Online Appendix

We present and discuss the actual experiment results (that we rank the adaptation strategies on) in this appendix.

Table O1 presents the results of the financial sentiment analysis with the phi-3-mini model.

Table O1. Financial Sentiment Analysis Results, phi-3-mini model

Model	Tech	FinanceSA		Train Time	Inference Time	VRAM	Hallucination %
Phi-3-Mini		Acc	F1-macro				
pre-trained, 4-bit	Zero-shot	0.6462	0.6529		6min 50s		0
	Random 3-shot	0.6667	0.6649		12min 6s		17.44
	RAG 3-shot	0.6427	0.6399		12min 16s		20.17
pre-trained, 8-bit	Zero-shot	0.6427	0.6500		6min 1s		0
	Random 3-shot	0.6684	0.6683		6min 7s		0.1
	RAG 3-shot	0.6358	0.6366		6min 3s		0.5
pre-trained, 16 bit	Zero-shot	0.6342	0.6404		35min 29s		11.62
	Random 3-shot	0.6239	0.6301		45min 3s		0.85
	RAG 3-shot	0.6547	0.6538		1h 11min 8s		0
				11.32 mins		2.604 GB	
qLoRA, 4 bit	Zero-shot	0.7846	0.7348		8min 6s		0
	Random 3-shot	0.7846	0.7348		8min 7s		0
	RAG 3-shot	0.7179	0.6834		13min 17s		0
				14.06 minutes		7.547 GB	

LoRA, 16-bit	Zero-shot	0.8034	0.7687		46min		0
	Random 3-shot	0.7812	0.7445		45min 18s		0
	RAG 3-shot	0.7350	0.7102		1h 31min 44s		0

Performance: from the results, we can observe that the best performance (both accuracy and f1-macro wise) came from the model with LoRA fine tuning, and zero-shot inference (accuracy: .8034, f1-macro: .7687). Both are comparable with the community results on Kaggle.com. We can also observe that quantized LoRA (qLoRA) yielded second best performance (accuracy: .7846, f1-macro: .7348). In general, the results confirmed that PEFT (LoRA) increased model performance on a specific downstream task, compared to pre-trained models with in-context learning.

Computational Efficiency: we can compare the computational efficiencies of the models in the fine tuning and inferencing phases, respectively. In terms of fine tuning, qLoRA saved approximately 20% of time (11.32 versus 14.06 minutes) and approximately 66% of VRAM (2.60 versus 7.55 GB), while the performance did not take much hit. This suggests that if the hardware resources are a hard constraint, users should consider using qLoRA to fine tune the model. The efficiency on the inferencing side is more drastic. In terms of inference time, using 4-bit or 8-bit quantization can save 82 - 92% of the inferencing time. This is particularly useful when designing GenAI-backed Apps and (near) real time generation is a consideration. Furthermore, given that performance and computational efficiency is a trade-off, the best of both worlds is the qLoRA model with Zero-shot inferencing.

Robustness to Hallucination: When measuring hallucinations we can observe that LoRA and qLoRA completely eliminated the problem on the Phi-3-mini model. Among the inferencing on the pre-trained models, we observe that RAG reduced hallucinations to 0 on the pre-trained 16-bit model, but not on the 4-but quantized model. The former observation is well supported by the literature (Addlesee, 2024; Ayala & Bechard, 2024). A plausible explanation for the latter observation is that the heavily quantized model may lead the model to generate unexpected contents.

Control/Post-processing: it is observed that for the fine tuned models (both LoRA and qLoRA), we can simply use a regular expression to extract the labels from the generated contents (model outputs). That is not the story for the pretrained models, because the labels can be in different spelling ("Pos" instead of "Positive" or "moderation" instead of "moderated"), encoded (1 instead of "Positive"), or with extra contents ("82.63 percent positive"). In some other cases, the model added incomplete reasoning (limited by the max_new_tokens) that made the label obscure. Customized clean up and mapping functions need to be implemented to extract the labels from pre-trained models.

Table O2 presents the results of the financial sentiment analysis with the tinyllama model.

