

A CBS Measurement Details

Our empirical method for measuring the CBS Section 3 is, in principle, sensitive to the choice of checkpoints and batch size multipliers. We therefore document the checkpoints and multipliers we used here.

OLMo 1B. When measuring the OLMo 1B CBS with branching, we set the base batch size to 1024 and the base learning rate to $0.0004 \cdot \sqrt{2}$, reflecting the default batch size of 512 and learning rate of 0.0004 in the OLMo codebase under a square-root scaling rule (Malladi et al., 2022). We then chose the following checkpoints and multipliers k :

1. Step 0: k ranging over 0.0625, 0.125, 0.25.
2. Steps 10K, 20K, ..., 50K: k ranging over 0.5, 1, ..., 5.
3. Steps 100K, 150K, ..., 450K: k ranging over 1, 2, ..., 8.

Figure 5 shows loss vs. batch size plots for all checkpoints of OLMo 1B.

OLMo 7B. We set the base batch size to 1024 and the base learning rate to 0.0003, as specified in the OLMo codebase. We then chose the following checkpoints and multipliers k :

1. Step 0: k ranging over 0.0625, 0.125, 0.25.
2. Steps 1K, 2K, 3K: k ranging over 0.25, 0.5, 1, 2, 3, 4.
3. Steps 10K, 20K, 30K: k ranging over 1, 2, 3, 4, 5.
4. Steps 72K, 150K, 200K, 239K, 300K, 350K, 400K, 477K: k ranging over 1, 2, 3, 4, 5, 6, 7, 8.

Appendix A shows loss vs. batch size plots for all checkpoints of OLMo 7B. These checkpoints were chosen manually as we developed this project. Over the course of our experimentation, we launched many additional runs beyond the ones discussed above. Since the choice of k can influence the conclusions of our method, we filtered down the included runs to make the choice of k systematic. 1B runs were launched on a single node of H100 GPUs, and 7B runs were launched on 8 nodes.

B Noise Scale Measurement Details

We use the gradient noise scale estimator proposed by McCandlish et al. (2018, Appendix A) to estimate the gradient noise scale. The method estimates the gradient noise scale using gradient norms at two different batch sizes B_{big} and B_{small} according to:

$$\begin{aligned} \mathcal{B}_{\text{simple}} &\approx \frac{\mathcal{S}}{\|\mathcal{G}\|^2}, \text{ where} \\ \mathcal{S} &= \frac{\|G_{\text{small}}\|^2 - \|G_{\text{big}}\|^2}{1/B_{\text{small}} - 1/B_{\text{big}}} \\ \|\mathcal{G}\|^2 &= \frac{B_{\text{big}}\|G_{\text{big}}\|^2 - B_{\text{small}}\|G_{\text{small}}\|^2}{B_{\text{big}} - B_{\text{small}}}. \end{aligned}$$

We use large batch size $B_{\text{big}} = 64$ and small batch size $B_{\text{small}} = 1$.

It holds that $\mathbb{E}[\mathcal{S}] = \text{tr}(\Sigma)$ and $\mathbb{E}[\|\mathcal{G}\|^2] = \|G\|^2$. We thus average \mathcal{S} and $\|\mathcal{G}\|^2$ over 4096 batches reduce variance and then return their ratio as our estimate of the noise scale $\mathcal{B}_{\text{simple}}$, using offline (i.e., unseen) data in each batch.

We estimate a confidence interval for $\mathcal{B}_{\text{simple}}$ in two steps. First, we estimate 95% confidence intervals for \mathcal{S} and $\|\mathcal{G}\|^2$, assuming the data are exponentially² and normally distributed, respectfully, based

²For the exponential distribution, we use approximate confidence interval under ‘‘Confidence Intervals’’ here: https://en.wikipedia.org/wiki/Exponential_distribution.

