We thank the reviewers for the valuable feedback. We will reflect minor errors instantly and try suggestions in the future. We want to address concerns and to clarify misconceptions. In this rebuttal, CAP denotes Wong et al. [R2 R3] Under $\ell_2$-norm, IBP does not work well enough. In IBP, the authors didn’t mention the certifiable training for $\ell_2$-norm. In CROWN-IBP, however, the authors explained that IBP can be applied to $\ell_2$-norm. We run the code for IBP provided by the authors of CROWN-IBP on CIFAR-10 under $\ell_2$-setting with $\ell_2$-norm $3^{|\Omega|}$, where we considered a wide range of parameters including those suggested by CROWN-IBP and our range of settings described in Section C. IBP achieved the verification (standard) accuracy of 22.6-23.0% (31.3-33.5%) which was inferior to CAP = 50.29% (60.14%) and to LMT = 37.20% (56.49%). It also implies LMT-bound is much tighter than IBP-bound under $\ell_2$-norm. Thus, we compared our method to LMT and CAP rather than IBP under $\ell_2$-case in the main text. [R2 R3] On the novelty, BCP was carefully designed in a layer-wise manner (Fig1) to obtain the tighter outer bound. One may simply use IBP to get the additional box constraint in (11), but this box constraint is redundant because it is much looser than the $\ell_2$-constraint in (11). BCP is designed to provide a nonredundant box constraint in (11) to tighten the bound. We observed the tightness in Fig2 and Fig3. As a result, BCP outperforms both LMT and IBP in a large margin (>12-28%) under $\ell_2$-norm. For further intuition behind the layerwise design of BCP, refer to Section B.1. [R2 R3] BCP outperforms the others with a meaningful margin. In Tab 1, the evaluation results at a single $\epsilon_{eval}$ seem to be a marginal improvement compared to IBP. However, when considering a wide range of $\epsilon_{eval}$ in Fig4 (and in FigS3), BCP outperforms CAP by 3.7-5.6% in standard accuracy on CIFAR-10. For $\epsilon_{eval} > 3^{|\Omega|}$, BCP outperforms CAP in a large margin. For example, when evaluating at $7^{|\Omega|}$, BCP (34.2%) defeats CAP (23.9%) by 10.3%. Moreover, only BCP can achieve a meaningful verification accuracy on Tiny ImagingNet, while others cannot. [R1 R2] Fig2 illustrates how BCP can tighten the outer region by introducing the box constraint. In Fig2 (a)-(c), we can easily visualize the high-dimensional ellipsoid $h_{\ell_2}(B_2^{(K-1)}) \subset \mathbb{R}^2$ in 2D plane with $\zeta_2$- and $\zeta_m$-axes (line 211-212) by projection. However, a high-dimensional parallellogram $h_{\ell_2}(B_{\infty}^{(K-1)})$ for $c > 2$ is hard to visualize in the 2D plane. Thus, we use the red lines in Fig2 to indicate that the projection of the outer region $h_{\ell_2}(B_2^{(K-1)} \cap B_{\infty}^{(K-1)})$ must lie above the red line and inside the ellipsoid. Based on (9), we used the verification boundary $(\zeta_y - \zeta_m \geq \zeta_y - \zeta_{m'}$, where the red line is obtained from the solution $\zeta_y', \zeta_{m'}$ of (9) for $\ell_2(B(x)) = h_{\ell_2}(B_2^{(K-1)} \cap B_{\infty}^{(K-1)})$, and the blue line is for $\zeta(B(x)) = \chi_{\ell_2}(B_2^{(K-1)})$. Fig2 (d) explicitly illustrates the ellipsoid and the parallelgram with the verification boundary for a toy binary classification ($c = 2$). Fig2 shows typical examples for which the verification succeeds with BCP but fails without BCP (R2-8-3). On comparing the gap between BCP and other baselines, the visualization of the layerwise design of BCP, as well as achieving the better performance in a wide range of $\epsilon_{eval}$ (Fig4). BCP can outperform LMT and IBP in a large margin with affordable computational overhead.

**[R3] On the implementation of BCP.** Our main focus is on $\ell_2$-certifiable training. In the main text, we describe the BCP algorithm and provide the results for $\ell_2$-case, so, in the code, we set $\text{args.linfty} = \text{False}$ as default. However, BCP can be applied to $\ell_p$-norm for any $p \in [0, \infty]$ (line 154-155). In Appendix, we present the results for $\ell_{\infty}$-norm in Tab S2, and the implementation is available by setting $\text{args.linfty} = \text{True}$ in the code (BCP.py line 216-217). Therefore, the description of the algorithm, including the $\ell_{\infty}$-case (line 154-159), is consistent with the implementation. Moreover, for reproducibility, we provided how to run the code in one-line command in README.md and explained details of the hyper-parameters in Section C, including the scheduling on $\epsilon$ and $\lambda$. For reference, in the $\ell_{\infty}$-case, BCP is not IBP but a generalized version of IBP because BCP still uses the $\ell_2$-bound propagation. [R3] $c$ and $z^{(K-1)}$ have the same shape. The notation $W_i$ is for the $i$-th row vector of the matrix $W$ (line 144). Let $W^{(K)} \in \mathbb{R}^{m \times n}$, then $W_i^{(K)}$ is an $n$-dimensional row vector, and $z^{(K-1)}$ is an $n$-dimensional column vector. Therefore, since $c^T = W_0^{(K)} - W_i^{(K)}$, $c$ has the same dimension to $z^{(K-1)}$. [R1] Trade-off between efficiency and accuracy. BCP is much faster than CAP (Tab1) as well as achieving the better performance in a wide range of $\epsilon_{eval}$ (Fig4). BCP can outperform LMT and IBP in a large margin with affordable computational overhead.