Table O2. Financial Sentiment Analysis Results, tinyllama model

Model	Tech	FinanceSA		Train Time	Inference Time	VRAM	Hallucination %
tinyllama		Acc	F1-macro				
pre-trained, 4-bit	Zero-shot	0.3452	0.2432		2min 46s		0.06
	Random 3-shot	0.4478	0.3779		3min 57s		19.49
	RAG 3-shot	0.4855	0.4340		3min 59s		15.04
pre-trained, 8-bit	Zero-shot	0.3333	0.2362		4min 20s		0.05
	Random 3-shot	0.4222	0.3733		4min 12s		15.9
	RAG 3-shot	0.5145	0.4814		4min 15s		12.82
pre-trained,16 bit	Zero-shot	0.3265	0.2533		11min 49s		0.03
	Random 3-shot	0.4512	0.4030		20min 29s		21.2
	RAG 3-shot	0.4821	0.4442		20min 32s		12.82
				3.4 minutes		1.797 GB	
qLoRA, 4 bit	Zero-shot	0.4760	0.3440		1min 59s		0
	Random 3-shot	0.4769	0.3516		3min 21s		43.93
	RAG 3-shot	0.5128	0.4036		3min 20s		34.53
				3.67 minutes		3.16 GB	
LoRA, 16-bit	Zero-shot	0.4496	0.4225		14min 11s		2.22
	Random 3-shot	0.4786	0.3851		15min		32.99
	RAG 3-shot	0.5350	0.4659		15min 1s		26.84

Performance: This model yielded inferior performances compared to the Phi-3-mini model, which is intuitive given the number of parameters is much smaller (1.1B vs 3.8B). The best performing configuration is LoRA 16-bit with RAG (accuracy: .5350, F1-macro: .4659). We believe that because of the very limited model size, that RAG actually provided additional information that helped the model make decisions. This observation is further supported by the performances from the pre-trained models, that both random 3-shot and RAG 3-shot improved model performances in both accuracy and f1-macro scores. We suggest that even if the users have very limited resources, they should attempt with small language models like Phi-3-mini without fine tuning, before moving on to even smaller models, with respect to satisfactory model performance.

Computational Efficiency: tinyllama follows a similar trend in computational efficiency, compared to the Phi-3-mini model. In terms of fine tuning, although there was only approximately 5% saving in time between qLoRA and LoRA (3.4 versus 3.67 minutes), the VRAM savings are more significant (~44%, 1.80 versus 3.16 GB). However, we believe that such significant savings are less relevant because of two reasons, First, above-shown performances made even the LoRA model, rather than the memory-efficient qLoRA model, less attractive to users. Secondly, with the memory footprint already being so low, unless the users are considering environments with very limited resources (e.g., mobile), such savings do not make too much sense. We can also observe a significant reduction in inferencing times, compared to the Phi-3-mini model, although it is worth noting that LoRA 16-bit had lower inferencing times than pre-trained 16-bit, particularly in random and RAG 3-shot, which may suggest that users should consider LoRA over using the pre-trained model off the shelf.

Robustness to Hallucination: hallucination is a significant issue in the tinyllama model. We observed that in contrast to the observations on the Phi-3-mini model, LoRA and qLoRA actually introduced heavier hallucinations than pre-trained models. But we also observed that LoRA 16-bit had less hallucinations than qLoRA e.g., 26.84% versus 34.63% in RAG 3-shot inferencing), and RAG did reduce the issue as well, with an average reduction over random 3-shot of approximately 20%.This further suggests that random few-shot inferencing should be a less desired technique in terms of prompt engineering.

Prompt Complexity: as discussed above, the performances of models improve when examples are added to the prompt, and RAG performs better than random 3-shot inferencing. One plausible explanation is that given the very limited parameter numbers, additional information in the examples lent much more help to model performances, and RAG provided richer information than random 3-shot.

Control/Post-processing: the model outputs are of lesser quality than Phi-3-mini, which requires more effort in post processing. We also observe some nonsensical generations, particularly from quantized (both pre-trained and fine-tuned) models.

Table O3 presents the results of the human moderation detection with the phi-3-mini model.