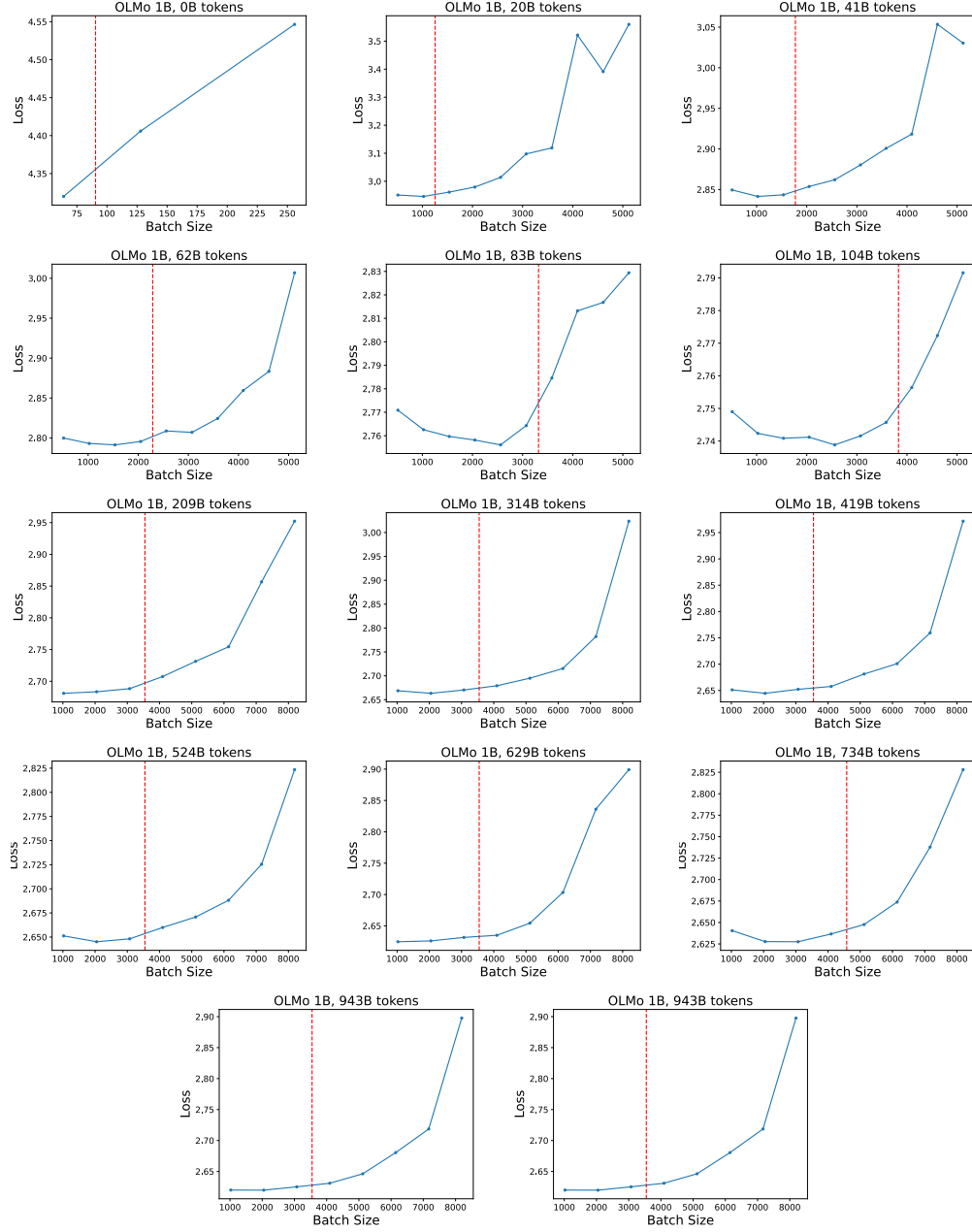
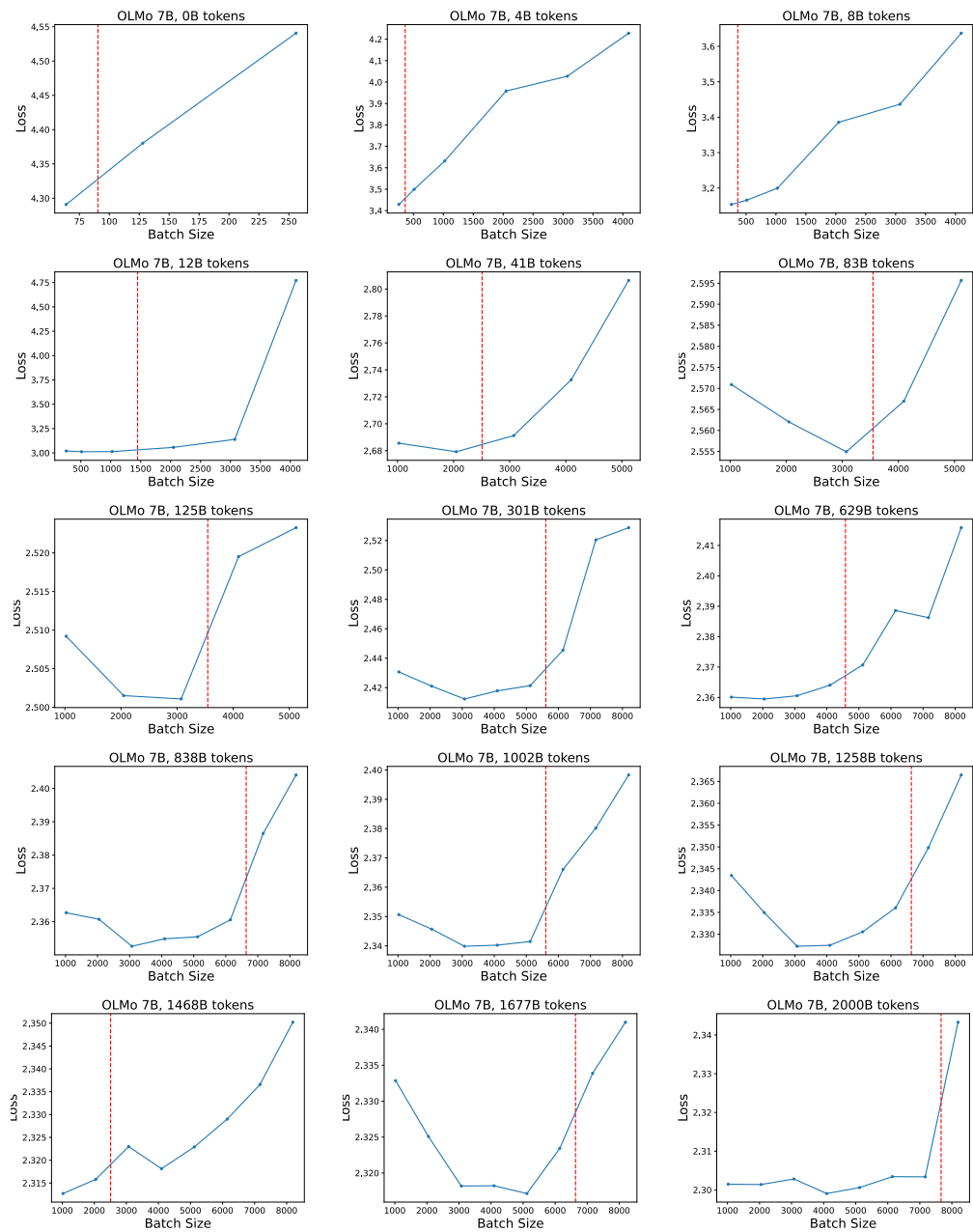


