

# A tour of data, eval and scaling language models with **Olmo**

David Heineman



At Ai2, we build  
fully open LM recipes

 OLMo

To facilitate research and accelerate the science of LMs, we need language models that are fully open.



# OLMo is being used to...

## Attribute model behavior to pretraining data

## Inform policy by studying LM development ecosystem

## Understand the effects of fine-tuning a pretrained model

### LESS: Selecting Influential Data for Targeted Instruction Tuning

Mingzhou Xia<sup>1</sup>, Sathika Malladi<sup>1</sup>, Sachin Gururanga<sup>1</sup>, Sameer Arora<sup>1</sup>, Danqi Chen<sup>1</sup>

#### Abstract

Instruction tuning has unlocked powerful capabilities in large language models (LLMs) using combinatorial datasets to develop general purpose chatbots. However, real-world applications often require a specialized suite of skills (e.g., reasoning). The challenge lies in identifying the most relevant data from these extensive datasets to effectively develop specific capabilities, a setting we frame as targeted instruction tuning. We propose LESS, an optimizer-aware and practically efficient algorithm to estimate data influence and perform low-rank gradient similarity search for instruction data selection. Crucially, LESS adapts existing influence formulations to work with the Adam optimizer and variable-length instruction data. LESS first constructs a highly readable and transferable *profiler* discounter with low-dimensional gradient features and then selects examples based on their similarity to few-shot examples embodying a specific capability. Experiments show that training on a LESS-selected 5% of the data can outperform full-dataset training on the full dataset across diverse downstream tasks. Furthermore, the selected data is highly transferable: smaller models can be leveraged to select useful data for larger models and models from different families. Our qualitative analysis shows that our method goes beyond surface form cues to identify data that encodes the necessary reasoning skills for the downstream application. To facilitate future work, we release code and data to the public at <https://github.com/least-llm/less>.

#### 1. Introduction

Instruction tuning has made large language models (LLMs) adapt at following human instructions (Ouyang et al., 2022) as versatile helpers (OpenAI, 2022, 2023; Anthropic, 2023; Google, 2023). Recent efforts curating highly diverse and wide-ranging instruction tuning datasets (Touretzky et al., 2023; Wang et al., 2023b; Yu et al., 2023; Xu et al., 2023) mirror today’s broader societal needs, generalization even from a small number of examples (Zhou et al., 2023). Regardless, it remains an open problem to understand how to best utilize these vast datasets. Many real-world applications call for cultivating a specific suite of capabilities in LLMs (e.g., reasoning, skills). However, training LLMs with mixed instruction tuning datasets can hinder the development of these specific capabilities. For example, Wang et al. (2023b) demonstrates that LLMs trained on a mix of instruction tuning datasets exhibit worse performance than those trained on a subset of the data. Additionally, considering the broad spectrum of user queries and the multitude of skills required to respond to them, there may not always be enough in-domain data available. Therefore, we hope to be able to effectively use the general instruction tuning data to improve specific capabilities. We frame this setting as *targeted instruction tuning*.

Given just a handful of examples embodying a specific capability, how can we effectively select relevant few-shot training data from a large collection of instruction datasets?

We address this problem by prioritizing training on data that directly minimizes loss on a target task instead of relying on surface form features (Gururanga et al., 2020; Xu et al., 2023b). Inspired by past work estimating the influence of individual training datapoints with gradient information (Pruthi et al., 2020; Han et al., 2023), we design an optimizer-aware approach to select such data. However, straightforward application of this influence formulation faces several challenges unique to instruction tuning: (1) LLMs are traditionally fine-tuned with the Adam optimizer (Kingma & Ba, 2015) instead of the canonical SGD optimizer. (2) Sparse expert-level gradients, variable-length instructions data and neural influence estimation, and (3) the large number of trainable parameters in LLMs make the computation and storage of gradient information extremely resource-intensive.

### Consent in Crisis: The Rapid Decline of the AI Data Commons

Shayne Longpre<sup>1</sup>, Robert Mahur<sup>1</sup>, Arif Le<sup>1</sup>, Campbell Lane<sup>1</sup>, Hamish Özdemirler<sup>1</sup>, William Brarner<sup>1</sup>, Naman Garg<sup>1</sup>, Naman Dhoti<sup>1</sup>, Martin Taha<sup>1</sup>, Sreyas Chakraborty<sup>1</sup>, Colin Rafferty<sup>1</sup>, Kevin Klyman<sup>1</sup>, Christopher Klamm<sup>1</sup>, Hattie Schofield<sup>1</sup>, Nikhil Singh<sup>1</sup>, Masoud Cheryx<sup>1</sup>, Ahmad Mhanna<sup>1</sup>, Adil Alami<sup>1</sup>, Candace Chinigo<sup>1</sup>, Du Han<sup>1</sup>, Shashank Sela<sup>1</sup>, Devika Mittal<sup>1</sup>, Dhanu Mirka<sup>1</sup>, Ehsan Alghamdi<sup>1</sup>, Ericor Shipilov<sup>1</sup>, Jianqiao Zhang<sup>1</sup>, Joanna Materzynska<sup>1</sup>, Kun Qian<sup>1</sup>, Kash Tey<sup>1</sup>, Lester Mackey<sup>1</sup>, Manan Dey<sup>1</sup>, Minsu Jung<sup>1</sup>, Muhammad Hammad<sup>1</sup>, Niklas Muennighoff<sup>1</sup>, Sanyam Prasad<sup>1</sup>, Sanyam Vasishta<sup>1</sup>, Shreya Mohapatra<sup>1</sup>, Vipul Gupta<sup>1</sup>, Vivek Shrivastava<sup>1</sup>, Yu-Min Chen<sup>1</sup>, Xuhui Zhou<sup>1</sup>, Yuhui Li<sup>1</sup>, Gaining Yong<sup>1</sup>, Lank Vilas<sup>1</sup>, Stella Biderman<sup>1</sup>, Haimin Li<sup>1</sup>, Daphne Ippolito<sup>1</sup>, Sara Hooker<sup>1</sup>, Jack Karbauer<sup>1</sup>, and Sandy Tsai<sup>1</sup>

<sup>1</sup>Team Lead, <sup>2</sup>Top Contributor, <sup>3</sup>Contributor (alphabetically), <sup>4</sup>Advisors

#### Abstract

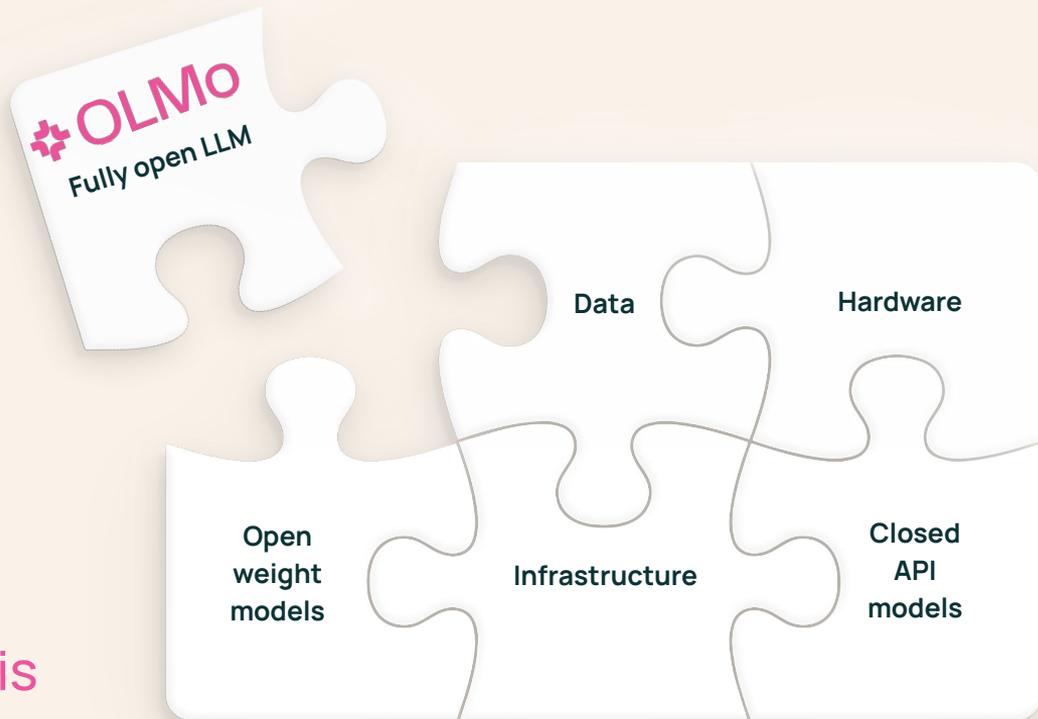
General purpose artificial intelligence (AI) systems are built on massive swaths of public web data, assembled into corpora such as Common Crawl, and indexed. To our knowledge, we conduct the first, large-scale, longitudinal audit of the content provided for the web domains underpinning AI training corpora. Our audit of 16,000 web domains provides an expansive view of crawlable web data and how collected data use performance an absolute one-time. We observe a consolidation of AI-

