We thank all the reviewers for acknowledging our novel contributions and providing valuable feedback. Below, we address the common concerns followed by detailed comments from each reviewer. We will include additional experimental results and discussions, and revise the final version according to reviewers' suggestions.

From R1, R2, and R3: Discussion of the computational cost. The actual time cost is highly platform-specific. A complete hardware-algorithm codesign is out of the scope of this paper, which mostly focuses on the theoretical properties of gradient quantization. Nevertheless, as requested by the reviewers, we investigate the quantization overhead for a $(N = 128, C = 64, H = W = 56)$ convolutional layer on a single CPU core. In this case, the actual convolution takes 480ms. Finding the range takes 11ms for PTQ and 24ms for PSQ and BHQ. Additionally, it takes BHQ 3µs for finding the optimal transformation (including sorting), and 21ms to perform the transformation with sparse-dense matrix multiplication. Therefore, the overhead for all the quantizers is small relative to the convolution.

As also pointed out by R1 and R3, dedicated hardware implementations may involve more subtleties. Instead of per-sample FP32 ranges, hardware may favor a per-tensor range with per-sample shift values ([0, 4] is enough according to our experience), where the accumulators are shifted before the final summation. We leave this as future work.

From R1 and R2: Evaluation in multiple domains. Upon reviewers' request, we test FQT on a transformer for machine translation on the IWSLT14' En-DE dataset. We quantize all the dense layers in the model and apply per-token quantization for PSQ and BHQ. Following the settings in the original paper, we fix the activation and weight to 8-bit, and vary the gradient bitwidth. The validation BLEU score, as well as the gradient variance, are shown in the table and figure below. The conclusion is the same as with the original paper, where BHQ achieves 3 fewer bits than PTQ.

From R1: No full-precision baseline is used in plots or tables. There could be some misunderstanding. We already reported the full-precision results as "Exact" in all our experiments (in both Figure 3 and Table 1).

From R2: Discuss the results in Table 1. We will discuss in the final version. Specially, we believe the improvement is larger on ResNet50 because its training accuracy reaches to 100% more quickly, so the gradient is sparser.

From R3: The paper does not analyze weight quantization. In our framework, the weight is indeed quantized in the sense that only the quantized weight is required for forward and backward computation (Fig. 1). However, our main focus is on studying the gap between QAT and FQT, i.e., the impact of gradient quantization. Therefore, the analysis of weight quantization, which is a special case of QAT, is out of the scope of this paper. Such analysis can be found in any QAT theory papers (e.g., arXiv 1903.05662), and is compatible with the gradient quantization analysis in this paper.

From R3: Theorem 2 approximates with the largest eigenvalue, which is still a worst-case bound. By taking the statistical approach, we study the exact *bias and variance* of the gradient, rather than the worst-case distance or angle. In this way, we can prove the unbiased gradient (Theorem 1) as well as an **exact** formula of the variance (Eq. 7). These results are not available in the worst-case analysis [20, 22]. The upper bound Eq. 8 mentioned by the reviewer still takes a statistical approach, since it upper bounds the *variance*, not the *distance*. Eq. 8 is more useful for intuitive explanations and designing better quantizers (PTQ and PSQ). Even with this imperfect bound we derive useful quantizers, and it is possible to design even better quantizers (e.g., mixed precision) based on more precise bounds of Eq. 7, where we leave as future work.

More deeply, we can directly analyze Eq. 7, where each term is the impact of the l-th layer quantizer to the k-th layer weight. We compute all the $O(L^2)$ terms and find that quantizers only impact nearby layers, and the impact decays exponentially with $l - k$. In this case, a lower bound consists only the $k = l$ terms is still reasonably accurate. Each term of this lower bound is proportional to the terms in Eq. 8, up to a constant irrelevant with the quantizer parameters.

From R3: Computing the range on the fly is cumbersome. We suspect there is some misunderstanding. Our PSQ quantizes the operands before the actual low-bitwidth multiplication. The accumulator value is known at this point since it is computed by the last layer's matrix multiplication, which is already finished.

Table: Validation BLEU score on the machine translation task.

Alg.	PTQ	PSQ	BHQ
Exact	34.55	–	–
QAT	34.47	–	–
8-bit	34.33	34.39	**34.51**
5-bit	0.02	33.17	**33.70**

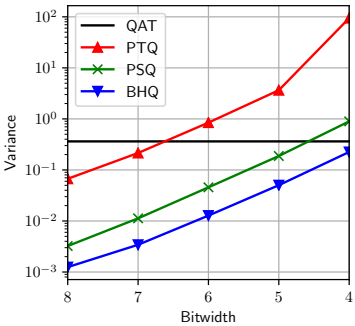

Figure: Gradient variance on the machine translation task.