	$\mathcal{S}_{\text{extent}}$ (E2E)	$0.145 \pm 0.160$	$10.0 \pm 8.8$
	$\mathcal{T}_{\text{extent}}$ (Frozen)	$0.096 \pm 0.113$	$11.2 \pm 9.5$
	$\mathcal{T}_{\text{extent}}$ (E2E)	$0.099 \pm 0.112$	<b><math>7.1 \pm 5.8</math></b>
	$\mathcal{S}_{\text{Baseline}}$ (E2E)	$0.084 \pm 0.118$	$9.6 \pm 7.0$
	$\mathcal{T}_{\text{Baseline}}$ (E2E)	$0.152 \pm 0.168$	$12.3 \pm 7.6$

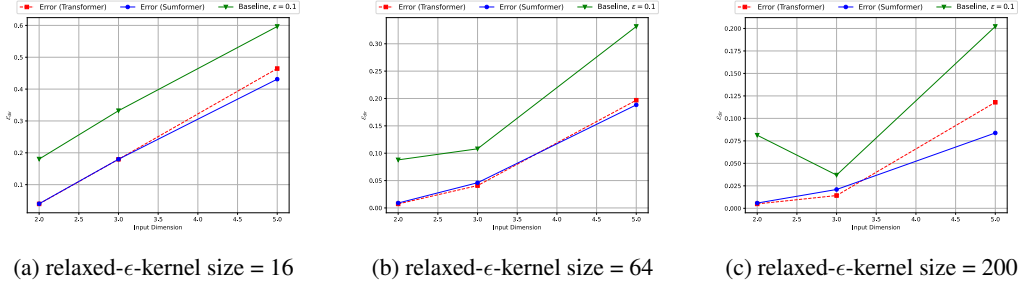


Figure 5: Directional width error ( $\mathcal{E}_{\text{dir}}$ ) vs. the input dimension of the point cloud. We train each model on ‘Uniform Ball’ datasets in  $\mathbb{R}^2$ ,  $\mathbb{R}^3$  and  $\mathbb{R}^5$  and test on in-distribution dataset. We notice that all models, as well as the baseline algorithm have increasing directional width error as the input point cloud dimension increases.

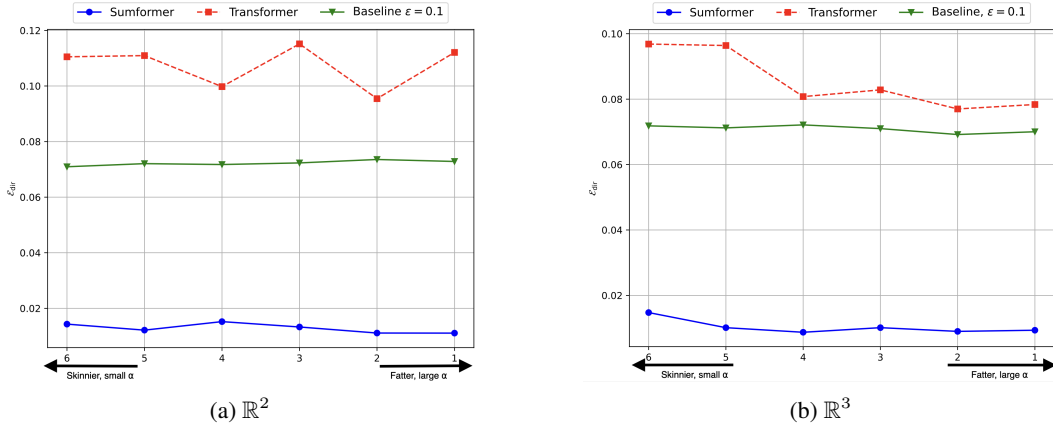


Figure 6: Directional width error ( $\mathcal{E}_{\text{dir}}$ ) vs  $\alpha$ -fatness on relaxed- $\epsilon$ -kernel approximation task for  $\mathbb{R}^2$  and  $\mathbb{R}^3$ . We vary the  $\alpha$ -fatness of the input data by varying the scale of the minor axes of ellipsoids in  $\mathbb{R}^2$  and  $\mathbb{R}^3$  and then sampling point clouds from such ellipsoids. Each input point cloud has 100 points and the output  $\epsilon$ -kernels are fixed at 64.