### Weak-to-Strong Extrapolation Expedites Alignment

Chujie Zhang<sup>1</sup>, Zhi Wang<sup>1</sup>, Hong Ji<sup>1</sup>, Miao Huang<sup>1</sup>, Nanyang Peng<sup>1</sup>  
<sup>1</sup>The Ohio State University, <sup>2</sup>UC Berkeley, <sup>3</sup>Microsoft, <sup>4</sup>Google, <sup>5</sup>Meta, <sup>6</sup>OpenAI, <sup>7</sup>Google DeepMind, <sup>8</sup>Meta AI, <sup>9</sup>Google Research, <sup>10</sup>Google Research, <sup>11</sup>Google Research, <sup>12</sup>Google Research, <sup>13</sup>Google Research, <sup>14</sup>Google Research, <sup>15</sup>Google Research, <sup>16</sup>Google Research, <sup>17</sup>Google Research, <sup>18</sup>Google Research, <sup>19</sup>Google Research, <sup>20</sup>Google Research, <sup>21</sup>Google Research, <sup>22</sup>Google Research, <sup>23</sup>Google Research, <sup>24</sup>Google Research, <sup>25</sup>Google Research, <sup>26</sup>Google Research, <sup>27</sup>Google Research, <sup>28</sup>Google Research, <sup>29</sup>Google Research, <sup>30</sup>Google Research, <sup>31</sup>Google Research, <sup>32</sup>Google Research, <sup>33</sup>Google Research, <sup>34</sup>Google Research, <sup>35</sup>Google Research, <sup>36</sup>Google Research, <sup>37</sup>Google Research, <sup>38</sup>Google Research, <sup>39</sup>Google Research, <sup>40</sup>Google Research, <sup>41</sup>Google Research, <sup>42</sup>Google Research, <sup>43</sup>Google Research, <sup>44</sup>Google Research, <sup>45</sup>Google Research, <sup>46</sup>Google Research, <sup>47</sup>Google Research, <sup>48</sup>Google Research, <sup>49</sup>Google Research, <sup>50</sup>Google Research, <sup>51</sup>Google Research, <sup>52</sup>Google Research, <sup>53</sup>Google Research, <sup>54</sup>Google Research, <sup>55</sup>Google Research, <sup>56</sup>Google Research, <sup>57</sup>Google Research, <sup>58</sup>Google Research, <sup>59</sup>Google Research, <sup>60</sup>Google Research, <sup>61</sup>Google Research, <sup>62</sup>Google Research, <sup>63</sup>Google Research, <sup>64</sup>Google Research, <sup>65</sup>Google Research, <sup>66</sup>Google Research, <sup>67</sup>Google Research, <sup>68</sup>Google Research, <sup>69</sup>Google Research, <sup>70</sup>Google Research, <sup>71</sup>Google Research, <sup>72</sup>Google Research, <sup>73</sup>Google Research, <sup>74</sup>Google Research, <sup>75</sup>Google Research, <sup>76</sup>Google Research, <sup>77</sup>Google Research, <sup>78</sup>Google Research, <sup>79</sup>Google Research, <sup>80</sup>Google Research, <sup>81</sup>Google Research, <sup>82</sup>Google Research, <sup>83</sup>Google Research, <sup>84</sup>Google Research, <sup>85</sup>Google Research, <sup>86</sup>Google Research, <sup>87</sup>Google Research, <sup>88</sup>Google Research, <sup>89</sup>Google Research, <sup>90</sup>Google Research, <sup>91</sup>Google Research, <sup>92</sup>Google Research, <sup>93</sup>Google Research, <sup>94</sup>Google Research, <sup>95</sup>Google Research, <sup>96</sup>Google Research, <sup>97</sup>Google Research, <sup>98</sup>Google Research, <sup>99</sup>Google Research, <sup>100</sup>Google Research, <sup>101</sup>Google Research, <sup>102</sup>Google Research, <sup>103</sup>Google Research, <sup>104</sup>Google Research, <sup>105</sup>Google Research, <sup>106</sup>Google Research, <sup>107</sup>Google Research, <sup>108</sup>Google Research, <sup>109</sup>Google Research, <sup>110</sup>Google Research, <sup>111</sup>Google Research, <sup>112</sup>Google Research, <sup>113</sup>Google Research, <sup>114</sup>Google Research, <sup>115</sup>Google Research, <sup>116</sup>Google Research, <sup>117</sup>Google Research, <sup>118</sup>Google Research, <sup>119</sup>Google Research, <sup>120</sup>Google Research, <sup>121</sup>Google Research, <sup>122</sup>Google Research, <sup>123</sup>Google Research, <sup>124</sup>Google Research, <sup>125</sup>Google Research, <sup>126</sup>Google Research, <sup>127</sup>Google Research, <sup>128</sup>Google Research, <sup>129</sup>Google Research, <sup>130</sup>Google Research, <sup>131</sup>Google Research, <sup>132</sup>Google Research, <sup>133</sup>Google Research, <sup>134</sup>Google Research, <sup>135</sup>Google Research, <sup>136</sup>Google Research, <sup>137</sup>Google Research, <sup>138</sup>Google Research, <sup>139</sup>Google Research, <sup>140</sup>Google Research, <sup>141</sup>Google Research, <sup>142</sup>Google Research, <sup>143</sup>Google Research, <sup>144</sup>Google Research, <sup>145</sup>Google Research, <sup>146</sup>Google Research, <sup>147</sup>Google Research, <sup>148</sup>Google Research, <sup>149</sup>Google Research, <sup>150</sup>Google Research, <sup>151</sup>Google Research, <sup>152</sup>Google Research, <sup>153</sup>Google Research, <sup>154</sup>Google Research, <sup>155</sup>Google Research, <sup>156</sup>Google Research, <sup>157</sup>Google Research, <sup>158</sup>Google Research, <sup>159</sup>Google Research, <sup>160</sup>Google Research, <sup>161</sup>Google Research, <sup>162</sup>Google Research, <sup>163</sup>Google Research, <sup>164</sup>Google Research, <sup>165</sup>Google Research, <sup>166</sup>Google Research, <sup>167</sup>Google Research, <sup>168</sup>Google Research, <sup>169</sup>Google Research, <sup>170</sup>Google Research, <sup>171</sup>Google Research, <sup>172</sup>Google Research, <sup>173</sup>Google Research, <sup>174</sup>Google Research, <sup>175</sup>Google Research, <sup>176</sup>Google Research, <sup>177</sup>Google Research, <sup>178</sup>Google Research, <sup>179</sup>Google Research, <sup>180</sup>Google Research, <sup>181</sup>Google Research, <sup>182</sup>Google Research, <sup>183</sup>Google Research, <sup>184</sup>Google Research, <sup>185</sup>Google Research, <sup>186</sup>Google Research, <sup>187</sup>Google Research, <sup>188</sup>Google Research, <sup>189</sup>Google Research, <sup>190</sup>Google Research, <sup>191</sup>Google Research, <sup>192</sup>Google Research, <sup>193</sup>Google Research, <sup>194</sup>Google Research, <sup>195</sup>Google Research, <sup>196</sup>Google Research, <sup>197</sup>Google Research, <sup>198</sup>Google Research, <sup>199</sup>Google Research, <sup>200</sup>Google Research, <sup>201</sup>Google Research, <sup>202</sup>Google Research, <sup>203</sup>Google Research, <sup>204</sup>Google Research, <sup>205</sup>Google Research, <sup>206</sup>Google Research, <sup>207</sup>Google Research, <sup>208</sup>Google Research, <sup>209</sup>Google Research, <sup>210</sup>Google Research, <sup>211</sup>Google Research, <sup>212</sup>Google Research, <sup>213</sup>Google Research, <sup>214</sup>Google Research, <sup>215</sup>Google Research, <sup>216</sup>Google Research, <sup>217</sup>Google Research, <sup>218</sup>Google Research, <sup>219</sup>Google Research, <sup>220</sup>Google Research, <sup>221</sup>Google Research, <sup>222</sup>Google Research, <sup>223</sup>Google Research, <sup>224</sup>Google Research, <sup>225</sup>Google Research, <sup>226</sup>Google Research, <sup>227</sup>Google Research, <sup>228</sup>Google Research, <sup>229</sup>Google Research, <sup>230</sup>Google Research, <sup>231</sup>Google Research, <sup>232</sup>Google Research, <sup>233</sup>Google Research, <sup>234</sup>Google Research, <sup>235</sup>Google Research, <sup>236</sup>Google Research, <sup>237</sup>Google Research, <sup>238</sup>Google Research, <sup>239</sup>Google Research, <sup>240</sup>Google Research, <sup>241</sup>Google Research, <sup>242</sup>Google Research, <sup>243</sup>Google Research, <sup>244</sup>Google Research, <sup>245</sup>Google Research, <sup>246</sup>Google Research, <sup>247</sup>Google Research, <sup>248</sup>Google Research, <sup>249</sup>Google Research, <sup>250</sup>Google Research, <sup>251</sup>Google Research, <sup>252</sup>Google Research, <sup>253</sup>Google Research, <sup>254</sup>Google Research, <sup>255</sup>Google Research, <sup>256</sup>Google Research, <sup>257</sup>Google Research, <sup>258</sup>Google Research, <sup>259</sup>Google Research, <sup>260</sup>Google Research, <sup>261</sup>Google Research, <sup>262</sup>Google Research, <sup>263</sup>Google Research, <sup>264</sup>Google Research, <sup>265</sup>Google Research, <sup>266</sup>Google Research, <sup>267</sup>Google Research, <sup>268</sup>Google Research, <sup>269</sup>Google Research, <sup>270</sup>Google Research, <sup>271</sup>Google Research, <sup>272</sup>Google Research, <sup>273</sup>Google Research, <sup>274</sup>Google Research, <sup>275</sup>Google Research, <sup>276</sup>Google Research, <sup>277</sup>Google Research, <sup>278</sup>Google Research, <sup>279</sup>Google Research, <sup>280</sup>Google Research, <sup>281</sup>Google Research, <sup>282</sup>Google Research, <sup>283</sup>Google Research, <sup>284</sup>Google Research, <sup>285</sup>Google Research, <sup>286</sup>Google Research, <sup>287</sup>Google Research, <sup>288</sup>Google Research, <sup>289</sup>Google Research, <sup>290</sup>Google Research, <sup>291</sup>Google Research, <sup>292</sup>Google Research, <sup>293</sup>Google Research, <sup>294</sup>Google Research, <sup>295</sup>Google Research, <sup>296</sup>Google Research, <sup>297</sup>Google Research, <sup>298</sup>Google Research, <sup>299</sup>Google Research, <sup>300</sup>Google Research, <sup>301</sup>Google Research, <sup>302</sup>Google Research, <sup>303</sup>Google Research, <sup>304</sup>Google Research, <sup>305</sup>Google Research, <sup>306</sup>Google Research, <sup>307</sup>Google Research, <sup>308</sup>Google Research, <sup>309</sup>Google Research, <sup>310</sup>Google Research, <sup>311</sup>Google Research, <sup>312</sup>Google Research, <sup>313</sup>Google Research, <sup>314</sup>Google Research, <sup>315</sup>Google Research, <sup>316</sup>Google Research, <sup>317</sup>Google Research, <sup>318</sup>Google Research, <sup>319</sup>Google Research, <sup>320</sup>Google Research, <sup>321</sup>Google Research, <sup>322</sup>Google Research, <sup>323</sup>Google Research, <sup>324</sup>Google Research, <sup>325</sup>Google Research, <sup>326</sup>Google Research, <sup>327</sup>Google Research, <sup>328</sup>Google Research, <sup>329</sup>Google Research, <sup>330</sup>Google Research, <sup>331</sup>Google Research, <sup>332</sup>Google Research, <sup>333</sup>Google Research, <sup>334</sup>Google Research, <sup>335</sup>Google Research, <sup>336</sup>Google Research, <sup>337</sup>Google Research, <sup>338</sup>Google Research, <sup>339</sup>Google Research, <sup>340</sup>Google Research, <sup>341</sup>Google Research, <sup>342</sup>Google Research, <sup>343</sup>Google Research, <sup>344</sup>Google Research, <sup>345</sup>Google Research, <sup>346</sup>Google Research, <sup>347</sup>Google Research, <sup>348</sup>Google Research, <sup>349</sup>Google Research, <sup>350</sup>Google Research, <sup>351</sup>Google Research, <sup>352</sup>Google Research, <sup>353</sup>Google Research, <sup>354</sup>Google Research, <sup>355</sup>Google Research, <sup>356</sup>Google Research, <sup>357</sup>Google Research, <sup>358</sup>Google Research, <sup>359</sup>Google Research, <sup>360</sup>Google Research, <sup>361</sup>Google Research, <sup>362</sup>Google Research, <sup>363</sup>Google Research, <sup>364</sup>Google Research, <sup>365</sup>Google Research, <sup>366</sup>Google Research, <sup>367</sup>Google Research, <sup>368</sup>Google Research, <sup>369</sup>Google Research, <sup>370</sup>Google Research, <sup>371</sup>Google Research, <sup>372</sup>Google Research, <sup>373</sup>Google Research, <sup>374</sup>Google Research, <sup>375</sup>Google Research, <sup>376</sup>Google Research, <sup>377</sup>Google Research, <sup>378</sup>Google Research, <sup>379</sup>Google Research, <sup>380</sup>Google Research, <sup>381</sup>Google Research, <sup>382</sup>Google Research, <sup>383</sup>Google Research, <sup>384</sup>Google Research, <sup>385</sup>Google Research, <sup>386</sup>Google Research, <sup>387</sup>Google Research, <sup>388</sup>Google Research, <sup>389</sup>Google Research, <sup>390</sup>Google Research, <sup>391</sup>Google Research, <sup>392</sup>Google Research, <sup>393</sup>Google Research, <sup>394</sup>Google Research, <sup>395</sup>Google Research, <sup>396</sup>Google Research, <sup>397</sup>Google Research, <sup>398</sup>Google Research, <sup>399</sup>Google Research, <sup>400</sup>Google Research, <sup>401</sup>Google Research, <sup>402</sup>Google Research, <sup>403</sup>Google Research, <sup>404</sup>Google Research, <sup>405</sup>Google Research, <sup>406</sup>Google Research, <sup>407</sup>Google Research, <sup>408</sup>Google Research, <sup>409</sup>Google Research, <sup>410</sup>Google Research, <sup>411</sup>Google Research, <sup>412</sup>Google Research, <sup>413</sup>Google Research, <sup>414</sup>Google Research, <sup>415</sup>Google Research, <sup>416</sup>Google Research, <sup>417</sup>Google Research, <sup>418</sup>Google Research, <sup>419</sup>Google Research, <sup>420</sup>Google Research, <sup>421</sup>Google