A Appendix

A.1 Additional results

In this section, we include more detailed results of experiments in Section 5, evaluating the intermediate-p robustness of models trained on different objectives, across different datasets and perturbation distributions. Specifically, the intermediate-q robustness metrics we consider, $\hat{Z}_{\text{MC}}$ and $\hat{Z}_{\text{PS}}$, correspond to estimations of the functional q-norm of the cross entropy loss function evaluated over the perturbation distribution using the Monte Carlo estimator and the path sampling estimator respectively. We also include the standard cross entropy loss and adversarial loss metrics for comparison purposes. The following results were obtained from multiple training runs using 3 different random seeds.

MNIST ℓ_∞-norm ball In Table 4, we report the mean and standard deviation of all intermediate-q robustness estimates over the ℓ_∞-norm ball with radius $\epsilon = 0.3$ on MNIST for models trained according to the cross entropy loss ("Standard"), the Monte Carlo estimate ("MC"), the path sampling estimate computed with Hamiltonian Monte Carlo ("PS"), and PGD adversarial loss. For training, we used $m = 50$ to compute the MC estimate, and $m = 25$ and $L = 2$ to compute the path sampling estimate. The adversarially trained model was trained using PGD with 50 iterations. The evaluation estimate $\hat{Z}_{\text{MC}}$ was computed with $m = 2000$ samples and $\hat{Z}_{\text{PS+HMC}}$ was computed with $m = 100$ samples and $L = 20$ leapfrog steps. The evaluation adversarial loss was computed with PGD with 100 iterations. While both estimators interpolate between loss over random samples (i.e. $q = 1$), and adversarial loss (i.e. $q = \infty$), the path sampling estimator with HMC consistently results in higher (better) estimates of the desired integral for the same number of iterations. These results also show promise for training according to the path sampling metric, as the models trained with the PS+HMC estimate of the objective for larger values of q get increasingly better performance on intermediate-q robustness metrics (i.e. lower $\hat{Z}_{\text{PS+HMC}}$) as compared to models trained with the MC estimate. However, the adversarially trained model still does better in terms of intermediate-q performance, likely due to the nature of MNIST. We also include test robust accuracy of each trained model (from the PGD-100 evaluation) in Table 5, which shows that training using these estimates with increasingly large q does indeed improve worst-case robust performance. We note that we didn't include standard accuracy here, because on MNIST, an adversarially trained model doesn't lose much in terms of standard performance, however on more challenging datasets, we would expect to see a similar decrease in standard accuracy as we increase q, and thus one could choose a value of q based on the desired trade-off between standard and robust accuracy.

CIFAR-10 ℓ_∞-norm ball In Table 6, we report the mean and standard deviation of all intermediate-q robustness estimates over the ℓ_∞-norm ball with radius $\epsilon = 0.03$ on CIFAR-10 in Table 4 for models trained according to the cross entropy loss ("Standard"), the Monte Carlo estimate ("MC"), and PGD adversarial loss. For training, we used $m = 10$ to compute the MC estimate, and the adversarially trained model was trained using PGD with 10 iterations. The evaluation estimate $\hat{Z}_{\text{MC}}$ was computed with $m = 500$ samples and $\hat{Z}_{\text{PS+HMC}}$ was computed with $m = 50$ samples and $L = 10$ leapfrog steps. The evaluation adversarial loss was computed with PGD with 50 iterations and 10 restarts. The results show that, given enough iterations, the intermediate-q robustness estimates do interpolate between loss over random samples and loss over worst-case samples. However, given the more challenging nature of CIFAR-10 vs. MNIST, more samples are needed to get good estimates of the desired integral, making the problem more computationally intensive. While for models trained with the MC estimator, we do see an improvement in intermediate-q robust performance from training with $q = 10$ vs. $q = 1$, training using values of q larger than 10 does not provide much, if any, additional benefit. However, given the same number of iterations, we were not able to improve upon these results by training using the path sampling estimator, suggesting that the number of samples is just not large enough to get a good estimate of the objective for either estimator.