Table O3. Human Moderation Detection Results, phi-3-mini model

Model	Tech	HumanMOD		Train Time	Inference Time	VRAM	Hallucination %
Phi-3-Mini		Acc	F1-macro				
pre-trained, 4-bit	Zero-shot	0.4912	0.4602		5min 37s		1.29
	Random 3-shot	0.5088	0.5069		10min 51s		1.29
	RAG 3-shot	0.6598	0.6486		10min 36s		8.69
pre-trained, 8-bit	Zero-shot	0.4724	0.4664		8min 32s		0.47
	Random 3-shot	0.5159	0.5117		12min 57s		1.29
	RAG 3-shot	0.5628	0.5440		12min 42s		1.76
pre-trained, 16 bit	Zero-shot	0.4806	0.4548		10min 41s		0.23
	Random 3-shot	0.4841	0.4822		15min		0.94
	RAG 3-shot	0.6439	0.6423		14min 43s		1.18
				50.82 minutes		3.32 GB	
qLoRA, 4 bit	Zero-shot	0.6864	0.6863		11min 59s		4.93
	Random 3-shot	0.6676	0.6614		16min 18s		12.1
	RAG 3-shot	0.6593	0.6529		16min 2s		8.7
				64.55 minutes		8.363 GB	
LoRA, 16-bit	Zero-shot	0.7099	0.7097		14min 22s		0.71
	Random 3-shot	0.6969	0.6949		16min 50s		2.12
	RAG 3-shot	0.6699	0.6676		16min 55s		0.82

Performance: the best performing configuration is LoRA 16-bit with zero-shot inferencing (accuracy: .7099, f1-macro: .7097). The second best performance is qLoRA with zero-shot inferencing (accuracy: .6864, f1-macro: .6863). This is consistent with our observation in the previous experiment, that due to limited context window size and prompt structure, few-shot inferencing did not help with model performances. On the other hand, few-shot inferencing did boost the model performance significantly in all configurations involving pre-trained models, particularly with the 4-bit and 16-bit models, suggesting that it is worthwhile to carefully select examples when inferencing from a pre-trained model. We also observe a slightly lower performance, compared to the previous experiment, on a simpler problem (binary versus multi-class classification). Previous literature has suggested that LLMs perform worse on rare problems, even those problems are simpler (Laban et al., 2023). Compared to sentiment analysis, which might be ubiquitous in the pre-training data, human moderation detection is a rarer problem. It is also worth noting that 8-bit quantization does not perform well, consequently it is not very popular in the practice. We recommend the users to consider 4-bit quantization mainly.

Computational Efficiency: quantization again showed significant savings in both time and VRAM. In terms of fine tuning, qLoRA led to an approximately 22% savings in time (50.82 versus 64.55 minutes) and approximately 61% in VRAM (3.32 versus 8.36 GB). Given that the sequence lengths in this dataset are significantly longer than the previous dataset (average: 91.86 versus 32.44 tokens), we suggest that users should give qLoRA more consideration when dealing with longer texts. In terms of inferencing times, we observe that pre-trained 16-bit, qLoRA and LoRA spent comparable time in respective configurations, which suggests: a) qLoRA models do not save on inferencing time compared to LoRA models, although the significant savings in fine tuning time should be taken into consideration; and b) given that pre-trained models do not save on inferencing times, it is better to consider using fine tuned models.

Robustness to Hallucination: overall all configurations had low hallucination ratios besides qLoRA and pre-trained 4-bit with RAG inferencing. This again confirmed that quantization leads to higher chances of hallucinations. However, we did observe that RAG did not help with the hallucination issue, we believe this is attributable to the issue of exceeding the context window size.

Control/Post-processing: The model performed remarkably well in terms of following the output format. To recap, the generation template in the training prompt was "`The social media post is {} by human`". Without giving that template to the model in the inferencing phase, the model strictly followed the structure, aside from the hallucinated cases. This suggests that the Phi-3-mini model is very capable of instruction following, particularly after fine tuning for the downstream tasks.

Table O4 presents the results of the human moderation detection with the tinyllama model.