Research, <sup>422</sup>Google Research, <sup>423</sup>Google Research, <sup>424</sup>Google Research, <sup>425</sup>Google Research, <sup>426</sup>Google Research, <sup>427</sup>Google Research, <sup>428</sup>Google Research, <sup>429</sup>Google Research, <sup>430</sup>Google Research, <sup>431</sup>Google Research, <sup>432</sup>Google Research, <sup>433</sup>Google Research, <sup>434</sup>Google Research, <sup>435</sup>Google Research, <sup>436</sup>Google Research, <sup>437</sup>Google Research, <sup>438</sup>Google Research, <sup>439</sup>Google Research, <sup>440</sup>Google Research, <sup>441</sup>Google Research, <sup>442</sup>Google Research, <sup>443</sup>Google Research, <sup>444</sup>Google Research, <sup>445</sup>Google Research, <sup>446</sup>Google Research, <sup>447</sup>Google Research, <sup>448</sup>Google Research, <sup>449</sup>Google Research, <sup>450</sup>Google Research, <sup>451</sup>Google Research, <sup>452</sup>Google Research, <sup>453</sup>Google Research, <sup>454</sup>Google Research, <sup>455</sup>Google Research, <sup>456</sup>Google Research, <sup>457</sup>Google Research, <sup>458</sup>Google Research, <sup>459</sup>Google Research, <sup>460</sup>Google Research, <sup>461</sup>Google Research, <sup>462</sup>Google Research, <sup>463</sup>Google Research, <sup>464</sup>Google Research, <sup>465</sup>Google Research, <sup>466</sup>Google Research, <sup>467</sup>Google Research, <sup>468</sup>Google Research, <sup>469</sup>Google Research, <sup>470</sup>Google Research, <sup>471</sup>Google Research, <sup>472</sup>Google Research, <sup>473</sup>Google Research, <sup>474</sup>Google Research, <sup>475</sup>Google Research, <sup>476</sup>Google Research, <sup>477</sup>Google Research, <sup>478</sup>Google Research, <sup>479</sup>Google Research, <sup>480</sup>Google Research, <sup>481</sup>Google Research, <sup>482</sup>Google Research, <sup>483</sup>Google Research, <sup>484</sup>Google Research, <sup>485</sup>Google Research, <sup>486</sup>Google Research, <sup>487</sup>Google Research, <sup>488</sup>Google Research, <sup>489</sup>Google Research, <sup>490</sup>Google Research, <sup>491</sup>Google Research, <sup>492</sup>Google Research, <sup>493</sup>Google Research, <sup>494</sup>Google Research, <sup>495</sup>Google Research, <sup>496</sup>Google Research, <sup>497</sup>Google Research, <sup>498</sup>Google Research, <sup>499</sup>Google Research, <sup>500</sup>Google Research, <sup>501</sup>Google Research, <sup>502</sup>Google Research, <sup>503</sup>Google Research, <sup>504</sup>Google Research, <sup>505</sup>Google Research, <sup>506</sup>Google Research, <sup>507</sup>Google Research, <sup>508</sup>Google Research, <sup>509</sup>Google Research, <sup>510</sup>Google Research, <sup>511</sup>Google Research, <sup>512</sup>Google Research, <sup>513</sup>Google Research, <sup>514</sup>Google Research, <sup>515</sup>Google Research, <sup>516</sup>Google Research, <sup>517</sup>Google Research, <sup>518</sup>Google Research, <sup>519</sup>Google Research, <sup>520</sup>Google Research, <sup>521</sup>Google Research, <sup>522</sup>Google Research, <sup>523</sup>Google Research, <sup>524</sup>Google Research, <sup>525</sup>Google Research, <sup>526</sup>Google Research, <sup>527</sup>Google Research, <sup>528</sup>Google Research, <sup>529</sup>Google Research, <sup>530</sup>Google Research, <sup>531</sup>Google Research, <sup>532</sup>Google Research, <sup>533</sup>Google Research, <sup>534</sup>Google Research, <sup>535</sup>Google Research, <sup>536</sup>Google Research, <sup>537</sup>Google Research, <sup>538</sup>Google Research, <sup>539</sup>Google Research, <sup>540</sup>Google Research, <sup>541</sup>Google Research, <sup>542</sup>Google Research, <sup>543</sup>Google Research, <sup>544</sup>Google Research, <sup>545</sup>Google Research, <sup>546</sup>Google Research, <sup>547</sup>Google Research, <sup>548</sup>Google Research, <sup>549</sup>Google Research, <sup>550</sup>Google Research, <sup>551</sup>Google Research, <sup>552</sup>Google Research, <sup>553</sup>Google Research, <sup>554</sup>Google Research, <sup>555</sup>Google Research, <sup>556</sup>Google Research, <sup>557</sup>Google Research, <sup>558</sup>Google Research, <sup>559</sup>Google Research, <sup>560</sup>Google Research, <sup>561</sup>Google Research, <sup>562</sup>Google Research, <sup>563</sup>Google Research, <sup>564</sup>Google Research, <sup>565</sup>Google Research, <sup>566</sup>Google Research, <sup>567</sup>Google Research, <sup>568</sup>Google Research, <sup>569</sup>Google Research, <sup>570</sup>Google Research, <sup>571</sup>Google Research, <sup>572</sup>Google Research, <sup>573</sup>Google Research, <sup>574</sup>Google Research, <sup>575</sup>Google Research, <sup>576</sup>Google Research, <sup>577</sup>Google Research, <sup>578</sup>Google Research, <sup>579</sup>Google Research, <sup>580</sup>Google Research, <sup>581</sup>Google Research, <sup>582</sup>Google Research, <sup>583</sup>Google Research, <sup>584</sup>Google Research, <sup>585</sup>Google Research, <sup>586</sup>Google Research, <sup>587</sup>Google Research, <sup>588</sup>Google Research, <sup>589</sup>Google Research, <sup>590</sup>Google Research, <sup>591</sup>Google Research, <sup>592</sup>Google Research, <sup>593</sup>Google Research, <sup>594</sup>Google Research, <sup>595</sup>Google Research, <sup>596</sup>Google Research, <sup>597</sup>Google Research, <sup>598</sup>Google Research, <sup>599</sup>Google Research, <sup>600</sup>Google Research, <sup>601</sup>Google Research, <sup>602</sup>Google Research, <sup>603</sup>Google Research, <sup>604</sup>Google Research, <sup>605</sup>Google Research, <sup>606</sup>Google Research, <sup>607</sup>Google Research, <sup>608</sup>Google Research, <sup>609</sup>Google Research, <sup>610</sup>Google Research, <sup>611</sup>Google Research, <sup>612</sup>Google Research, <sup>613</sup>Google Research, <sup>614</sup>Google Research, <sup>615</sup>Google Research, <sup>616</sup>Google Research, <sup>617</sup>Google Research, <sup>618</sup>Google Research, <sup>619</sup>Google Research, <sup>620</sup>Google Research, <sup>621</sup>Google Research, <sup>622</sup>Google Research, <sup>623</sup>Google Research, <sup>624</sup>Google Research, <sup>625</sup>Google Research, <sup>626</sup>Google Research, <sup>627</sup>Google Research, <sup>628</sup>Google Research, <sup>629</sup>Google Research, <sup>630</sup>Google Research, <sup>631</sup>Google Research, <sup>632</sup>Google Research, <sup>633</sup>Google Research, <sup>634</sup>Google Research, <sup>635</sup>Google Research, <sup>636</sup>Google Research, <sup>637</sup>Google Research, <sup>638</sup>Google Research, <sup>639</sup>Google Research, <sup>640</sup>Google Research, <sup>641</sup>Google Research, <sup>642</sup>Google Research, <sup>643</sup>Google Research, <sup>644</sup>Google Research, <sup>645</sup>Google Research, <sup>646</sup>Google Research, <sup>647</sup>Google Research, <sup>648</sup>Google Research, <sup>649</sup>Google Research, <sup>650</sup>Google Research, <sup>651</sup>Google Research, <sup>652</sup>Google Research, <sup>653</sup>Google Research, <sup>654</sup>Google Research, <sup>655</sup>Google Research, <sup>656</sup>Google Research, <sup>657</sup>Google Research, <sup>658</sup>Google Research, <sup>659</sup>Google Research, <sup>660</sup>Google Research, <sup>661</sup>Google Research, <sup>662</sup>Google Research, <sup>663</sup>Google Research, <sup>664</sup>Google Research, <sup>665</sup>Google Research, <sup>666</sup>Google Research, <sup>667</sup>Google Research, <sup>668</sup>Google Research, <sup>669</sup>Google Research, <sup>670</sup>Google Research, <sup>671</sup>Google Research, <sup>672</sup>Google Research, <sup>673</sup>Google Research, <sup>674</sup>Google Research, <sup>675</sup>Google Research, <sup>676</sup>Google Research, <sup>677</sup>Google Research, <sup>678</sup>Google Research, <sup>679</sup>Google Research, <sup>680</sup>Google Research, <sup>681</sup>Google Research, <sup>682</sup>Google Research, <sup>683</sup>Google Research, <sup>684</sup>Google Research, <sup>685</sup>Google Research, <sup>686</sup>Google Research, <sup>687</sup>Google Research, <sup>688</sup>Google Research, <sup>689</sup>Google Research, <sup>690</sup>Google Research, <sup>691</sup>Google Research, <sup>692</sup>Google Research, <sup>693</sup>Google Research, <sup>694</sup>Google Research, <sup>695</sup>Google Research, <sup>696</sup>Google Research, <sup>697</sup>Google Research, <sup>698</sup>Google Research, <sup>699</sup>Google Research, <sup>700</sup>Google Research, <sup>701</sup>Google Research, <sup>702</sup>Google Research, <sup>703</sup>Google Research, <sup>704</sup>Google Research, <sup>705</sup>Google Research, <sup>706</sup>Google Research, <sup>707</sup>Google Research, <sup>708</sup>Google Research, <sup>709</sup>Google Research, <sup>710</sup>Google Research, <sup>711</sup>Google Research, <sup>712</sup>Google Research, <sup>713</sup>Google Research, <sup>714</sup>Google Research, <sup>715</sup>Google Research, <sup>716</sup>Google Research, <sup>717</sup>Google Research, <sup>718</sup>Google Research, <sup>719</sup>Google Research, <sup>720</sup>Google Research, <sup>721</sup>Google Research, <sup>722</sup>Google Research, <sup>723</sup>Google Research, <sup>724</sup>Google Research, <sup>725</sup>Google Research, <sup>726</sup>Google Research, <sup>727</sup>Google Research, <sup>728</sup>Google Research, <sup>729</sup>Google Research, <sup>730</sup>Google Research, <sup>731</sup>Google Research, <sup>732</sup>Google Research, <sup>733</sup>Google Research, <sup>734</sup>Google Research, <sup>735</sup>Google Research, <sup>736</sup>Google Research, <sup>737</sup>Google Research, <sup>738</sup>Google Research, <sup>739</sup>Google Research, <sup>740</sup>Google Research, <sup>741</sup>Google Research, <sup>742</sup>Google Research, <sup>743</sup>Google Research, <sup>744</sup>Google Research, <sup>745</sup>Google Research, <sup>746</sup>Google Research, <sup>747</sup>Google Research, <sup>748</sup>Google Research, <sup>749</sup>Google Research, <sup>750</sup>Google Research, <sup>751</sup>Google Research, <sup>752</sup>Google Research, <sup>753</sup>Google Research, <sup>754</sup>Google Research, <sup>755</sup>Google Research, <sup>756</sup>Google Research, <sup>757</sup>Google Research, <sup>758</sup>Google Research, <sup>759</sup>Google Research, <sup>760</sup>Google Research, <sup>761</sup>Google Research, <sup>762</sup>Google Research, <sup>763</sup>Google Research, <sup>764</sup>Google Research, <sup>765</sup>Google Research, <sup>766</sup>Google Research, <sup>767</sup>Google Research, <sup>768</sup>Google Research, <sup>769</sup>Google Research, <sup>770</sup>Google Research, <sup>771</sup>Google Research, <sup>772</sup>Google Research, <sup>773</sup>Google Research, <sup>774</sup>Google Research, <sup>775</sup>Google Research, <sup>776</sup>Google Research, <sup>777</sup>Google Research, <sup>778</sup>Google Research, <sup>779</sup>Google Research, <sup>780</sup>Google Research, <sup>781</sup>Google Research, <sup>782</sup>Google Research, <sup>783</sup>Google Research, <sup>784</sup>Google Research, <sup>785</sup>Google Research, <sup>786</sup>Google Research, <sup>787</sup>Google Research, <sup>788</sup>Google Research, <sup>789</sup>Google Research, <sup>790</sup>Google Research, <sup>791</sup>Google Research, <sup>792</sup>Google Research, <sup>793</sup>Google Research, <sup>794</sup>Google Research, <sup>795</sup>Google Research, <sup>796</sup>Google Research, <sup>797</sup>Google Research, <sup>798</sup>Google Research, <sup>799</sup>Google Research, <sup>800</sup>Google Research, <sup>801</sup>Google Research, <sup>802</sup>Google Research, <sup>803</sup>Google Research, <sup>804</sup>Google Research, <sup>805</sup>Google Research, <sup>806</sup>Google Research, <sup>807</sup>Google Research, <sup>808</sup>Google Research, <sup>809</sup>Google Research, <sup>810</sup>Google Research, <sup>811</sup>Google Research, <sup>812</sup>Google Research, <sup>813</sup>Google Research, <sup>814</sup>Google Research, <sup>815</sup>Google Research, <sup>816</sup>Google Research, <sup>817</sup>Google Research, <sup>818</sup>Google Research, <sup>819</sup>Google Research, <sup>820</sup>Google Research, <sup>821</sup>Google Research, <sup>822</sup>Google Research, <sup>823</sup>Google Research, <sup>824</sup>Google Research, <sup>825</sup>Google Research, <sup>826</sup>Google Research, <sup>827</sup>Google Research, <sup>828</sup>Google Research, <sup>829</sup>Google Research, <sup>830</sup>Google Research, <sup>831</sup>Google Research, <sup>832</sup>Google Research, <sup>833</sup>Google Research, <sup>834</sup>Google Research, <sup>835</sup>Google Research, <sup>836</sup>Google Research, <sup>837</sup>Google Research, <sup>838</sup>Google Research, <sup>839</sup>Google Research, <sup>840</sup>Google Research, <sup>841</sup>Google Research, <sup>842</sup>Google Research, <sup>843</sup>