CIFAR-10 spatial transformations In Table 7, we report the mean and standard deviation of all intermediate-q robustness estimates over nondifferentiable spatial transformations on CIFAR-10 for models trained according to the cross entropy loss ("Standard") and the Monte Carlo estimate ("MC"). For training, we used $m = 10$ to compute the MC estimate. The evaluation estimates $\hat{Z}_{\text{MC}}$ and $\hat{Z}_{\text{PS}}$ were computed with $m = 500$ samples, where in this case the path sampling estimate $\hat{Z}_{\text{PS}}$ is based on

Table 4: Mean and standard deviation over 3 training runs with different random seeds for experiments on MNIST for the ℓ_∞-norm ball with radius 0.3. The results show that the intermediate-q robustness metrics, $\hat{Z}_{\text{MC}}$ and $\hat{Z}_{\text{PS+HMC}}$, interpolate between loss over random samples (when $q = 1$) and adversarial loss (when $q = \infty$), with the path sampling estimator consistently resulting in higher (better) estimates for larger values of q. Models trained using the PS+HMC estimate with larger q have better intermediate-q performance (lower $\hat{Z}_{\text{PS+HMC}}$) for corresponding values of q than those trained using the MC estimate.

Train method	Standard	$\hat{Z}_{\text{MC}}$				$\hat{Z}_{\text{PS+HMC}}$				Adv. loss
		$q=1$	$q=10$	$q=10^2$	$q=10^3$	$q=1$	$q=10$	$q=10^2$	$q=10^3$	
Standard	0.028	0.043	0.140	0.251	0.268	0.043	0.160	1.420	4.456	11.649
	±0.001	±0.004	±0.023	±0.041	±0.045	±0.004	±0.028	±0.202	±0.495	±0.893
MC $q=1$	0.034	0.032	0.084	0.143	0.154	0.032	0.088	0.692	2.133	7.363
	±0.001	±0.000	±0.001	±0.002	±0.003	±0.000	±0.002	±0.026	±0.087	±0.435
MC $q=10$	0.027	0.026	0.058	0.098	0.105	0.026	0.058	0.412	1.336	3.722
	±0.001	±0.002	±0.002	±0.002	±0.002	±0.002	±0.001	±0.011	±0.050	±0.231
MC $q=10^2$	0.026	0.025	0.055	0.093	0.099	0.025	0.055	0.388	1.261	3.492
	±0.001	±0.001	±0.001	±0.001	±0.001	±0.001	±0.001	±0.015	±0.069	±0.291
MC $q=10^3$	0.026	0.025	0.055	0.093	0.100	0.025	0.055	0.390	1.268	3.488
	±0.002	±0.002	±0.001	±0.002	±0.001	±0.002	±0.001	±0.013	±0.059	±0.242
PS $q=1$	0.037	0.035	0.094	0.163	0.174	0.035	0.101	0.781	2.334	8.881
	±0.001	±0.001	±0.003	±0.005	±0.007	±0.001	±0.005	±0.027	±0.064	±0.616
PS $q=10$	0.034	0.031	0.075	0.126	0.135	0.031	0.075	0.467	1.307	5.012
	±0.001	±0.001	±0.001	±0.002	±0.003	±0.001	±0.001	±0.014	±0.029	±0.353
PS $q=10^2$	0.030	0.028	0.060	0.099	0.107	0.028	0.058	0.304	0.816	2.613
	±0.003	±0.002	±0.006	±0.010	±0.011	±0.002	±0.006	±0.007	±0.027	±0.193
PS $q=10^3$	0.024	0.024	0.047	0.077	0.083	0.024	0.045	0.239	0.684	1.646
	±0.001	±0.001	±0.001	±0.001	±0.001	±0.001	±0.001	±0.008	±0.037	±0.095
PGD-50	0.032	0.039	0.051	0.076	0.081	0.039	0.048	0.101	0.187	0.270
	±0.005	±0.006	±0.007	±0.010	±0.011	±0.006	±0.007	±0.016	±0.040	±0.033

Table 5: Robust accuracy (PGD-100) of experiments on MNIST for perturbations in the ℓ_∞ ball of radius $\epsilon = 0.3$.

