A companion *in 2 GB*

What a smaller model buys on a phone, what compression costs, and how much "no measurable loss" can be made to prove

MITWA LAB, MUMBAI

IN PLAIN WORDS

Bloom's model used to take up 6.2 GB. The version that ships takes 2.0 GB, and in our tests we could not measure any loss. This note explains why the smaller size matters and how much weight that last phrase can bear. An iPhone 16 has 8 GB of memory, so a 6.2 GB model does not realistically fit and a 2.0 GB model does. A smaller model is also quicker and easier on the battery, roughly in proportion to its size, because the phone has to read through the whole model for every piece of a word it writes. Shrinking a model adds a little noise to it, and we show how fast that noise grows. The careful part is the testing. A test with 143 items cannot rule out a problem that turns up 2% of the time, and one with 60 items cannot rule out 5%. So "no measurable loss" is true and it is limited, and we describe what a stronger test would look like.

ABSTRACT

A companion that runs on a phone has to fit in the phone's memory, and that constraint decides more than whether it runs. During generation every weight is read once per token, which bounds token time below by model size over memory bandwidth, so a model 3.1 times smaller is up to 3.1 times faster and moves 3.1 times fewer bytes per token. Compression has a cost. Using the standard model of quantization error, we show that each bit removed from a weight doubles the relative noise it carries, and that a linear layer passes this noise on at the same relative size. We then ask what an evaluation that finds no loss is entitled to conclude. A test of n paired items with no item worse can exclude a loss rate above roughly $3/n$ and nothing smaller; ruling out a 1% loss takes 299 clean items, and 0.1% takes 2,995. We report one result from Bloom, an on-device companion: the build that ships is 2.0 GB, down from 6.2 GB, with no measurable loss on our evaluations and neither build judged better on a 60-scenario safety suite. By the bound above, tests of 60 and 143 items leave room for losses of up to 4.9% and 2.1%. We list what a compression evaluation should report. How Bloom was compressed is outside the scope of this note.

01 Introduction

Whether a language model fits on a phone sounds like a packaging detail. In practice it settles where the model runs, and with that who can see what is said to it, what each reply costs to produce, and whether the thing works on a plane. A model that is too large for the device has to run on a server, and most of the hard questions about a companion begin there.

Bloom runs entirely on the iPhone. The build we started from was 6.2 GB; the build that ships is 2.0 GB. Every iPhone 16 has 8 GB of memory (9to5Mac, 2024), which the model shares with the operating system and with everything else the phone is doing, so the difference between those two sizes is roughly the difference between a model that cannot be shipped and one that can.

Compression is not free, though, and the usual way of reporting it hides the interesting part. A sentence such as "no measurable loss" describes the test as much as the model. In this note we try to say exactly what the smaller size buys, what compression does to a model in principle, and what a clean evaluation result can and cannot establish. The note makes the following contributions.

- We describe what size does on a device in four terms a user would notice: whether the model fits,

how fast it writes, how much energy a reply takes, and how long it takes to install (Sections 3 and 4).

- We work through the standard model of quantization noise and show that removing a bit from each weight doubles the relative noise, and that a linear layer passes that noise on at the same relative size (Section 5).
- We derive how small a loss an evaluation of a given size can exclude, and how many items it takes to exclude a loss of 1% or 0.1% (Section 6).
- We report the Bloom result, read it against that bound, and set out what a compression evaluation ought to report (Sections 7 to 9).

Bloom is a companion and makes no claim to be a therapist. Nothing a person says to it leaves the phone, which is the reason the model has to fit there in the first place.

02 Related work

Quantization. Storing each weight in fewer bits is the oldest and most widely used way to shrink a model, and the theory of the error it introduces goes back to the 1940s (Gray and Neuhoff, 1998). For language models the main empirical finding is that precision can be cut a long way before accuracy suffers. Across more than 35,000 experiments on models from 19 million to 176 billion parameters, 4-bit weights gave the best accuracy for a given total number of bits, and the relationship reversed at 3 bits (Dettmers and Zettlemoyer, 2023).