# What does fully open mean?

- Model **weights**
- Intermediate **checkpoints**
- Detailed **recipes**
- All the **data**
- **Code** to reproduce
- **Documentation** and **analysis**



# OLMo is being used to...

## Develop new architectures

NousResearch **OLMo-Bitnet-1B** like 115

Text Generation Transformers PyTorch allenai/dolma o1mo custom

License: apache-2.0

Model card

**LLaVaOLMoBitnet1B: Ternary LLM goes Multimodal!**

Jainaveen Sundaram, Ravi Iyer  
(jainaveen.sundaram, ravi@hankai.lyer)@intel.com

**OLMo-Bitnet**

OLMo-Bitnet-1B method description

**Models are in 1**

It was trained on merely a research

As separate training hyperparameters comparison ca

### Zamba: A Compact 7B SSM Hybrid Model

Paolo Glorioso Quentin Anthony Yury Tokpanov James Whittington Jonathan Pflaht  
Adam Ibrahim Beren Millidge  
(paolo, qantam, yury, james, jonathan, adam, beren)@zephyr.com

Zyhra  
Palo Alto, CA

### SOAP: IMPROVING AND STABILIZING SHAMPOO USING ADAM

Nikhil Vyas\*  
Harvard University

Deven Morwani\*  
Harvard University

Rosie Zhao<sup>1</sup>  
Harvard University

Itai Shapira<sup>1</sup>  
Harvard University

David Brandfonbrener  
Kempner Institute at Harvard University

Lucas Janson  
Harvard University

Sham Kakade  
Kempner Institute at Harvard University

#### ABSTRACT

There is growing evidence of the effectiveness of Shampoo, a higher-order preconditioning method, over Adam in deep learning optimization tasks. However, Shampoo's drawbacks include additional hyperparameters and computational overhead when compared to Adam, which only updates running averages of first- and second-moment quantities. This work establishes a formal connection between Shampoo (implemented with the  $1/2$  power) and Adafactor — a memory-efficient approximation of Adam — showing that Shampoo is equivalent to running Adafactor in the eigenbasis of Shampoo's preconditioner. This insight leads to the design of a simpler and computationally efficient algorithm: Shampoo<sup>2</sup> with Adam in the Preconditioner's eigenbasis (SOAP). With regards to improving Shampoo's computational efficiency, the most straightforward approach would be to simply compute Shampoo's eigendecomposition less frequently. Unfortunately, as our empirical results show, this leads to performance degradation that worsens with this frequency. SOAP mitigates this degradation by continually updating the running average of the second moment, just as Adam does, but in the current (slowly changing) coordinate basis. Furthermore, since SOAP is equivalent to running Adam in a rotated space, it introduces only one additional hyperparameter (the preconditioning frequency) compared to Adam. We empirically evaluate SOAP on language model pre-training with 360m and 660m sized models. In the large batch regime, SOAP reduces the number of iterations by over 40% and wall clock time by over 35% compared to Adam<sup>2</sup>, with approximately 20% improvements in both metrics compared to Shampoo. An implementation of SOAP is available at <https://github.com/nikhilvyas/soap>.

