We thank reviewers for their valuable comments. We respond to the main concerns below.

[R1/R2] Infinite width assumption: the infinite width assumption is needed due to the technical detail that the norm preservation of activations and gradients are studied in expectation over weights and not for a particular instance of weights. Taking expectation is technically equivalent to considering infinite width and this strategy has been used in previous papers that have studied weight initialization for un-normalized networks [5, 9].

[R1] Setup in Fig. 2 (right): we will clarify this in the revised manuscript. Both plots in Fig. 2 report results for the same experiment, with the grid search parameters described in Table 1 in the appendix. The left plot shows the best result at each depth, whereas the one on the right plots the accuracy for every job in our hyperparameter sweep.

[R1] Very deep ResNet experiments: Similar to that in Zhang et al. [31], we chose 10k block ResNet to stress the point that our initialization indeed prevents gradient explosion/vanishing problem because otherwise we cannot train at such depth. We run this experiment for 1 epoch for two reasons: (1) computational expense, and (2) to show that training does not diverge at such depth when using our initialization with large learning rates compared to baselines.

[R1] Train/test split (L243): our intention was to highlight that the experimental setup is slightly different from that in other works using the same architectures. We will rephrase L243 to better express this. We agree that training on the complete train+val set with the best hyperparameters would boost performance, but we would also lose the ability to track overfitting and perform early stopping. In order to reduce the impact of these factors and provide a fair comparison, we decided to adopt the standard train/val/test split and follow best practices in hyperparameter tuning.

[R2] Surface area of the unit ball: while the closed form formula is available for high dimensional spheres, the ratio seems hard to compute analytically because the constants and ratio of surface area do not trivially cancel out.

[R2] Derivatives wrt $\mathbf{a}^l$: it is sufficient to study the derivative with respect to pre-activations to show gradient explosion/vanishing does not happen for weights. Derivative of weights depend on this term due to the chain rule.

[R2] Layer width in Fig. 1: width does indeed vary in Fig 1, as can be seen in the first line of `get_norm_ratios` in the provided notebook (`synthetic_data_experiment.ipynb`). We will make this explicit in the revised manuscript.

[R2] Orthogonal init when $\mathbf{n_l > n_{l-1}}$: we followed the standard practice of orthogonal initialization, i.e. we orthogonalize columns instead of rows as an approximation when $n_l > n_{l-1}$.

[R3] Initialization scheme (forward/backward) for ResNets: the initialization derived for forward and backward pass for ResNet are identical. For fully connected layers within residual blocks, we use the initialization derived in forward pass of fully connected layers (c.f. L180-183).

[R3] Normalizing outputs of the ReLU as well: an analysis of normalized ReLU output is beyond the scope of this submission, as we focus on Weight Normalized networks (which only normalizes weights of the network). Therefore, we believe that a comparison with the kernel counterparts of deep network would be distracting to the message of this paper since our goal is specifically to study explosion/vanishing of activation and gradient in this network architecture.

[R3] Complete definitions for ResNet: everything is the same as in previous works except for the structure of residual blocks (c.f. L176). This difference is described in L178 and illustrated in Figure 1 (right) in the supplementary material.

[R1/R3] Additional WN baselines in Table 1: we report two additional baseline results (proposed and data dependent init without warmup) for each architecture and dataset in Table 1 below as suggested by the reviewers. Without warmup, proposed init is better than data dependent init. Warmup improves the performance of proposed init (except on wide ResNet, which we suspect happens due to large width where our theory holds more strongly).

Dataset	**Architecture**	**Method**	**Test Error** (%)
CIFAR-10	ResNet-56	WN (proposed init + warmup)	7.20 ± 0.12
		WN (proposed init + no warmup)	7.87 ± 0.14
		WN (datadep init + no warmup)	9.19 ± 0.24
	ResNet-110	WN (proposed init + warmup)	6.69 ± 0.11
		WN (proposed init + no warmup)	7.71 ± 0.14
		WN (datadep init + no warmup)	9.33 ± 0.10
	WRN-40-10	WN (proposed init + warmup + cutout)	4.75 ± 0.08
		WN (proposed init + no warmup + cutout)	4.74 ± 0.14
		WN (datadep init + no warmup + cutout)	6.10 ± 0.23
CIFAR-100	ResNet-164	WN (proposed init + warmup + cutout)	25.31 ± 0.26
		WN (proposed init + no warmup + cutout)	27.30 ± 0.49
		WN (datadep init + no warmup + cutout)	30.26 ± 0.51