Running on the device. Work on on-device inference treats memory as the binding constraint. One line of research keeps the weights in flash storage and brings them into memory on demand, which lets a model up to twice the size of the available memory run at all (Alizadeh et al., 2024). We take the simpler case in which the model has to fit in memory, and ask what fitting buys.

Showing that nothing was lost. Statisticians distinguish between failing to find a difference and showing that there is none. The second requires an equivalence test, in which the largest difference one is willing to ignore is fixed in advance (Lakens, 2017). A closely related result is the rule of three: when nothing goes wrong in a number of trials, the true rate can still be as high as about three divided by that number (Hanley and Lippman-Hand, 1983). We used the same rule in our note on memory (Mitwa Lab, 2026a), and Section 4 below refers to our note on reply length (Mitwa Lab, 2026b).

03 Formulation

A model with N weights stored at an average of b bits each occupies

$$S = \frac{Nb}{8} \text{ bytes} \tag{1}$$

so for a fixed number of weights, size and precision move together. Going from 6.2 GB to 2.0 GB is a factor of 3.1, which for the same weights means 3.1 times fewer bits per weight on average, for example from 16 bits to a little over 5. That is only an example. We do not disclose the number of weights or the precisions Bloom uses.

On the device, the model shares memory with everything else. If the phone has M bytes of memory and the system lets an app use a fraction of it, the model fits only when

$$S + S_{\text{work}} \le \phi M \tag{2}$$

where S_{work} is the working memory needed while generating and φ is the fraction an app may use. Apple does not publish φ, and in practice an app cannot use all of the device's memory. With M = 8 GB, a 6.2 GB model would need 78% of the phone's memory before any working memory is counted, and a 2.0 GB model needs 25%.

04 What a smaller model buys

4.1 It fits

The first benefit is the plain one. By (2), a model that takes more than three quarters of the device's memory leaves too little for the operating system, let alone for the app around it. A model that takes a quarter leaves room. Everything else in this section assumes the model is on the device, and at 6.2 GB on an 8 GB phone that assumption fails.

4.2 It writes faster

A language model writes one token at a time, and to produce each token it reads every weight once. If the device can move R bytes a second from memory to the processor, the time per token cannot be less than

$$t_{\text{token}} \ge \frac{S}{R} \tag{3}$$

Proposition 1 (Bandwidth bound).
When generation is limited by memory bandwidth, time per token is proportional to model size. Two builds of sizes S_1 and S_2 then differ in speed by the factor S_1/S_2, which for Bloom is 3.1.

Table 1 shows what (3) gives at three bandwidths. We have not measured Bloom's speed for this note, and the bandwidths are illustrative; a real device also spends time on arithmetic, which (3) ignores. The ratio between the two builds is the robust part. To connect this with reply length: Bloom's average reply is 11 words (Mitwa Lab, 2026b), on the order of 15 tokens. At 50 GB a second the bound is 1.9 seconds for the larger build and 0.6 seconds for the smaller.

BANDWIDTH	6.2 GB BUILD	2.0 GB BUILD
25 GB/s	248 ms	80 ms
50 GB/s	124 ms	40 ms
100 GB/s	62 ms	20 ms

Table 1. Lower bound on time per token from (3). The bandwidths are illustrative and are not measurements of any phone.

4.3 It uses less energy

Moving data costs energy. If each byte read from memory costs ε joules, the memory energy per token is about εS, so it falls in the same proportion as the size. A reply from the 2.0 GB build moves about a third of the bytes that the same reply would from the 6.2 GB build. As with speed, this is a first-order argument and we have not measured it on a device.

4.4 It installs sooner

The model has to reach the phone before any of this matters. On a 50 megabit connection, 6.2 GB takes about sixteen and a half minutes to download and 2.0 GB a little over five. The smaller build also leaves 4.2 GB of storage free that the larger one would occupy.