Figure 5: All loss vs. batch size plots for OLMo 1B. Overall, the red line moves to the right over time, showing that the CBS increases.



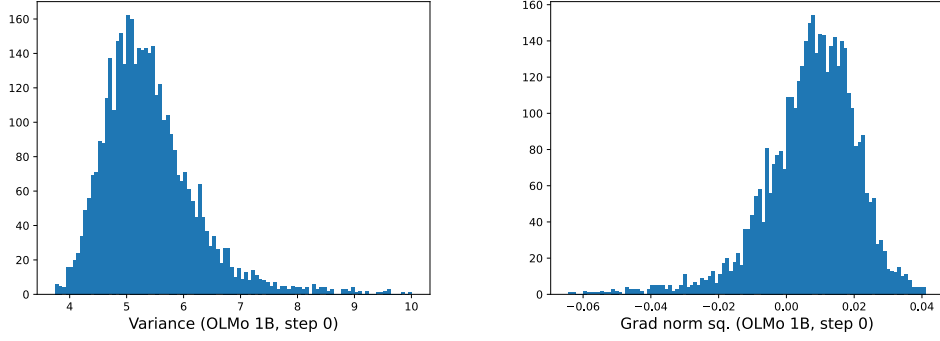


Figure 6: Representative histograms for \mathcal{S} and $\|\mathcal{G}\|^2$, showing data from the 1st to 99th percentiles. The distribution for \mathcal{S} is positive, leading us to use an exponential distribution, while the fact that some samples of $\|\mathcal{G}\|^2$ are negative motivates a normal distribution.

