General questions

▷ **Lack of novelty (R1, R5)** We humbly disagree that our proposed method lacks novelty.
As noted by both R2 and R3, our paper raised, formulated and addressed a novel problem to achieve an "in-situ" flexible model that can trade-off between accuracy and robustness at test time based on user specification, for the first time in the adversarial defense literature. Our proposed framework is featured by a joint model-data sampling method, which takes the trade-off parameter λ as a conditional input and trains the network with sampling this additional parameter differently at each minibatch (the loss is adapted accordingly). The above problem definition and "joint model-data sampling method" are the main novelties of the paper, as acknowledged by both R2 and R3.
Although we leverage dual BN and FiLM as the off-the-shelf tools (we never claimed them as our novelty), their usages are also properly customized to fit our framework. For dual BN, we observe that the conflict between standard and adversarial feature statistics is a roadblock for our new problem of packing them in one model, and dual BN [16] can be an effective solution. Difference between our modified and the original version in [16] are summarized in lines 206-217. For FiLM, we condition on λ while the original condition on input data.
▷ **Training cost (R1, R2)** As mentioned in footnote 1 (page 8), CATS models only have a tiny FLOPs overhead (around 1.7% on ResNet34) compared with PGD-ATS baseline models. CAT requires almost the same number of epochs to converge compared with baseline PGD-AT, and thus almost identical training time. (To R1) The model in Fig 3 is actually not large, since the dual BN structure only introduces a tiny overhead (brought by an auxiliary BN), and the S-BN enables weight sharing over channels with different widths. We will make this clearer in revision.

Response to R1

▷ **Sampling strategy of λ.** For both the dedicated training baseline (PGD-AT) and our method, increasing λ will increase RA and sacrifice SA, no matter how densely λ is sampled. We empirically find our method robust to various λ sampling strategies. Increasing λ sampling set size (see lines 302-310) or uniformly sampling from the continuous set $[0, 1]$ (see lines 520-523 in Appx D) achieve very similar performance with our default sampling strategy from $\mathbb{S}_1$. For non-uniform sampling, if we sample from $\{0, 0.1, 0.2, 0.3, 0.4, 1\}$ with probabilities $\{0.3, 0.2, 0.2, 0.1, 0.1, 0.1\}$ respectively, the performance is still similar with our default strategy: within $\pm 0.5\%$ for SA/RA on CIFAR10.
▷ **Utility to explore trade-off.** We do not intend to claim CAT as any "ultimate" tool to solve or analyze the trade-off. One application of CAT is to sketch (approximately estimate) the empirical achievable trade-off between accuracy and robustness of the same model trained under different λ values, in a cheap and efficient way (avoiding training many times). From our experimental results, the SA-RA trade-off curves produced by CAT are highly aligned with dedicatedly trained ones. We will make this clearer in final version. Theory for CAT will be our immediate next work.

Response to R2 & R3

Thanks for appreciating our work. Your suggestions on writings, related works, and more experiments (including ImageNet, medical image classification, or other human-in-loop decision making systems) are highly valuable and will be addressed in the final version. (R2) We will further improve the performance in our future work. (R2) In general, users set $\lambda = 0$ to maximize standard accuracy, and increase λ towards 1 when more robustness is demanded at some price of accuracy. In practical applications, one "rule of thumb" suggestion would be to quickly examine SA and RA at a few anchor λs such as 0, 0.5, 1, and then test further in one most desired interval. (R3) When fed with $\lambda < 0$ or $\lambda > 1$ at test time, the performance of CAT gradually decays as expected, since it is designed to fit λs within range $[0, 1]$ only.

Response to R5

▷ **Relation with ensembling/NAS?** We agree that there exist relationships, and will discuss in the final version.
▷ **Interpretation of Fig 4, 6.** We apologize for the confusion. The trade-offs in Fig 4 and 6 are indeed over different parameters: λ varies from the largest to the smallest value in $\mathbb{S}_1$, for points from top-left to bottom-right on each curve, as noted in the figure captions. Figs 4,6 show that the performance of CAT aligns with the SOTA dedicated adversarial training, that can be empirically considered as the model's best achievable trade-off. Also, our method is verified to generalize to **unseen** λ values (see Fig 5), so it is evidently more than "simply putting together different subnetworks".
▷ **Formulation and term issues.** We will update Eq (1) as you suggested. For PGD-AT and MMA, we can include clean images in adversarial training as a standard variant (e.g., Appx B in [7], Eq. (8) in [22]), so that $\lambda \neq 1$ and $\mathcal{L}_c \neq 0$. For TRADES, we can scale the trade-off parameters to achieve sum-one, whose optimization remained unchanged. We will make our notations stricter in revision, meanwhile none of them affected our method or claims.
▷ **Evaluation on different attacks; Fig 5 on CIFAR10.** We evaluated our models defended with PGD-7 on three other different attacks (including a stronger PGD-20) in Appx B (on SVHN). We also provide RAs against PGD-40 on CIFAR10 per your suggestion here: (λ from 0 to 1 in $\mathbb{S}_1$) **Baseline (PGD-AT)**: 1.04%, 46.5%, 47.9%, 51.1%, 51.3%, 53.2%. **Ours (CAT)**: 17.3%, 49.2%, 50.1%, 50.42%, 50.46%, 51.8%.
Results in Fig 5 can also generalize to CIFAR10: (λ from 0 to 1 in $\mathbb{S}_1 \cup \mathbb{S}_3$) **Baseline (PGD-AT)**: (SA) 94.83%, 91.81%, 89.47%, 88.17%, 88.07%, 87.9%, 87.81%, 86.90%, 86.88%, 86.68%, 86.58%, 86.57%; (RA) 1.12%, 47.1%, 47.41%, 48.25%, 48.9%, 50.61%, 51.83%, 51.99%, 53.17%, 53.35%, 53.43%, 53.52%. **Ours (CAT)**: (SA) 93.12%, 89.82%, 89.57%, 89.28%, 89.15%, 89.07%, 88.98%, 88.88%, 88.53%, 88.32%, 88.28%, 88.04%; (RA) 17.54%, 50.21%, 50.33%, 50.76%, 50.77%, 50.82%, 50.85%, 50.86%, 51.05%, 51.64%, 51.89%, 51.91%.