4.5 What follows from being on the device

Once the model runs on the phone, two things follow that have nothing to do with speed. Replies cost the maker nothing to produce, since no server is involved, which is what makes it possible to sell Bloom once and not by the month. And what a person says never has to leave the device. We return to the second point in a later note.

05 What compression costs

Quantization replaces each weight with the nearest value on a grid. If the grid spacing is Δ, the rounding error is spread evenly between $-\Delta/2$ and $\Delta/2$, and its variance is $\Delta^2/12$ (Gray and Neuhoff, 1998). Suppose the weights have standard deviation σ and the grid covers k standard deviations either side of zero with b bits, so that $\Delta = 2k\sigma/2^b$. The noise relative to the weights is then

$$\rho(b) = \frac{\Delta/\sqrt{12}}{\sigma} = \frac{k}{\sqrt{3}\,2^b} \tag{4}$$

Proposition 2 (A bit halves the noise).
Under (4), $\rho(b-1) = 2\rho(b)$: each bit removed from a weight doubles its relative noise. For a linear layer with independent rounding errors, the noise in the output relative to the output is also $\rho(b)$, whatever the input.

Figure 1 plots (4) for $k = 4$. At 16 bits the noise is 0.0035% of the signal, at 8 bits it is 0.9%, and at 4 bits it is 14%. In the example of Section 3, going from 16 bits to a little over 5 multiplies the noise by about 1,800, to roughly 6.5%. The second half of the proposition says where that noise goes: a layer hands on to the next one an output that is noisy by the same fraction.

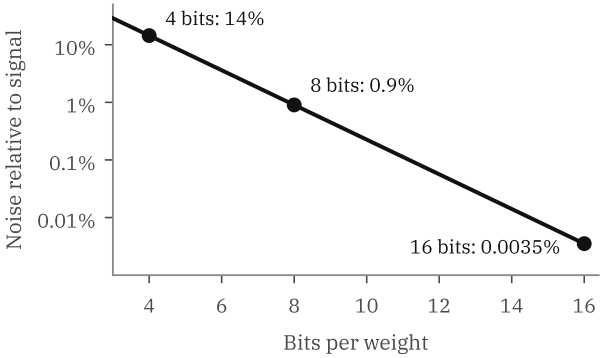


Figure 1. Relative noise from uniform quantization, from (4) with $k = 4$. The vertical axis is logarithmic. Each step of one bit to the left doubles the noise.

A model carrying several per cent of noise in every layer might be expected to fall apart, and the evidence is that it does not. Accuracy holds up down to about 4 bits per weight and then gives way (Dettmers and Zettlemoyer, 2023). The simple model above does not explain why. It treats every weight as equally important and the grid as uniform, and neither is true of a trained network or of the methods used in practice. We take two things from it all the same. The noise is real and it grows quickly as bits are removed, so the absence of a measured loss is a finding that needs checking. And there is a cliff: the last bit removed can cost more than all the ones before it.

06 What "no measurable loss" can show

Suppose the two builds are compared on n items, and on each item the smaller build either matches the larger one or is worse. Let d be the fraction of all possible items on which it would be worse. The chance that a test of n items contains none of them is

$$\Pr(\text{no worse item}) = (1-d)^n \tag{5}$$

which can be large even when d is not small. Turning this round gives the largest d that a clean result leaves open at confidence $1-\delta$:

$$d \le 1-\delta^{1/n} \approx \frac{3}{n} \quad (\delta = 0.05) \tag{6}$$

Proposition 3 (The resolution of a test). *A comparison of n items in which the smaller build is never worse excludes, at 95% confidence, a loss rate above $1-0.05^{1/n}$ and no rate below it. To exclude a loss rate above m with a clean result takes $n \ge \ln(1/\delta)/(-\ln(1-m))$ items, about 3/m.*

Bloom's release checks include a behaviour bank of 143 turns and a safety suite of 60 scenarios. By (6), a clean result on 143 items leaves open a loss on up to 2.1% of items, and a clean result on 60 leaves open up to 4.9% (Figure 2). To bring the first figure down to 1% would take 299 items, and to reach 0.1% would take 2,995.