## Study how models acquire/forget knowledge during training

### Adaptive Pre-training Data Detection for Large Language Models via Surprising Tokens

Angi Zhang

Anwesa, Assemblé, Acee  
Understanding How Transformers Answer Multiple Choice Questions

Sarah Wiegrefe<sup>1,2\*</sup> Oyvind Tafjord<sup>1</sup> Yonatan Belinkov<sup>1</sup>  
Hannaneh Hajishiraf<sup>1,2\*</sup> Ashish Sabharwal<sup>2</sup>

<sup>1</sup>Allen Institute for AI, <sup>2</sup>University of Washington, <sup>3</sup>Technion  
wiegrefe@arizona@gmail.com

#### Abstract

Multiple-choice question answering is a key component of performance in language models that is tested by its benchmarks. However, recent evidence that models can have quite a strong bias, particularly when the task is verified slightly (such as by a shell choice order). In this work we ask if careful models perform formally to employ vocabulary projection and patching methods to localize key tokens that encode relevant information for the correct answer. We find that pre-training a specific answer symbol in  $v$  space to a single middle layer, and spec multi-head self-attention mechanism that subsequent layers increase the of the predicted answer symbol in  $v$  space, and that this probability is associated with a sparse set of attention unique nodes. We additionally used synthetic tasks on downstream question to pinpoint when a model has learned MQA, and show that we can separate answer symbol tokens in  $v$  space as a property of models unable to format MQA tasks.

#### Generalization vs. Memorization: Tracing Language Models' Capabilities Back to Pretraining Data

Antonis Antonides<sup>1</sup> Xinyi Wang<sup>2</sup>

#### Abstract

Despite the proven utility of large language models (LLMs) in real-world applications, a lack of understanding regarding how they leverage their large-scale pretraining to achieve such capabilities. In this investigation the interplay between generalization and memorization in pretraining LLM training data. Our experiments focus on general task types: translation, question and multiple-choice reasoning. With varying degrees of generality, we investigate performance, leading to the hypothesis that LLMs' ability to learn from a delicate balance of memorization and generalization with sufficient task training data, and point the way to analyses that could further improve understanding of these models.

### How Do Large Language Models Acquire Factual Knowledge During Pretraining?

Haoyan Chang<sup>1</sup> Jiahao Park<sup>1</sup> Seunghyeon Yee<sup>1</sup> Seokyeung Yoo<sup>2</sup>

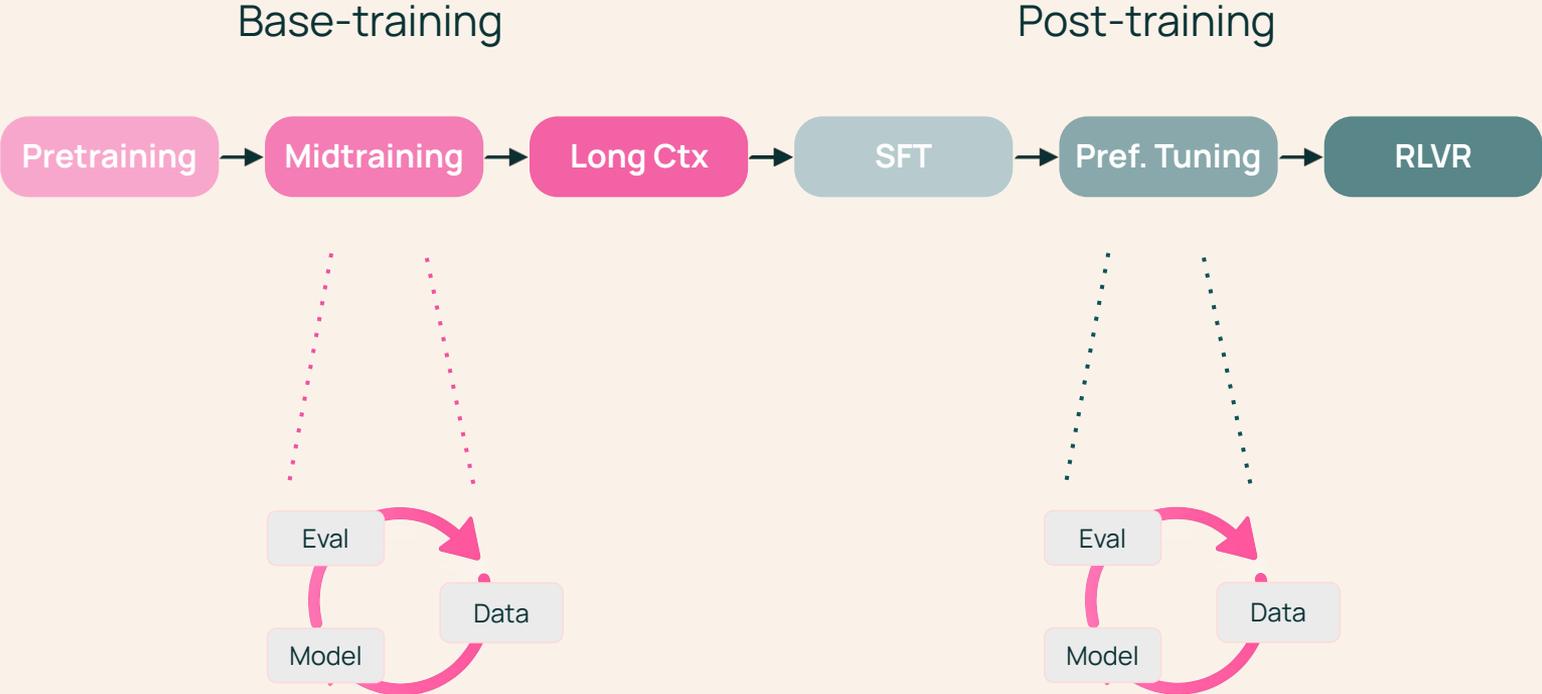
#### Detection and Measurement of Syntactic Templates in Generated Text

Chantall Shah<sup>1</sup> Yanai Elazar<sup>1,3</sup> Junyi Jessy Li<sup>1</sup> Byron C. Wallace<sup>1</sup>  
<sup>1</sup>Northwestern University, <sup>2</sup>Allen Institute for AI, <sup>3</sup>University of Washington, <sup>4</sup>The University of Texas at Austin  
(shah.c, wallace@northwestern.edu)

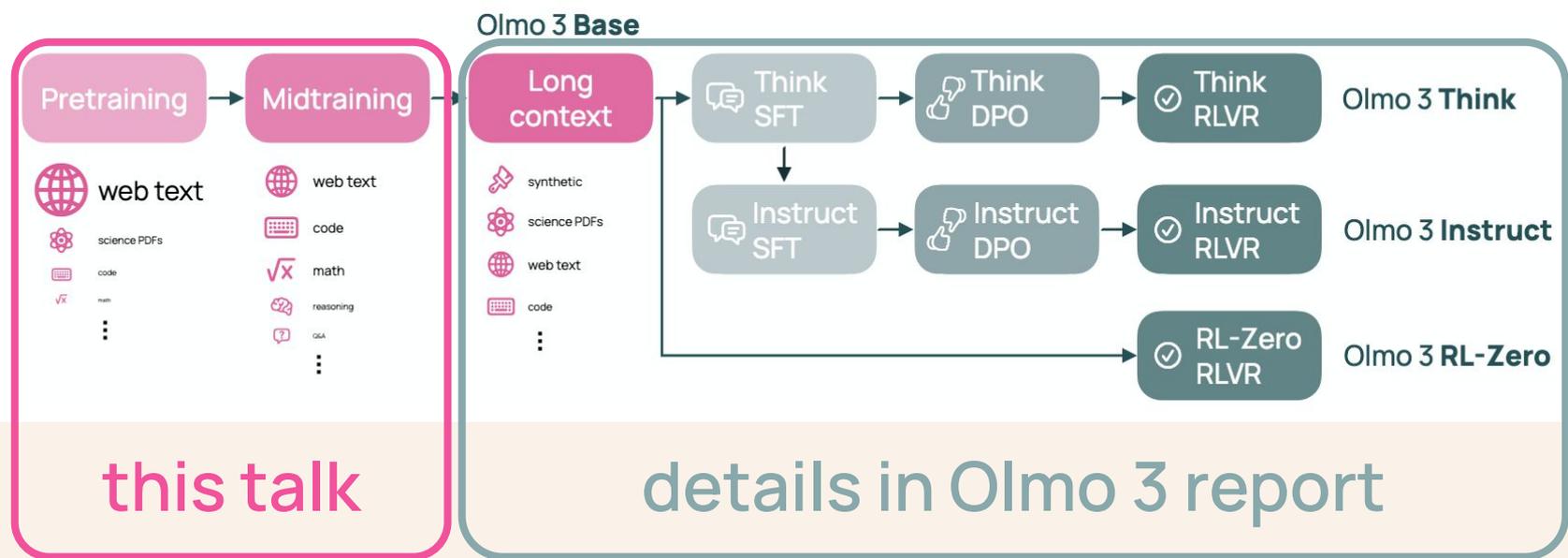
#### Abstract

Recent work on evaluating the diversity of text generated by LLMs has focused on word-level features. Here we offer an analysis of syntactic features that characterize general repetition in models, beyond frequent n-grams. Specifically, we define syntactic templates and show that models tend to produce templated text in downstream tasks at a higher rate than what is found in human-reference texts. We find that most (76%) templates in model-generated text can be found in pre-training data (compared to only 35% of human-authored text), and are not over-represented during fine-tuning processes such as RLHF. This connection to pre-training data allows us to analyze syntactic templates in models where we do not have the pre-training data. We also find that templates as features are able to differentiate between models, tasks, and domains, and are useful for qualitatively evaluating common model constructions. Finally, we demonstrate the use of templates as a useful tool for analyzing style memorization of training data in LLMs.