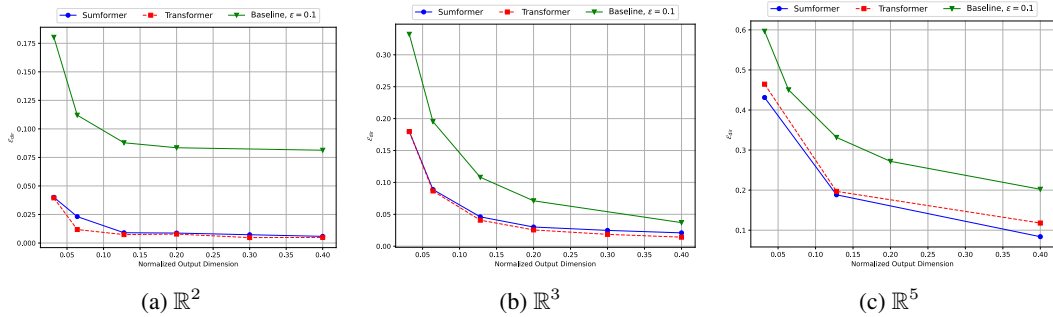


Figure 7: Direction width error  $\mathcal{E}_{\text{dir}}$  vs output size of relaxed- $\epsilon$ -kernel size across dimensions. All models are trained and evaluated on point clouds sampled from uniform balls/disks with 500 points per set. Notice that, similar to the baseline algorithm, each model will have better results as we increase output point set size (with SumFormer implementation of  $\mathcal{N}_{\phi_\epsilon}$  still outperforming Transformer).

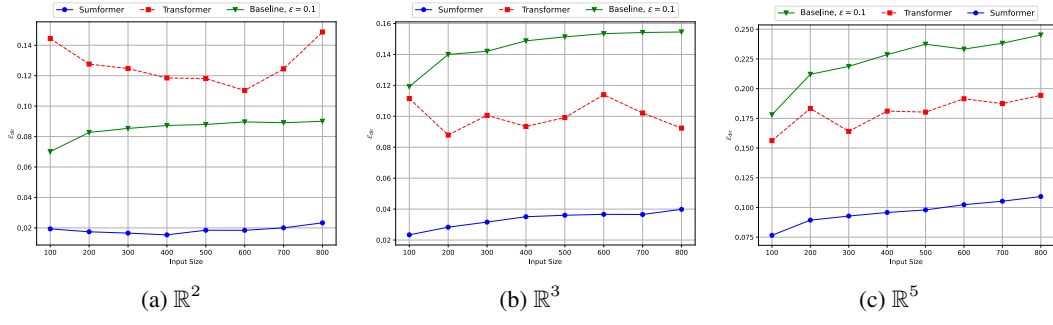


Figure 8: Directional width error  $\mathcal{E}_{\text{dir}}$  vs. input point set size across different dimensions. All models are trained on the uniform ball/disk dataset and evaluated on uniform ellipses/ellipsoids of varying sizes.

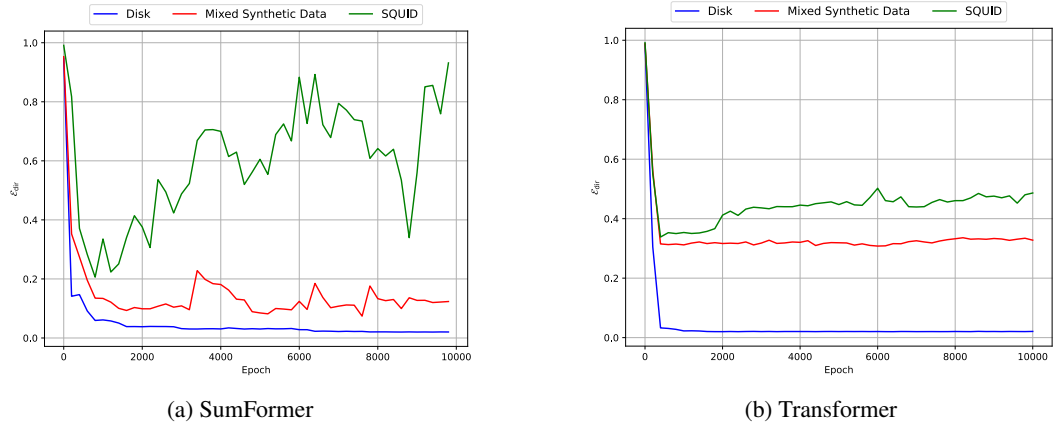


Figure 9: Test error( $\mathcal{E}_{\text{dir}}$ ) on uniform disk dataset across training epochs for  $\mathcal{N}_{\phi_\epsilon}$  where the input point clouds are in  $\mathbb{R}^2$  and the size of relaxed- $\epsilon$ -kernel is 16. All models were trained on the uniform disk dataset.