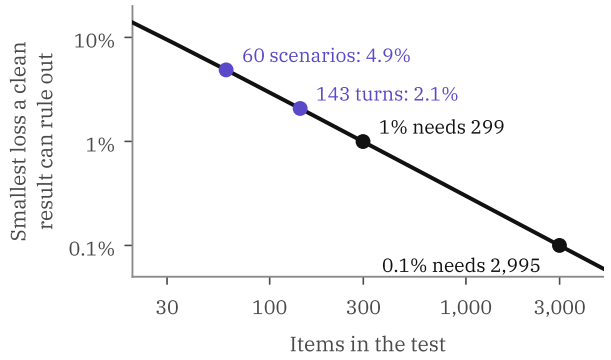


Figure 2. The smallest loss rate that a test with no failures can rule out at 95% confidence, from (6). Both axes are logarithmic. The two marked tests are the sizes of Bloom's behaviour bank and safety suite.

Whether 2.1% matters depends on how often the behaviour comes up. As an illustration, a person who exchanges thirty replies a day sees about 11,000 in a year, and a loss on 2.1% of them would touch some 230. A test of 143 items cannot tell that situation apart from no loss at all. This is the distinction that equivalence testing was designed for: a difference that was not detected has not thereby been shown to be absent, and showing absence means naming, before the test, the largest loss one is prepared to accept (Lakens, 2017).

Two further points make the problem sharper for compression than for most changes. The errors that compression introduces are unlikely to be spread evenly. They tend to show up in rare inputs, which are the ones a small test is least likely to contain. And pass-or-fail comparisons throw away information. A continuous measure of how far the two builds' outputs differ, such as the average divergence between their next-token distributions, can detect a shift long before any item flips from pass to fail.

07 Result

We ask whether the smaller build is smaller in the ways that matter and whether anything was lost. Bloom has been live on the iPhone since 11 June 2026, and the build that ships is the 2.0 GB one. Table 2 collects what follows from the two sizes.

	BEFORE	SHIPPED
Model size	6.2 GB	2.0 GB
Share of an 8 GB phone's memory	78%	25%
Token time bound, relative	1	0.32
Memory energy per token, relative	1	0.32
Download at 50 Mbit/s	16.5 min	5.3 min

Table 2. The two builds. Size is measured; the other rows follow from size by (2) and (3) and the arguments of Section 4, and were not measured on a device.

On quality, we found no measurable loss between the two builds on our evaluations, and on the 60-scenario safety suite neither build was judged better than the other. Each release also goes through the 143-turn behaviour bank and a review by psychologists.

Read against Section 6, that result is reassuring up to a point. It is good evidence against a large loss. It is not evidence against a loss of a few per cent, because tests of this size cannot see one. We are also not reporting the item-level counts, the precision or method used, or any on-device measurement of speed, memory or energy, so the gains in Table 2 beyond size itself are expectations.

08 Limitations and broader impact

Only the two sizes are measurements. The speed and energy figures come from a first-order model in which reading the weights is the bottleneck. On a real phone, arithmetic, caching and the working memory

of the conversation all take a share, and the true gain may be smaller than 3.1.

The noise model in Section 5 is the textbook one. It assumes a uniform grid, independent errors and weights that all matter equally, and it says nothing about why trained networks tolerate the noise as well as they do. We use it to show how the cost of compression scales, and it should not be read as a description of Bloom.

The bound in Section 6 assumes that test items are independent draws from the inputs that matter. A test built by the team that built the model is unlikely to be that, and items that share a conversation carry less information than their number suggests. Both push the real resolution of our tests below what (6) gives.