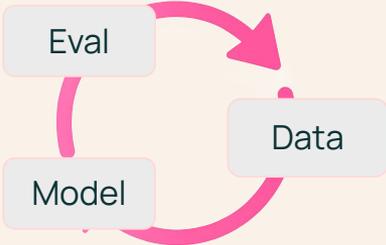
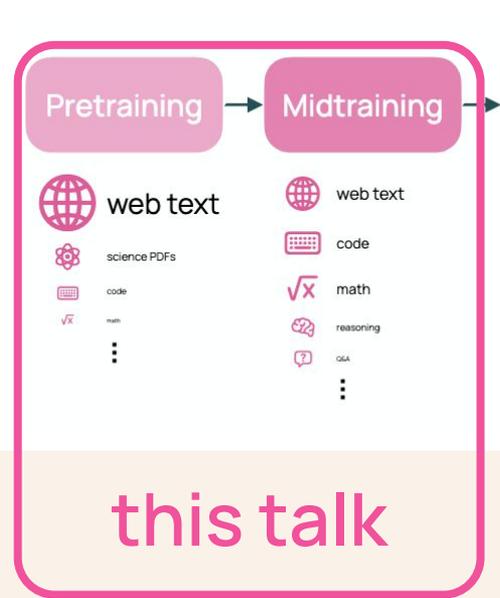
# Olmo 3 Model Flow