Table O4. Human Moderation Detection Results, tinyllama model

Model	Tech	HumanMOD		Train Time	Inference Time	VRAM	Hallucination %
tinyllama		Acc	F1-macro				
pre-trained, 4-bit	Zero-shot	0.4947	0.3505		4min 16s		21.73
	Random 3-shot	0.4971	0.3867		4min 52s		15.85
	RAG 3-shot	0.4971	0.3867		4min 54s		12.67
pre-trained, 8-bit	Zero-shot	0.4947	0.3505		5min 17s		13.26
	Random 3-shot	0.4971	0.3867		5min 32s		11.17
	RAG 3-shot	0.4971	0.3867		5min 41s		9.81
pre-trained,16 bit	Zero-shot	0.4747	0.4301		6min 33s		10.58
	Random 3-shot	0.5159	0.4557		7min 22s		7.76
	RAG 3-shot	0.5159	0.4557		7min 28s		6.81
				12.35 minutes		2.471 GB	
qLoRA, 4 bit	Zero-shot	0.5029	0.4223		6min 7s		16.8
	Random 3-shot	0.4865	0.4241		6min 45s		18.21
	RAG 3-shot	0.5029	0.4223		6min 44s		17.27
				12.82 minutes		4.672 GB	
LoRA, 16-bit	Zero-shot	0.5223	0.5236		8min 2s		0.59
	Random 3-shot	0.5147	0.5103		9min 26s		5.76
	RAG 3-shot	0.5029	0.4619		9min 40s		5.64

Performance: similar to the previous experiment, the performance on the tinyllama model took a big hit besides zero-shot inferencing on the pre-trained model, with or without quantization. Again we believe that the limited model size of tinyllama is the main driver behind this observation. The best performance is zero-shot inferencing on the LoRA 16-bit model (accuracy: .5223, f1-macro: .5236). The pre-trained 16-bit on few-shot inferencing (both random and RAG) beat qLoRA on all inferencing methods. This again suggests that on models with similar sizes of tinyllama, quantization decreases the performance.

Computational Efficiency: in terms of fine tuning, qLoRA brought only an approximately 4% of savings on time (12.35 versus 12.82 minutes), but an approximately 47% saving on VRAM. But again the inferior performance rendered the significant savings less relevant. On the other hand, the pre-trained 16-bit model reached similar performance with lower fine tuning time compared to the LoRA 16-bit model (6.51 - 7.49 versus 8.01 - 9.66 minutes). However, we encourage the users to consider the LoRA model because of the following point.

Robustness to Hallucination: only the LoRA 16-bit model yielded satisfactory level of hallucination reduction (.59 - 5.64%), that is the reason we recommended the users to consider this configuration over the pre-trained 16-bit counterpart. Again, RAG is proven to be effective to reduce hallucination in all configurations.

Control/Post-processing: similar to the previous experiment, the generated outcomes from tinyllama required much more effort to post-process. For instance, the fine tuned models, both qLoRA and LoRA, no longer follow the generation template closely. We again observed nonsensical generations from the models, both pre-trained and fine-tuned.

REFERENCES

Addlesee, A. (2024). Grounding LLMs to In-prompt Instructions: Reducing Hallucinations Caused by Static Pre-training Knowledge. In T. Dinkar, G. Attanasio, A. C. Curry, I. Konstas, D. Hovy, & V. Rieser (Eds.), *Proceedings of Safety4ConvAI: The Third Workshop on Safety for Conversational AI @ LREC-COLING 2024* (pp. 1–7). ELRA and ICCL. https://aclanthology.org/2024.safety4convai-1.1

Ayala, O., & Bechard, P. (2024). Reducing hallucination in structured outputs via Retrieval-Augmented Generation. In Y. Yang, A. Davani, A. Sil, & A. Kumar (Eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)* (pp. 228–238). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.naacl-industry.19

Laban, P., Kryscinski, W., Agarwal, D., Fabbri, A., Xiong, C., Joty, S., & Wu, C.-S. (2023). SummEdits: Measuring LLM Ability at Factual Reasoning Through The Lens of Summarization. In H. Bouamor, J. Pino, & K. Bali (Eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (pp. 9662–9676). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.emnlp-main.600