We thank all reviewers for their careful reading of the manuscript and their constructive comments.

Reviewer-1: The search space of n. In Figure 2(b), $log(n)$ has to be an integer, which is $9 \leq log(n) \leq 14$.

Reviewer-1: The latency per slot. Instead of the latency per slot, we reported the end-to-end inference latency of a privacy-preserving neural network as the key performance metric. First, we would like to emphasize that the numbers of n and q we provided in Figure 3 and 6 are **all in the log scale**. We also would like to point out that the comment of "The range for q also shows about a 20% variation" is **inaccurate**. What we have is "The range for log(q) also shows about a 20% variation". Second, although we agree with the reviewer on the analysis on the latency per slot, we think the reviewer misunderstood the latency per slot on our baseline using the same n and q for all layers. Different neural network layers have different numbers of input and output channels, and different weight kernel sizes. Therefore, different layers in a neural network require different values of n to pack all weights. Our baseline selected a large n to have enough slots to pack the layer with the largest number of weights. However, for the other layers, most slots are empty since they have less weights. In this way, the "effective" latency per slot of our baseline is very long, where the effective latency means the latency per non-empty slot.

Reviewer-1: The 70% latency reduction. We would like to emphasize the fact that the numbers of n and q we provided in Figure 3 and 6 are **all in the log scale** again. We measured the latency of each HE layers on a real machine. By reducing n and q, the cache hit rate greatly increases during the NTT and CRT computations. Therefore, we observed a great latency reduction. We do NOT think simply scaling the latency per slot with n and q is a good latency estimation.

Reviewer-1: Comparison against the fixed aggressive setting. We compared AutoPrivacy against DARL [4] in Table 2 and Figure 6. DARL aggressively sets the same n and q for all layers of a neural network.

Reviewer-1: The models were used. We explained the models we studied in Section 4. We studied a 7-layer CNN network used by [5] (7CNET), ResNet32 [24] (RESNET), and MobileNet-V2 [25] (MOBNET). We quantized all models with 8-bit. Due to the limited space, we cannot include the details of the network architecture in the manuscript. We will try to add the information in the next version of this manuscript.

Reviewer-1: Figure 3 does not represent the runtime per slot. Figure 3 shows the execution time of a HE multiplication we measured on a real machine with different values of n and q. Again, we believe the reviewer underestimated the "effective" latency per slot of our baseline.

Reviewer-2: Analysis on the decryption of multiplied ciphertexts. We believe it is difficult to do a mathematical analysis on the error rate of a neural network with different values of n and q. A mathematical error derivation is too complicated for an inference of a specific privacy-preserving neural network. This is why we propose AutoPrivacy that selects a set of n and q, and feeds them into the HE protocol and real HE-enable neural network to calculate the accuracy. The error tolerance of a privacy-preserving neural network is architecture- and application-dependent.

Reviewer-2: Figure 1(d) Why n does not influence the accuracy. The decryption error is not related to n. n decides the number of slots that can be packed in a cyphertext. Any large convolution can be broken into smaller pieces, so that we can always use a smaller n to perform the computation but with longer latency.

Reviewer-3: Comparison against the other search techniques. In this paper, we present a new and important problem on the parameter selection of privacy-preserving neural networks. Presenting the problem is our first contribution. Our design is the first work to identify the fact that the mathematical error derivation is not necessary for the inferences of a privacy-preserving neural network. We can achieve better inference latency than two of the most recent state-of-the-art designs. We will compare AutoPrivacy against other search techniques in the next version of this manuscript.

Reviewer-3: 19.55 seconds of inference latency on the Cifar dataset isn't practically useful. The inference time is architecture-dependent, but not dataset-dependent. By the architecture of 7CNNet, our fastest inference on CIFAR-10 requires only 6.92 seconds, which is much faster than all existing state-of-the-art designs.