# Olmo 3 Model Flow

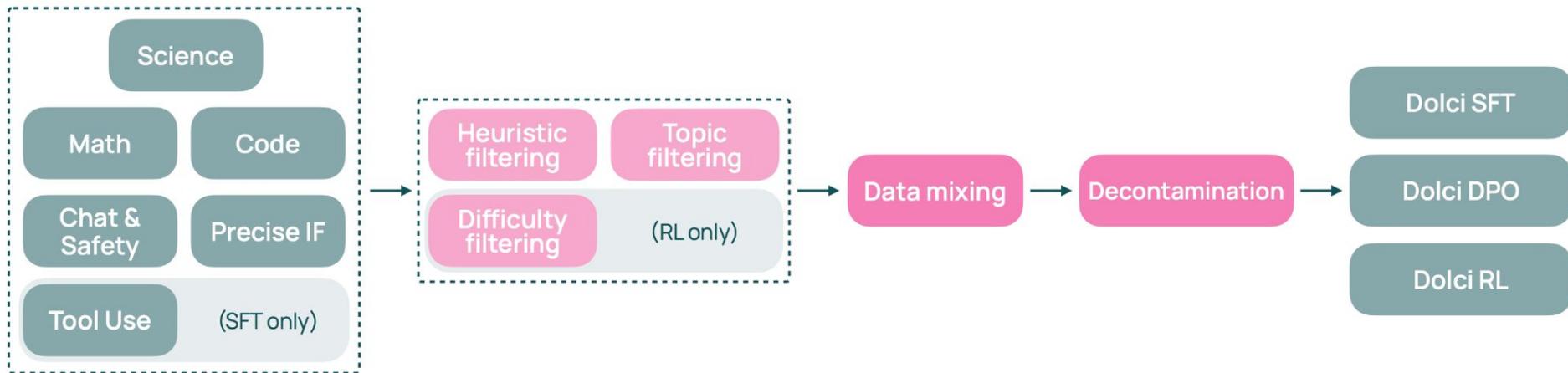


# Olmo 3 Model Flow





# Posttrain for Olmo 3 Think



## SFT Data

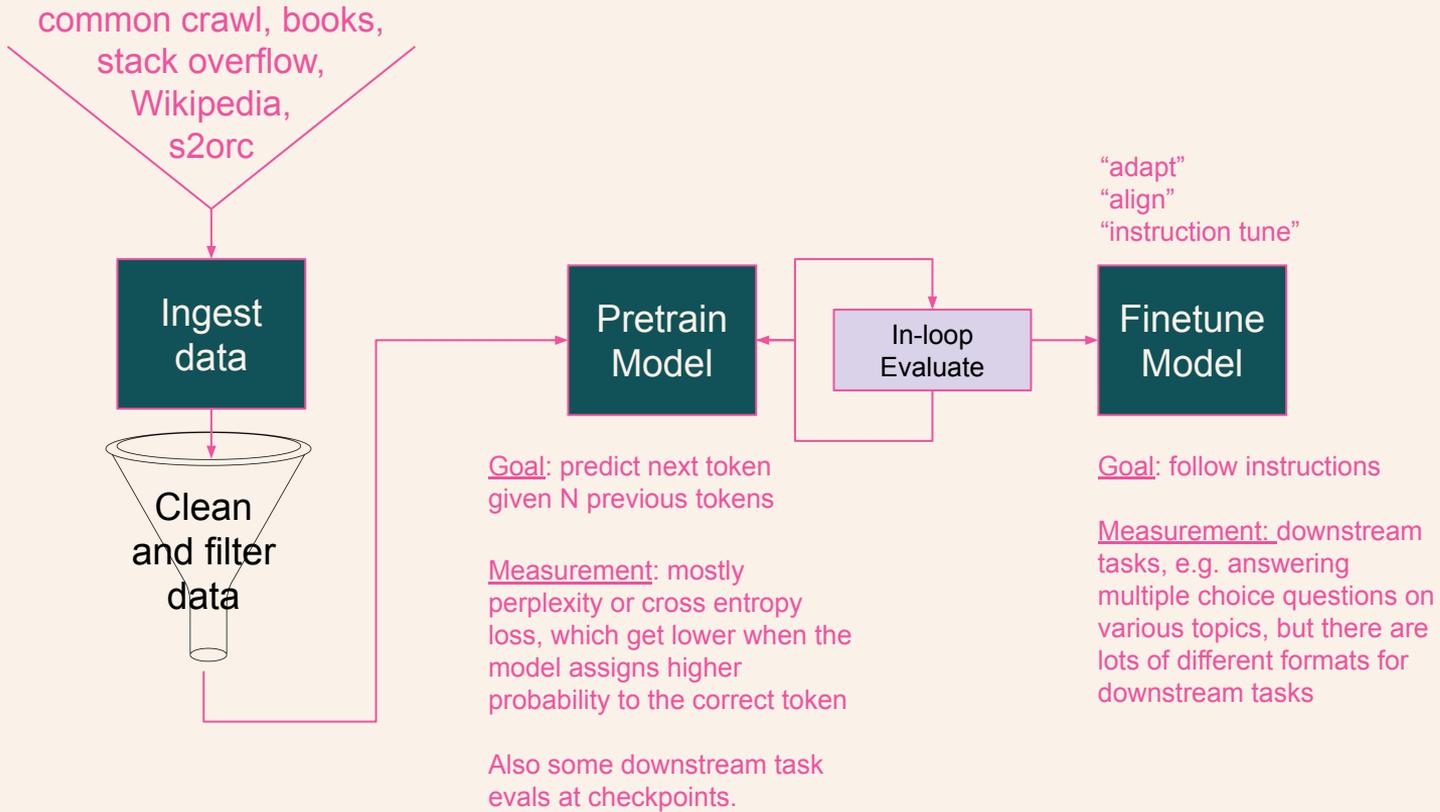
Modular data  
curation & mixing

## Preference data

Learn from contrast  
between data points

## Scaled up RL

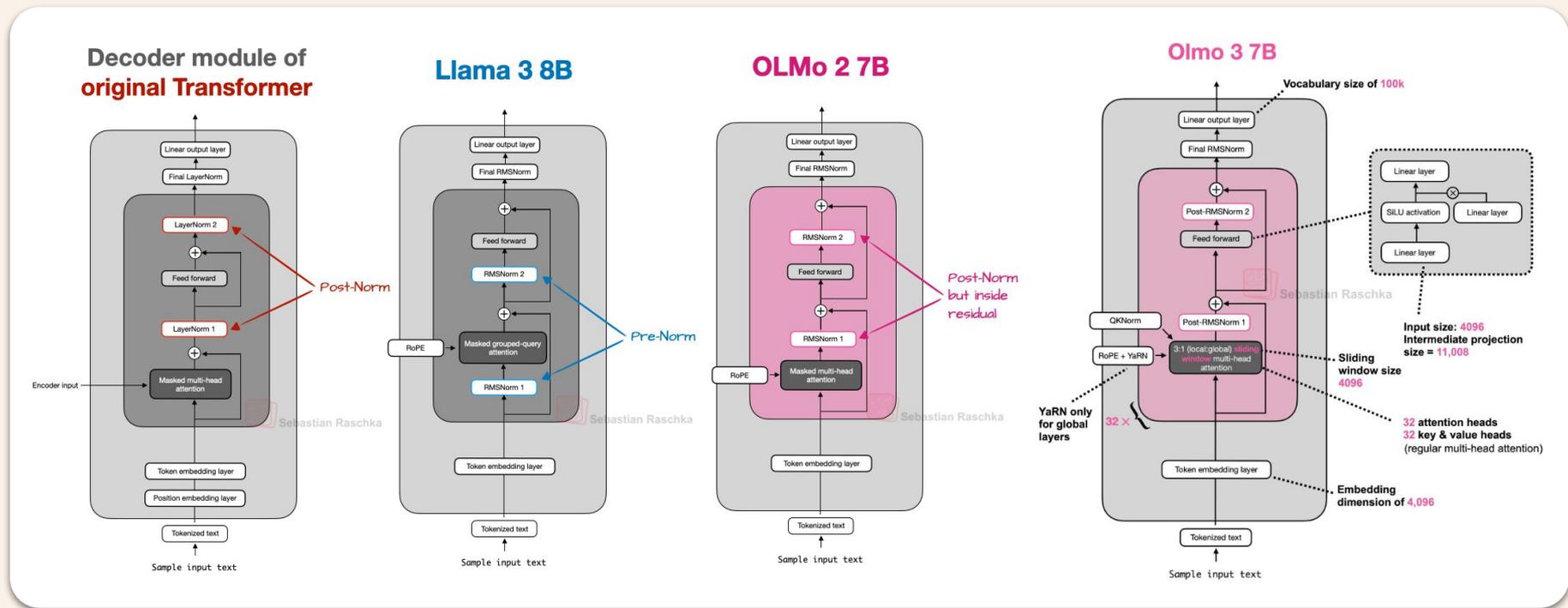
Enable thinking &  
adaptive RL



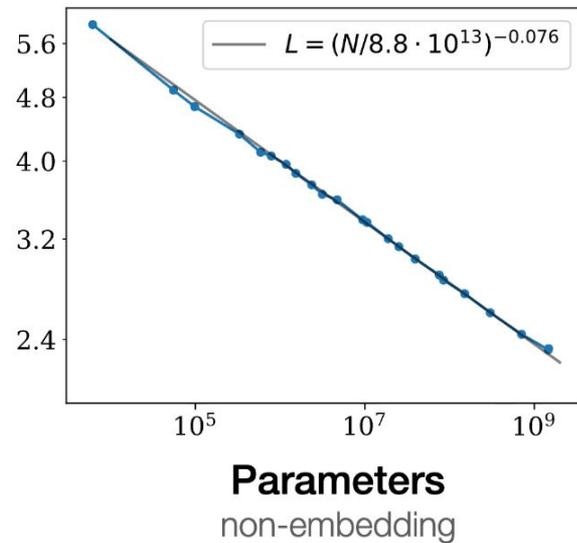
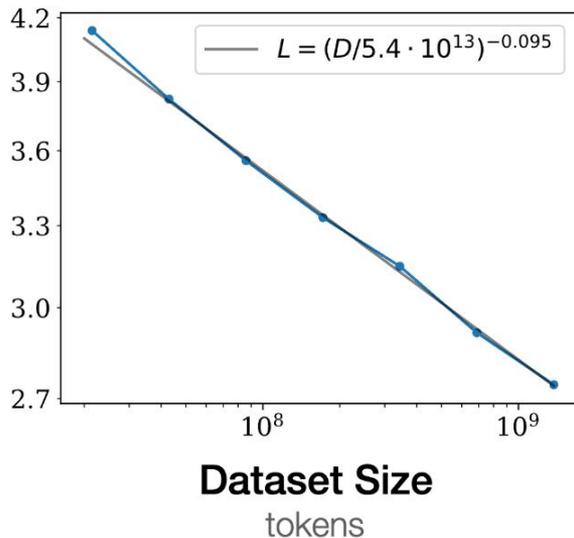
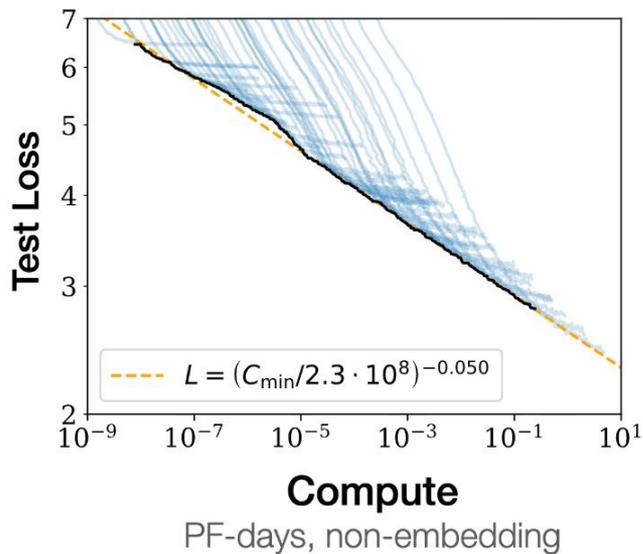


# Background

# Background: Architecture Changes



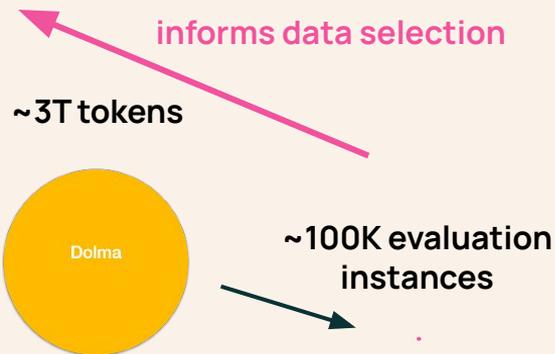
# Background: Scaling Pretrained Models

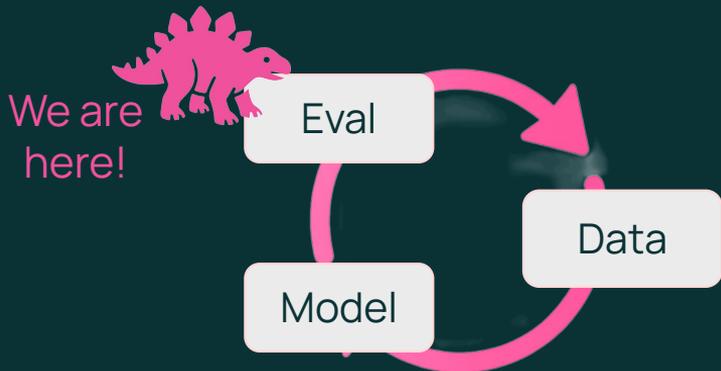


Rough estimate:  
text in ~30M books in the US Library of  
Congress  $\approx$  between 1/400 and 1/24 PB

Common Crawl  
~1 PB

## Background: Data





## Signal and Noise: A Framework for Reducing Uncertainty in Language Model Evaluation

David Heineman<sup>1\*</sup> Valentin Hofmann<sup>1,2\*</sup> Ian Magnusson<sup>1,2\*</sup> Yuling Gu<sup>1\*</sup>  
Noah A. Smith<sup>1,2\*</sup> Hannaneh Hajishirzi<sup>1,2\*</sup> Kyle Lo<sup>1</sup> Jesse Dodge<sup>1</sup>

<sup>1</sup>Allen Institute for Artificial Intelligence

<sup>2</sup>Paul G. Allen School of Computer Science & Engineering, University of Washington  
contact: davidh@allenai.org

### Abstract

Developing large language models is expensive and involves making decisions with small experiments, typically by evaluating on large, multi-task evaluation suites. In this work, we analyze specific properties which make a benchmark more reliable for such decisions, and interventions to design higher-quality evaluation benchmarks. We introduce two key metrics that show differences in current benchmarks: *signal*, a benchmark's ability to separate better models from worse models, and *noise*, a benchmark's sensitivity to random variability between training steps. We demonstrate that benchmarks with a better *signal-to-noise* ratio are more reliable when making decisions at small scale, and those with less *noise* have lower scaling law prediction error. These results suggest that improving *signal* or *noise* will lead to more useful benchmarks, so we introduce three interventions designed to directly affect *signal* or *noise*. For example, we propose that switching to a metric that has better *signal* and *noise* (e.g., perplexity rather than accuracy) leads to better reliability and improved scaling law error. We also find that filtering noisy subtasks, to improve an aggregate *signal-to-noise* ratio, leads to more reliable multi-task evaluations. We also find that averaging the output of a model's intermediate checkpoints to reduce *noise* leads to consistent improvements. We conclude by recommending that those creating new benchmarks, or selecting which existing benchmarks to use, aim for high *signal* and low *noise*. We use 30 benchmarks for these experiments, and 375 open-weight language models from 60M to 32B parameters, resulting in a new, publicly available dataset of 900K evaluation benchmark results, totaling 200M instances.

[github.com/allenai/signal-and-noise](https://github.com/allenai/signal-and-noise) [huggingface.co/datasets/allenai/signal-and-noise](https://huggingface.co/datasets/allenai/signal-and-noise)

### 1 Introduction

Language model development is expensive. During the development process, researchers need to make decisions such as what architecture to use, what training methods to employ, and what data to

