Dataset			OSSN	RKO	SVCM	BCOP
CIFAR10 ($\epsilon = 36/255$)	**Small**	Clean	61.83 ± 0.86	61.75 ± 0.57	61.88 ± 0.52	$\mathbf{64.25} \pm 0.39$
		Robust	47.77 ± 0.74	47.39 ± 0.44	47.17 ± 0.41	$\mathbf{49.95} \pm 0.17$
	Large	Clean	68.28 ± 0.51	69.47 ± 0.24	69.44 ± 0.26	$\mathbf{72.01} \pm 0.24$
		Robust	54.26 ± 0.40	55.41 ± 0.21	53.57 ± 0.18	$\mathbf{58.26} \pm 0.17$

We thank all the reviewers for their thorough feedback. We have updated all tables to report results from 5 repeated trials. Our reported improvements are consistent throughout. We would like to re-emphasize our main contributions here: **(1)** To the best of our knowledge, we are the first to reveal the disconnectedness of the space of orthogonal convolutions. We believe our analysis demonstrates this space is unexpectedly complicated and inherently difficult to optimize over. **(2)** We analyze and identify the shortcomings of existing methods of enforcing Lipschitz-constrained convolutions. In particular, we find gradient attenuation to be a common problem among many of these methods and propose using orthogonal convolutions to circumvent this. **(3)** We adapt Xiao et al. [40]*'s orthogonal convolution initialization procedure to be used for optimizing over the orthogonal convolution space. Our parameterization alleviates the issues of the disconnected orthogonal convolution space that arose in our analysis. We verified its effectiveness on adversarial robustness and Wasserstein distance estimation tasks over the pre-existing methods.

Reviewer 1: All empirical results now include error bars (see example table, top). We observed statistical significance throughout. We discuss other points below.

Quality - (1) Our BCOP parameterization lies in the space of orthogonal convolutions, which is 1-Lipschitz only under the L_2 metric. We will make clear that we focus on Lipschitz convolutional networks with the L_2 metric only.

Quality - (2a) To clarify, the statement was trying to demonstrate a relationship between the gradient norm before and after back-propagating through a 1-Lipschitz function. To be precise, let $\mathbf{y} = f(\mathbf{x})$ for some 1-Lipschitz f, and $\mathcal{L}(\mathbf{y})$ be a loss function. We have $\left\|\frac{\partial \mathcal{L}}{\partial \mathbf{x}}\right\|_2 = \left\|\frac{\partial \mathcal{L}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{x}}\right\|_2 \leq \left\|\frac{\partial \mathcal{L}}{\partial \mathbf{y}}\right\|_2 \left\|\frac{\partial \mathbf{y}}{\partial \mathbf{x}}\right\|_2 \leq \mathrm{Lip}(f) \left\|\frac{\partial \mathcal{L}}{\partial \mathbf{y}}\right\|_2 = \left\|\frac{\partial \mathcal{L}}{\partial \mathbf{y}}\right\|_2$, where $\left\|\frac{\partial \mathcal{L}}{\partial \mathbf{x}}\right\|_2$ and $\left\|\frac{\partial \mathcal{L}}{\partial \mathbf{y}}\right\|_2$ are the input and output gradient norm correspondingly, and $\mathrm{Lip}(f)$ is the Lipschitz constant of the function f. We will adjust the phrasing of the statement in the paper to include this detailed explanation.

Quality - (2b, 2c, 2d) Gouk et al. [15]* [1] write that OSSN will "project it back to the closest matrix in the feasible set measured by the matrix distance metric induced by taking the operator norm". One can prove that OSSN is a valid projection under 2-norm but not Frobenius norm (while SV clipping is a valid projection for both norms as R1 suggested). Because OSSN uses a different norm for the steepest descent direction and the projection step, it's not guaranteed to converge; we give a counterexample in Section A of the supplemental material.

Quality - (2e) By "reshaping a kernel into a matrix", we were referring to **flattening** a 2-D convolution kernel tensor of shape (c_o, c_i, k, k) into a $c_o \times c_i k^2$ matrix, where c_i, c_o, k are input channel size, output channel size and kernel size, respectively; whereas, the **matrix form** of the convolution operator is a $hwc_o \times hwc_i$ matrix (h, w are the input/output spatial dimensions). Tsuzuku et al. [36]* has shown a constant factor of the spectral norm of the **"reshaped/flattened kernel"** bounds the Lipschitz constant of the convolution operator, arising from the repetitions of each convolution kernel tensor element in the **matrix form** due to overlapping convolution windows.

Quality - (2f) The optimal dual function must have gradient norm 1 almost everywhere on the support (see Corollary 1 in Gemicic et al. [41]), which can be achieved by gradient norm preservation throughout the network. However, we did not mean to imply that limiting the function space we are optimizing over to be gradient norm preserving is theoretically the best way to estimate Wasserstein distance. We will adjust the writing and supply the additional references accordingly.

Clarity As pointed out by the reviewer, "Lipschitz network" indeed refers to a network with a specified Lipschitz constant that is enforced tightly. This will be clarified. We will re-organize method and experiment sections to clarify notations and key experimental details.

Reviewer 2 expressed concerns over novelty of this work. As discussed above, we do not simply combine methods from Xiao et al. [40]* and Anil et al. [1]*. Xiao et al. [40]*'s algorithm is used for initializing orthogonal convolutions while we need to parameterize the orthogonal convolution space to be optimized over. Moreover, our theoretical analysis enables our BCOP parameterization to be configured to maximize the expressiveness of the orthogonal convolution.

Reviewer 3 inquired about run-time comparison of our Lipschitz convolutional network against standard non-Lipschitz convolutional network. During the training of the "large" architecture described in the paper, [2] the BCOP-parameterized network takes 0.138 seconds per training iteration while a standard non-Lipschitz network with the same architecture only takes 0.041 seconds per training iteration. As for the other Lipschitz methods, RKO takes 0.120 seconds and OSSN (with one power iteration) takes 0.113 seconds. We will report these values in the paper.

Additional Reference: [41] Mevlana Gemici, Zeynep Akata, and Max Welling. Primal-dual Wasserstein GAN. 2018.

[1] The starred references are from the original paper. Any additional references are provided below.

[2] Training speed benchmark setup: CIFAR10, NVIDIA P100, batch size of 128.