on manual inspection of their distributions (cf. Figure 6). We denote these intervals $[a_{\mathcal{S}}, b_{\mathcal{S}}]$ and $[a_{\|\mathcal{G}\|^2}, b_{\|\mathcal{G}\|^2}]$, respectively. We then define the confidence interval for $\mathcal{B}_{\text{simple}}$ as follows:

$$\left[\frac{a_{\mathcal{S}}}{b_{\|\mathcal{G}\|^2}}, \frac{b_{\mathcal{S}}}{a_{\|\mathcal{G}\|^2}} \right].$$

If our estimates for \mathcal{S} or $\|\mathcal{G}\|^2$ (or their lower or upper bounds) come out negative, we consider them to be 0.

The checkpoints considered for OLMo 1B are steps 0, 10K, 20K, 40K, ..., 100K, 200K, ..., 400K. For OLMo 7B, we use checkpoints at steps 0, 10K, ..., 40K, 60K, 70K, ..., 100K, 200K, ..., 400K. The noise scale experiment for each checkpoint (for both the 1B and 7B models) was launched on a single GPU.

C License Information

The OLMo models (Groeneveld et al., 2024; OLMo et al., 2025) and pretraining code, which we use, are released under Apache-2.0 license. C4 (Dodge et al., 2021) is released under ODC-BY license. The Pile (Gao et al., 2020) is released under MIT license.

D Deriving CBS Scaling Laws: An Attempt

In this section, we explore whether our empirical fits for the critical batch size over training can be used to derive scaling laws for aggregate critical batch size that have been derived in prior work. These scaling laws assume we want to use a fixed batch size B over training, and then train many different models to the same target loss. They then measure the critical B^* up to which increases in batch size do not diminish token efficiency. The standard finding from such work is that CBS grows $\propto \sqrt{T}$, where T is the total training budget in tokens. This is consistent with our finding that CBS increases over the course of training—moreover, we now seek to analyze whether this scaling law can be derived from our empirical measurements of local CBS. If so, this would provide converging evidence and a simpler method for fitting CBS scaling laws that only requires training a single model.

To begin, we assume that the goal of picking a fixed batch size B is to minimize the L2 distance to the local CBS over the course of training. It is not obvious that minimizing L2 distance is the right way to pick the fixed CBS: for instance, we might want to weight training at a batch size *above* the local CBS more negatively than training below it. Regardless, we will proceed for now under the assumption that this is the right perspective. We also make the weaker assumption that $f(t) = 0$, in line with our empirical findings (Section 3). It follows that the best batch size to train at (i.e., fixed CBS) is simply the average local CBS over training:

798 **Proposition 1.** Let $f(t)$ be integrable with $f(0) = 0$ and define

$$R_2 = \sqrt{\int_0^T (B - f(t))^2 dt}.$$

799 Then R_2 is minimized by $B^* = \frac{1}{T} \int_0^T f(t) dt$.

800 *Proof.* We can first simplify the expression for $(R_2)^2$:

$$\begin{aligned} (R_2)^2 &= \int_0^T (B - f(t))^2 dt \\ &= \int_0^T (B^2 - 2Bf(t) + f(t)^2) dt \\ &= B^2T - \int_0^T (2Bf(t) - f(t)^2) dt. \end{aligned}$$

801 Now, taking the derivative with respect to B , we get

$$\frac{d}{dB} (R_2)^2 = 2BT - 2 \int_0^T f(t) dt.$$

802 Note that the second derivative $2T$ is positive. Thus, setting the derivative to 0 and solving for B , we
803 conclude that the following value of B minimizes R_2 :

$$B = \frac{1}{T} \int_0^T f(t) dt. \quad \square$$

804 Thus, under the assumptions we have made, if we are trying to pick a fixed batch size that best
805 approximates the local CBS throughout training, we can simply pick the average CBS over training.
806 We can use Proposition 1 to derive a scaling law for the fixed B^* as a function of the final CBS or,
807 equivalently, the total steps T . We now consider various reasonable functional forms $f(t)$ for the
808 CBS.