Training method	Robust accuracy
Standard	4.69%
MC $q=1$	25.21%
MC $q=10$	43.54%
MC $q=10^2$	47.27%
MC $q=10^3$	45.45%
PS $q=1$	22.27%
PS $q=10$	43.57%
PS $q=10^2$	65.11%
PS $q=10^3$	69.45%
PGD-50	91.55%

using Gaussian random walk Metropolis Hastings to sample from the unnormalized loss distribution rather than Hamiltonian Monte Carlo due to the non-differentable perturbation distribution. The evaluation adversarial loss was approximated by averaging the maximum loss value encountered for each example during the Metropolis Hastings sampling process. As with the case of the ℓ_∞-norm ball perturbation distribution on CIFAR-10, a larger number of samples are needed to get good estimates of the desired integral, making training using these estimates challenging, as shown by the lack of improvement in robust performance for larger values of q for models trained according to MC $q = 10^2$ vs. MC $q = 10$.

A.2 Additional figures

We include additional plots showing the convergence of Monte Carlo and path sampling estimates on a single test batch given an increasing number of samples. In Figure 2, we plot intermediate-q robustness estimates for $q = 10$ and $q = 100$ on a standard trained model over the ℓ_∞-norm ball

Table 6: Mean and standard deviation over 3 training runs with different random seeds for experiments on CIFAR-10 for the ℓ_∞-norm ball with radius 0.03. The results show that the intermediate-q robustness metrics, $\hat{Z}_{\text{MC}}$ and $\hat{Z}_{\text{PS+HMC}}$, interpolate between loss over random samples (when $q = 1$) and adversarial loss (when $q = \infty$), with the path sampling estimator consistently resulting in higher (better) estimates for larger values of q. How to train for better intermediate-q robust performance on CIFAR-10, while still being computationally reasonable, remains an open question.

Train method	Standard	$\hat{Z}_{\text{MC}}$				$\hat{Z}_{\text{PS+HMC}}$				Adv. loss
		$q=1$	$q=10$	$q=10^2$	$q=10^3$	$q=1$	$q=10$	$q=10^2$	$q=10^3$	
Standard	0.382	0.453	0.787	1.153	1.216	0.453	0.841	2.718	4.991	18.142
	±0.002	±0.006	±0.023	±0.037	±0.039	±0.006	±0.026	±0.090	±0.117	±0.448
MC $q=1$	0.400	0.405	0.532	0.717	0.756	0.405	0.546	1.490	3.140	14.240
	±0.005	±0.004	±0.006	±0.009	±0.010	±0.004	±0.006	±0.015	±0.017	±0.011
MC $q=10$	0.393	0.398	0.468	0.598	0.630	0.398	0.471	1.037	2.365	12.051
	±0.002	±0.003	±0.004	±0.005	±0.005	±0.003	±0.004	±0.013	±0.019	±0.036
MC $q=10^2$	0.399	0.402	0.466	0.589	0.620	0.402	0.468	0.980	2.269	12.084
	±0.003	±0.003	±0.003	±0.003	±0.004	±0.003	±0.003	±0.006	±0.020	±0.135
MC $q=10^3$	0.399	0.405	0.469	0.593	0.625	0.405	0.471	0.993	2.302	12.173
	±0.003	±0.003	±0.002	±0.003	±0.004	±0.003	±0.002	±0.005	±0.009	±0.128
PGD-10	0.731	0.733	0.734	0.743	0.761	0.733	0.734	0.743	0.796	1.411
	±0.005	±0.005	±0.005	±0.005	±0.005	±0.005	±0.005	±0.005	±0.003	±0.009