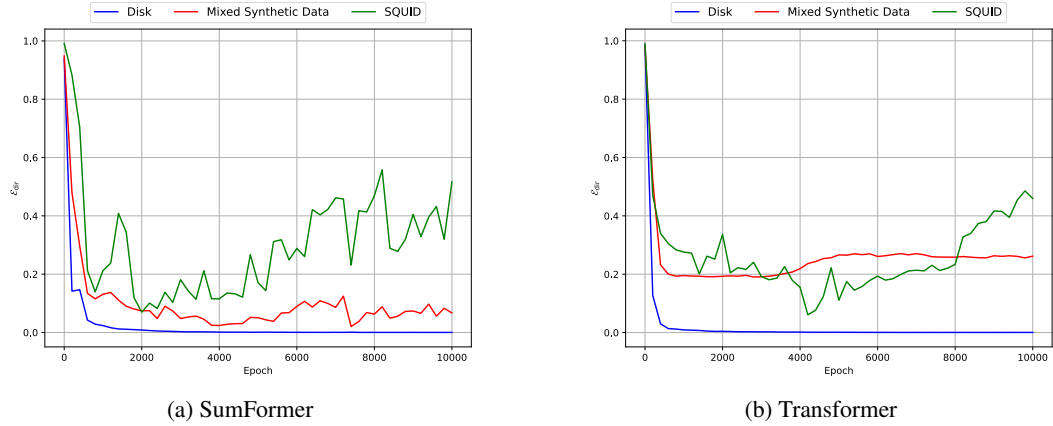
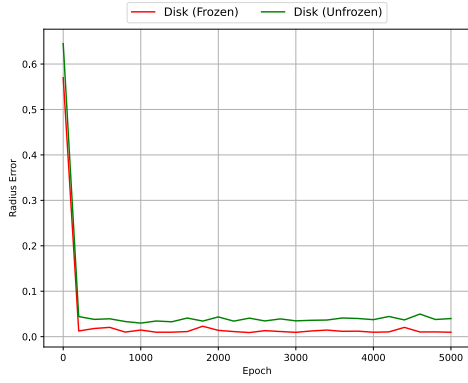
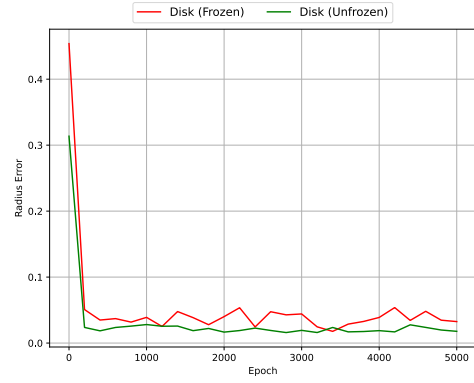


Figure 10: Test error( $\mathcal{E}_{\text{dir}}$ ) on uniform disk dataset across training epochs for  $\mathcal{N}_{\phi_\epsilon}$  where the input point clouds are in  $\mathbb{R}^2$  and the size of relaxed- $\epsilon$ -kernel is 64. All models were trained on the uniform disk dataset.

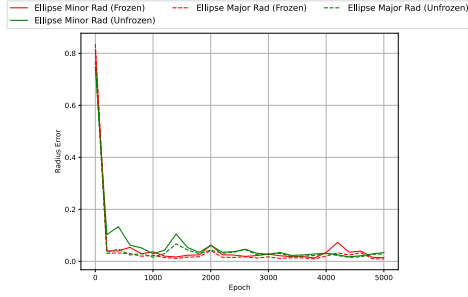


(a) SumFormer

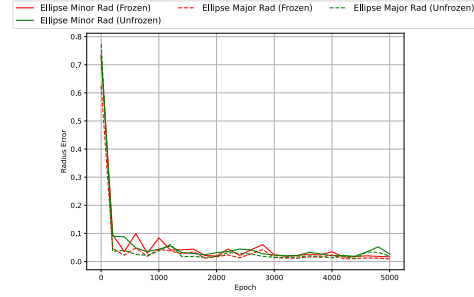


(b) Transformer

Figure 11: Relative error in predicted radius ( $\mathcal{E}_r$ ) on uniform ball/disk test datasets per training epoch for minimum enclosing ball in  $\mathbb{R}^2$ . Models were trained on uniform disks.

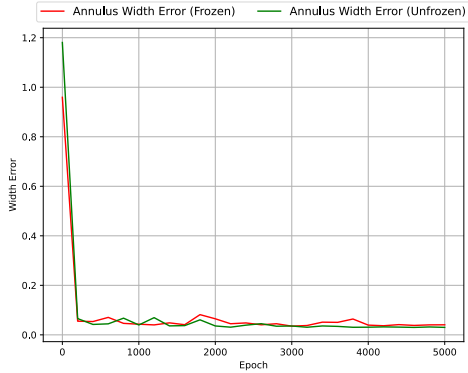


(a) SumFormer

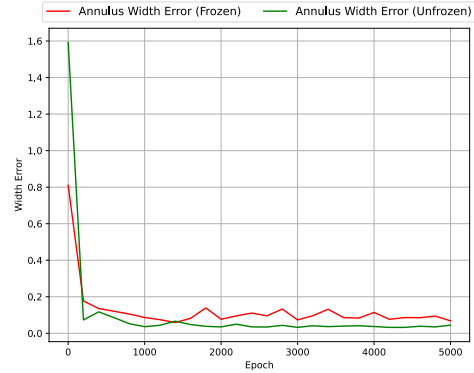


(b) Transformer

Figure 12: Relative error in predicted major and minor radius ( $\mathcal{E}_{r,maj}$  and  $\mathcal{E}_{r,min}$ ) on uniform ellipse test datasets per training epoch for minimum enclosing ellipse in  $\mathbb{R}^2$ . Models were trained on uniform ellipses.



(a) SumFormer



(b) Transformer

Figure 13: Relative error in predicted width ( $\mathcal{E}_w$ ) on uniform annulus test datasets per training epoch for minimum enclosing annulus in  $\mathbb{R}^2$ . Models were trained on uniform annuli.