809 **D.1 Power Law CBS Scaling**

810 We first consider the prediction for the fixed CBS scaling law if the local CBS evolves as a power law.

811 **Proposition 2** (B^* for power-law CBS). Let $f(t) = t^c$ for $c > 0$. Then the fixed CBS is

$$B^* = \frac{1}{c+1} T^c.$$

812 *Proof.* Plug in and solve the integral:

$$\begin{aligned} B &= \frac{1}{T} \int_0^T t^c dt \\ &= \frac{1}{T} \cdot \left[\frac{t^{c+1}}{c+1} \right]_0^T \\ &= \frac{T^c}{c+1}. \end{aligned} \quad \square$$

813 In the case where $c = 1/2$ (square root), $B^* = \frac{2}{3} B_T^* = \frac{2}{3} \sqrt{T}$, which derives the \sqrt{T} scaling law
814 proposed by prior work (Zhang et al., 2024).

task	split	# shots	reference
ARC-Challenge	Test	5	(Clark et al., 2018)
ARC-Easy	Test	5	(Clark et al., 2018)
CommonsenseQA	Val	5	(Talmor et al., 2019)
HellaSwag	Val	5	(Zellers et al., 2019)
MMLU	Val and Test	5	(Hendrycks et al., 2021)
PIQA	Val	5	(Bisk et al., 2020)
Social IQa	Val	5	(Sap et al., 2019)
WinoGrande	Val	5	(Sakaguchi et al., 2020)
GSM8K	Gold	5	(Cobbe et al., 2021)
Minerva	Gold	0	(Lewkowycz et al., 2022)
Humaneval	Gold	0	(Chen et al., 2021)
MBPP	Gold	0	(Austin et al., 2021)
Copycolors 10-way		0	(Wiegrefe et al., 2024)

815 D.2 Logarithmic CBS Scaling

816 **Proposition 3** (B^* for log CBS). *Let $f(t) = \log(t + 1)$. Then the fixed CBS is*

$$B^* = \frac{T}{T+1} \log(T+1) - 1.$$

817 *Proof.* Plug in and solve the integral:

$$\begin{aligned}
B &= \frac{1}{T} \int_0^T \log(t+1) dt \\
&= \frac{1}{T} \cdot \left[((t+1) \log(t+1) - t) \right]_0^T \\
&= \frac{T}{T+1} \log(T+1) - 1. \quad \square
\end{aligned}$$

818 Thus, for large T , the fixed CBS will scale as $B^* \approx \log T$.

819 D.3 Discussion

820 These results show that, if we are choosing the fixed batch size to minimize average distance to the
821 CBS as it evolves over training, we should pick it, more or less, as a simple function that slightly
822 discounts the final CBS. Specifically, if we believe that the local CBS grows as \sqrt{T} during training,
823 then this derives the \sqrt{T} scaling law for B^* proposed in prior work.

824 One limitation of this view is that the L2 residuals may not be the right way to measure closeness
825 to the CBS. In particular, it may be worse to overestimate the CBS compared to underestimate, as
826 training above the CBS (with a scaled up learning rate) can be unstable. We thus do not read too much
827 into this analysis, but view it as a potentially useful starting point for future empirical and theoretical
828 that derives CBS scaling laws from the development of the local CBS over training.

829 E BPB Evaluation on Downstream Tasks

830 This section lists the datasets we used to compute BPB measures for downstream tasks. For multiple-
831 choice tasks, we use the Cloze/Completion Formulation (CF), and compute the BPB metric on the
832 gold answer. For completion tasks, we simply compute BPB over the correct answer. This approach
833 was inspired by Bhagia et al. (2024). The selection of tasks follows the guidelines from Magnusson
834 et al. (2025).