On broader impact, compression is what puts a private model in a person's pocket, and we think that is worth a great deal. The same privacy means we cannot watch the shipped model at work. If compression has damaged a behaviour that our tests did not cover, we will not see it unless someone tells us, which is a reason to test harder before release and to say plainly what the tests can show. This note makes no clinical claim.

09 What an evaluation should report

From the above, a report that a model was compressed without loss should include at least these.

- The size before and after, and the average bits per weight.
- The largest loss the authors were prepared to accept, fixed before the comparison was run.
- The number of items, and the loss rate the comparison can exclude by (6).
- Item-level counts of better, same and worse, so that readers can compute their own bounds.
- Rare and safety-critical inputs sampled more heavily than their natural frequency, since that is where compression errors are expected.
- A continuous measure of divergence between the two builds' outputs alongside the pass-or-fail results.
- Speed, peak memory and energy per reply measured on a named device, in place of estimates like those in Table 2.

10 Conclusion

Shrinking Bloom from 6.2 GB to 2.0 GB is what allows it to run on an iPhone at all, and by a simple argument it should also make each reply about three times quicker and cheaper in energy. Compression adds noise that grows quickly as precision falls, and yet we could not measure a loss.

We believe that result, and we also know its limits. Tests of 60 and 143 items can exclude a large loss and cannot exclude a small one. The work that remains is to measure the gains on a device, to enlarge the tests until they can exclude a loss of 1% or less, and to add a continuous measure that would notice a shift before any single item fails.

Disclosure. Bloom is a product of Mitwa AI, and this note was written by the lab that builds it. No user conversation is available to us, and none was used. Reproducibility. The results in Sections 4 to 6 are closed-form and proved in the text or in Appendix A, and Table 2 follows from the two sizes. The test sets, the item-level results and the compression method are not released with this note.

References

9to5Mac (2024). Apple's Johny Srouji confirms all iPhone 16 models have 8GB RAM. 13 September 2024. 9to5mac.com/2024/09/13/iphone-16-ram-apple-intelligence/

Alizadeh, K., Mirzadeh, I., Belenko, D., Khatamifard, S. K., Cho, M., Del Mundo, C. C., Rastegari, M., and Farajtabar, M. (2024). LLM in a flash: Efficient large language model inference with limited memory. In Proceedings of ACL 2024. arxiv.org/abs/2312.11514

Dettmers, T., and Zettlemoyer, L. (2023). The case for 4-bit precision: k-bit inference scaling laws. In Proceedings of ICML 2023, PMLR 202, 7750–7774. proceedings.mlr.press/v202/dettmers23a.html

Gray, R. M., and Neuhoff, D. L. (1998). Quantization. IEEE Transactions on Information Theory, 44(6), 2325–2383. doi.org/10.1109/18.720541

Hanley, J. A., and Lippman-Hand, A. (1983). If nothing goes wrong, is everything all right? Interpreting zero numerators. JAMA, 249(13), 1743–1745. doi.org/10.1001/jama.1983.03330370053031

Lakens, D. (2017). Equivalence tests: A practical primer for t tests, correlations, and meta-analyses. Social Psychological and Personality Science, 8(4), 355–362. doi.org/10.1177/1948550617697177

Mitwa Lab (2026a). Memory you can trust: The asymmetric cost of invented memories in AI companions. Technical note 01.

Mitwa Lab (2026b). Eleven words: Reply length in AI companions. Technical note 02.

A Appendix A: Proof of Proposition 2

Replacing b by $b - 1$ in (4) halves 2^b and so doubles ρ. For the layer, write each output as a sum of weights times inputs. If the rounding errors are independent with variance $\Delta^2/12$, the error in an output has variance $(\Delta^2/12)$ times the squared length of the input. If the weights are independent with variance σ^2, the output itself has variance σ^2 times the same squared length. The input cancels in the ratio, leaving $\Delta^2/12\sigma^2 = \rho^2$.