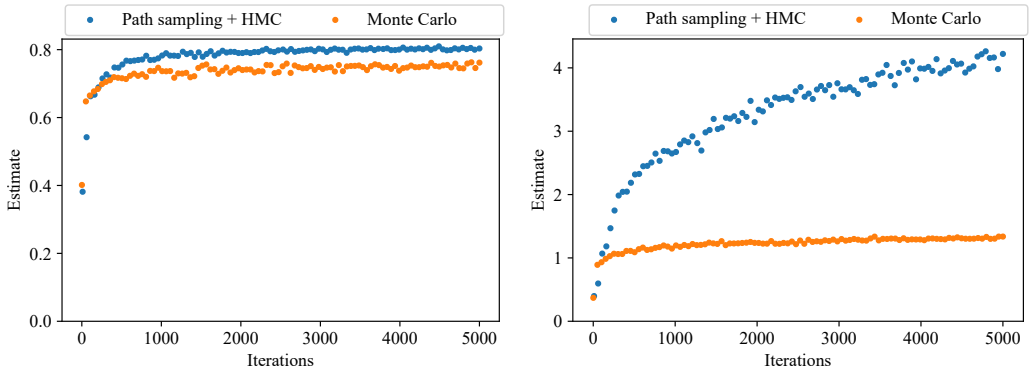
Table 7: Mean and standard deviation of robustness estimates over 3 training runs with different random seeds for experiments on CIFAR-10 with non-differentiable parameterizations of flips, rotation, translation, and scaling. The results show that the intermediate-q robustness metrics, $\hat{Z}_{\text{MC}}$ and $\hat{Z}_{\text{PS}}$, interpolate between loss over random samples (when $q = 1$) and adversarial loss (when $q = \infty$), with the path sampling estimator consistently resulting in higher (better) estimates for larger values of q. However, training for intermediate-q robust performance on CIFAR-10 on this perturbation distribution is more challenging due to the computional complexity.

Train method	Standard	$\hat{Z}_{\text{MC}}$				$\hat{Z}_{\text{PS}}$				Adv. loss
		$q=1$	$q=10$	$q=10^2$	$q=10^3$	$q=1$	$q=10$	$q=10^2$	$q=10^3$	
Standard	0.186	0.450	2.268	3.687	3.865	0.444	2.450	4.636	4.889	5.625
	±0.007	±0.017	±0.067	±0.095	±0.114	±0.017	±0.068	±0.113	±0.111	±0.134
MC $q=1$	0.154	0.191	0.800	1.246	1.300	0.186	0.879	1.615	1.711	2.021
	±0.011	±0.002	±0.026	±0.024	±0.023	±0.004	±0.018	±0.039	±0.018	±0.036
MC $q=10$	0.963	1.019	1.086	1.348	1.423	1.015	1.078	1.416	1.502	1.596
	±0.014	±0.012	±0.010	±0.027	±0.029	±0.012	±0.010	±0.033	±0.078	±0.038
MC $q=10^2$	2.014	2.131	2.131	2.137	2.164	2.130	2.131	2.136	2.171	2.190
	±0.001	±0.002	±0.001	±0.002	±0.001	±0.002	±0.002	±0.002	±0.001	±0.006

perturbation distribution on CIFAR-10 ($\epsilon = 0.3$). Again, we see that HMC-based path sampling provides a much better estimate of the integral than random sampling, especially for higher q, for the same number of iterations (where iterations for the MC estimate is equal to m samples, and iterations for the PS+HMC estimate is equal to m samples times L leapfrog steps). Additionally, convergence plots for robustness estimates over the spatial transformations on CIFAR-10 (described in Section 5.2) are shown in Figure 3, specifically for $q = 1$ and $q = 100$ on a standard trained model over the spatial transformations. We see that path sampling and Monte Carlo sampling converge to nearly the same estimate for $q = 1$, but quickly diverge for $q = 100$, showing that path sampling works much better as an estimator even when the perturbation set is non-differentiable and we do not have the advantage of the Hamiltonian Monte Carlo sampler.

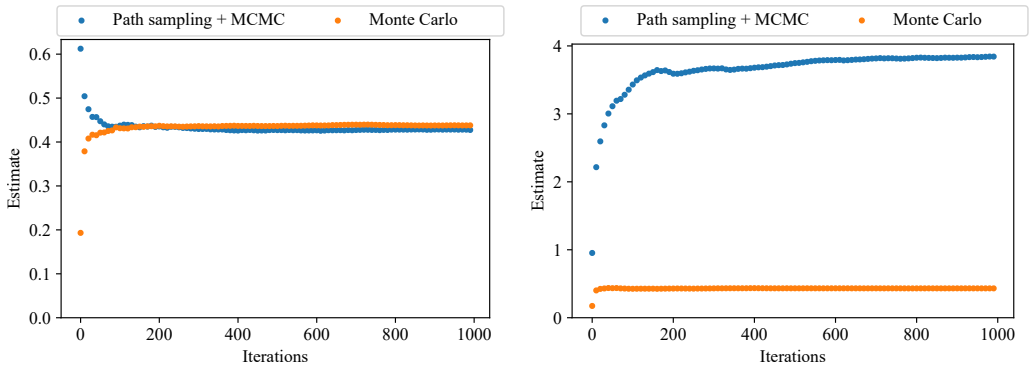
A.3 Experimental details

On both the ℓ_∞ experiments and the spatial transform CIFAR-10 experiments, we train a PreAct ResNet18 architecture using the SGD optimizer for 200 epochs with a starting learning rate of 0.1, Nesterov momentum of 0.9, and weight decay 0.0005. On MNIST we train using the Adam optimizer for 10 epochs with a starting learning rate of 0.001. We use a convolutional ReLU architecture with two convolutional layers with 32 and 64 channels and kernel sizes of 4×4, which are followed by a



(a) (b)

Figure 2: Comparison of the convergence of the path sampling with HMC estimate ($\hat{Z}_{\text{PS+HMC}}$) and Monte Carlo sampling estimate ($\hat{Z}_{\text{MC}}$) of the functional q-norm of the loss over the ℓ_∞-norm ball perturbation distribution for (a) $q = 10$, and (b) $q = 1000$ with increasing iterations on a standard trained model on CIFAR-10. Iterations corresponds m for the Monte Carlo estimator, and to $m \times L$ for the path sampling + HMC estimator, where we fix $L = 10$. For a smaller $q = 10$ (a), path sampling with HMC is slightly better than Monte Carlo sampling. For a larger $q = 1000$ (b), using path sampling with HMC allows for a much better estimate of robustness.



(a) (b)

Figure 3: Comparison of the convergence of the path sampling ($\hat{Z}_{\text{PS}}$) and Monte Carlo sampling estimate ($\hat{Z}_{\text{MC}}$) of the functional q-norm of the loss over the spatial transforms for (a) $q = 1$, and (b) $q = 100$ with increasing iterations (corresponding to number of samples m) on a standard trained model on CIFAR-10. For a smaller $q = 1$ (a), Monte Carlo sampling and path sampling converge to the nearly the same estimate, whereas for a larger $q = 100$ (b), the estimates diverge quickly, with path sampling providing a much better estimate given the same number of samples.

fully connected layer with 1024 units. Each training and evaluation run is performed using a single Quadro RTX 8000 GPU.

A.3.1 ℓ_∞-norm ball

We use a learning rate schedule that divides the starting learning rate by 10 halfway and two thirds of the way through training. On CIFAR-10 training, we train for 200 epochs and we also do *not* use random flip/crop data augmentation that is typically used for training CIFAR-10. For adversarial training and evaluation, we set the step size such that $\alpha = 2.5 \cdot \epsilon/m$, where m is the number of PGD steps, and we use early stopping for adversarial training based on the adversarial validation loss. For training and evaluation runs using HMC-based path sampling, we use $\sigma = 0.1$, and set the step size

such that $\alpha = \rho \cdot \sigma^2/L$. For $q = 1$ and $q = 10$, we set $\rho = 0.6$, for $q = 100$ we set $\rho = 0.4$, and for $q = 1000$, we set $\rho = 0.2$.

A.3.2 Non-differentiable spatial transforms

We perform standard training for 50 epochs, and training using the Monte Carlo sampling estimator (for $q = 1$, $q = 10$, and $q = 100$) for 200 epochs. We use a learning rate schedule that linearly increases from 0 to the maximum value of 0.1 for the first two fifths of training epochs, and then linearly decreases to 0. When training using the MC estimator, we do not perform random flip/crop data augmentation.