# How to make sense of the wall of results?

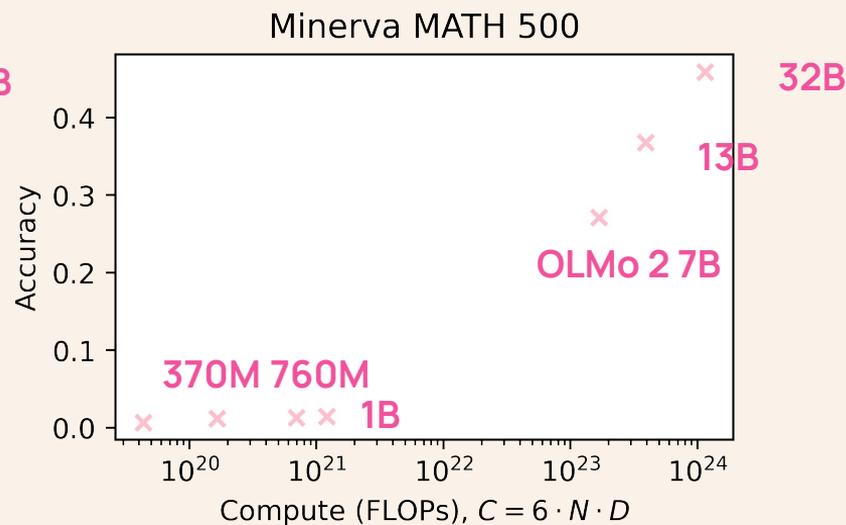
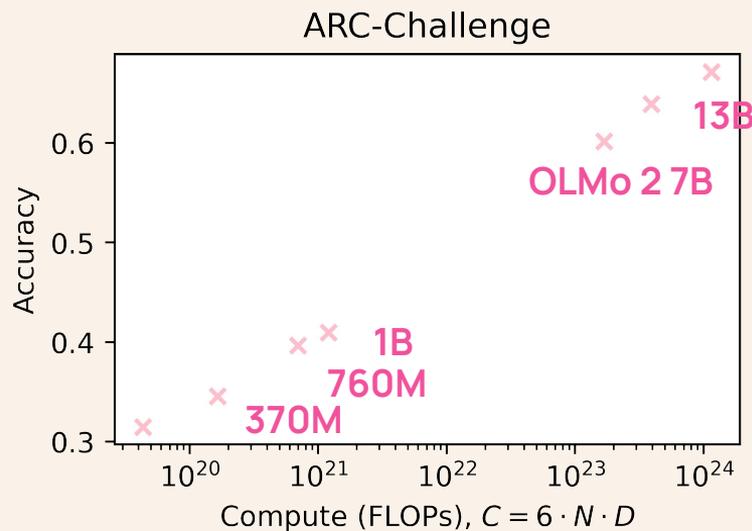
	Fully-open Models						Open-weight Models					
	Olmo 3 32B	Marin 32B	Apertus 70B	Gaperon 24B	LLM 360 K2V270B	OLMO 2 32B	Owen 2.5 32B	Gemma 3 27B	Mistral 3.1 24B	Seed 36B	Gemma 2 27B	Llama 3.1 70B
<b>OlmoBaseEval Math</b>	61.9	49.3	39.7	20.7	46.2	53.9	64.7	63.2	59.5	15.3	57.5	62.0
GSMsk	80.6	69.1	63.0	33.3	66.7	77.6	81.1	81.3	79.3	26.9	76.3	81.2
GSM Symbolic	61.2	42.0	38.6	14.5	44.4	53.1	56.2	61.2	59.1	10.3	57.3	64.6
MATH	43.8	36.8	17.4	14.2	27.4	31.0	56.7	47.0	40.1	8.7	38.8	40.2
<b>OlmoBaseEval Code</b>	39.7	30.8	23.3	19.4	35.2	20.5	48.3	41.6	42.4	54.9	41.0	36.3
BigCodeBench	43.7	34.5	24.0	17.0	39.8	22.2	48.1	44.0	46.4	50.7	43.4	43.4
HumanEval	65.8	52.3	32.5	31.2	51.2	29.4	65.6	62.1	65.5	71.3	57.5	57.4
DeepSeek LeetCode	2.0	1.3	1.2	0.0	2.3	0.8	8.0	5.8	0.1	13.0	4.7	0.2
DS 1000	29.4	26.3	17.8	11.0	25.4	20.4	43.3	34.3	36.3	44.0	29.7	29.5
MBPP	59.6	52.1	37.6	36.7	53.5	37.1	69.8	60.0	61.9	72.0	61.7	55.5
MultiPL HumanEval	36.0	18.5	18.4	13.0	31.3	10.5	49.7	37.7	39.0	69.2	40.3	32.2
MultiPL MBPPP	41.5	30.5	31.3	26.5	42.8	23.2	53.6	47.2	47.7	63.8	49.7	35.9
<b>OlmoBaseEval MC STEM</b>	74.5	75.9	70.0	56.2	75.6	75.3	82.2	80.2	81.5	83.4	75.6	80.1
ARC MC	94.7	93.4	90.7	72.7	93.0	94.4	97.0	95.8	96.2	97.3	94.1	95.2
MMLU STEM	70.8	68.4	57.8	45.3	64.7	64.7	79.7	74.9	76.1	82.8	65.8	70.0
MedMCQA MC	57.6	61.8	55.9	42.6	63.7	60.2	68.8	64.7	68.8	69.6	61.8	67.8
MedQA MC	53.8	60.8	52.4	35.4	61.4	62.2	68.4	68.7	70.4	70.1	61.0	72.3
SciQ MC	95.5	95.1	93.3	84.9	95.3	95.1	97.1	96.8	96.3	97.1	95.1	95.4
<b>OlmoBaseEval MC Non-STEM</b>	85.6	84.5	78.5	64.1	83.5	84.2	89.3	86.7	87.9	89.0	83.2	86.1
MMLU Humanities	78.3	78.9	74.1	56.7	79.3	79.7	85.0	80.5	82.7	85.7	79.3	83.4
MMLU Social Sci.	84.0	83.7	79.2	58.9	84.9	84.5	88.4	86.2	88.6	90.1	85.8	87.4
MMLU Other	75.1	75.4	70.1	55.4	76.3	75.6	81.2	80.2	81.9	82.4	76.9	79.4
CSQA MC	82.3	80.1	76.9	60.6	78.6	81.2	89.9	79.0	80.5	81.1	78.1	79.0
PiQA MC	85.6	90.5	79.0	72.0	87.3	87.7	93.3	90.3	91.0	92.5	89.0	91.5
SocialIQa MC	83.9	82.4	79.3	71.3	81.2	82.3	86.6	81.2	81.0	84.9	81.0	83.5
CoQA Gen2MC MC	96.4	93.9	87.5	67.3	92.0	94.4	96.8	95.8	94.9	96.9	94.3	95.1
DROP Gen2MC MC	87.2	71.0	56.5	48.0	64.8	68.6	86.6	84.6	86.5	90.1	66.6	70.3
Jeopardy Gen2MC MC	92.3	95.3	93.2	77.0	95.3	96.6	97.0	95.9	97.2	96.2	92.0	97.1
NaturalQs Gen2MC MC	78.0	81.0	71.9	47.5	82.4	78.6	79.9	82.0	84.6	81.4	74.5	82.4
SQuAD Gen2MC MC	98.2	97.6	95.7	90.0	96.7	97.4	97.9	97.7	97.9	98.1	97.5	97.7
<b>OlmoBaseEval GenQA</b>	79.8	80.3	75.0	65.3	77.1	79.1	68.5	73.5	78.0	76.0	72.9	81.6
HellaSwag RC	84.8	87.2	84.5	75.2	87.6	87.5	86.3	86.0	86.2	84.8	86.7	88.4
Winogrande RC	90.3	90.5	87.7	80.3	88.9	89.4	87.5	91.3	90.8	89.3	90.8	91.7
Lambada	75.7	76.7	74.8	58.3	76.8	77.0	76.2	77.5	79.3	76.1	76.9	79.6
Basic Skills	93.5	91.1	87.5	83.2	90.6	88.7	94.2	94.9	91.9	96.0	93.2	92.4
DROP	80.9	76.5	56.3	59.4	69.7	76.3	53.7	75.9	74.9	76.1	73.2	78.3
Jeopardy	75.3	80.5	77.2	58.9	79.8	79.1	74.0	82.1	80.3	77.4	80.7	84.0
NaturalQs	49.0	55.1	43.1	33.5	47.6	51.4	39.3	49.2	45.1	30.7	47.1	53.1
SQuAD	94.5	94.4	90.7	89.3	91.2	94.0	64.9	92.4	92.6	89.1	93.0	92.9
CoQA	74.1	70.7	72.8	49.8	61.5	68.7	40.4	12.4	61.1	64.4	14.9	73.9
<b>OlmoBaseEval HeldOut</b>												
LBPP	21.8	17.3	8.1	4.3	13.4	8.2	40.3	17.7	30.3	42.6	19.7	11.8
BBH	77.6	70.1	58.8	36.6	73.2	64.6	81.1	77.4	81.4	85.0	74.8	80.8
MMLU Pro MC	49.7	48.1	39.6	21.3	45.3	46.9	61.1	53.1	58.9	62.2	47.6	50.4
Deepmind Math	29.6	26.7	20.1	28.3	32.5	22.0	40.7	30.4	35.3	31.3	27.6	40.2

**Table 2 Results comparing Olmo 3 Base 32B to other base models using the OlmoBaseEval Main suite** (details in Section §3.3). OLMO 3 was not evaluated on held-out benchmarks prior to release.

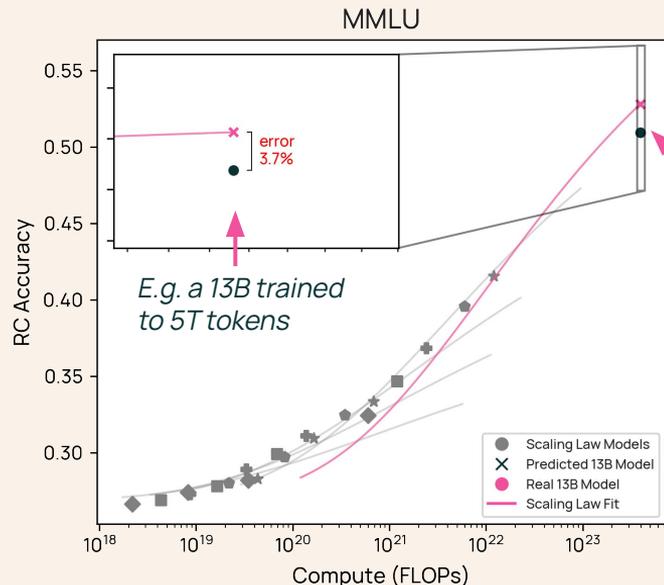
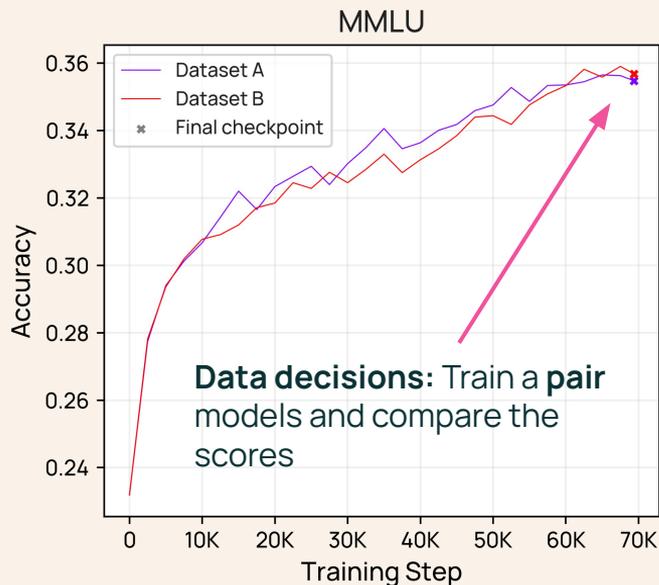
We evaluate many families of abilities (QA, Math, Coding, ...)

HellaSwag CSQA DROP TriviaQA DS-1000 BigCodeBench  
CoQA Lambada PiQA ARC-C MedMCQA HumanEval MBPP  
WinoGrande BoolQ ARC-E MMLU GSM8K CruxEval  
SocialIQA Jeopardy SQuAD Minerva MATH

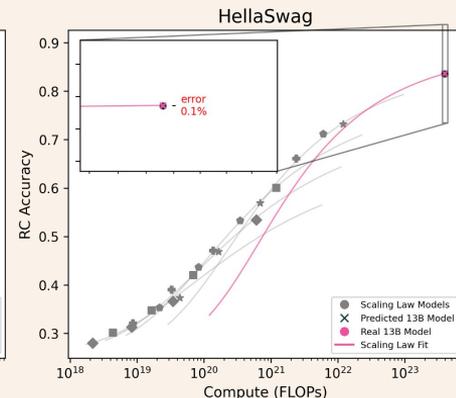
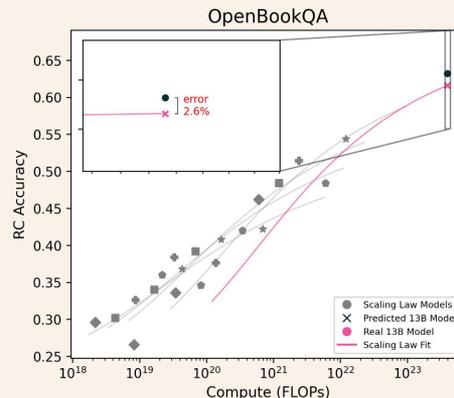
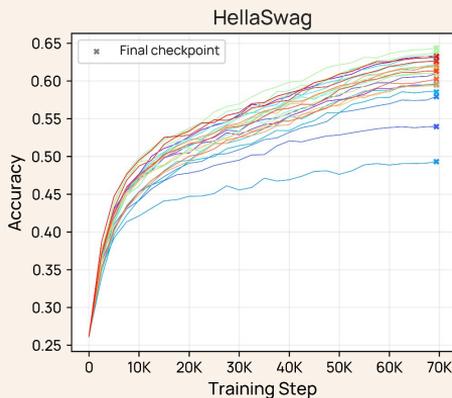
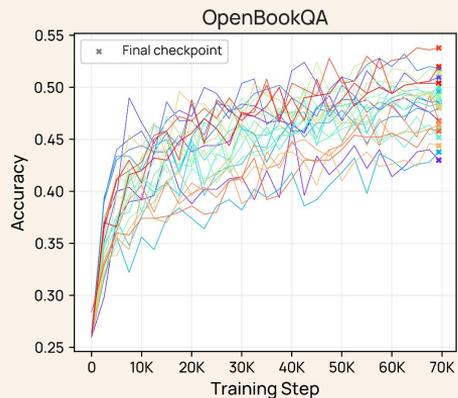
## ... but how to get **signal** for small-scale decisions?



# Making decisions in pretraining == extrapolating eval

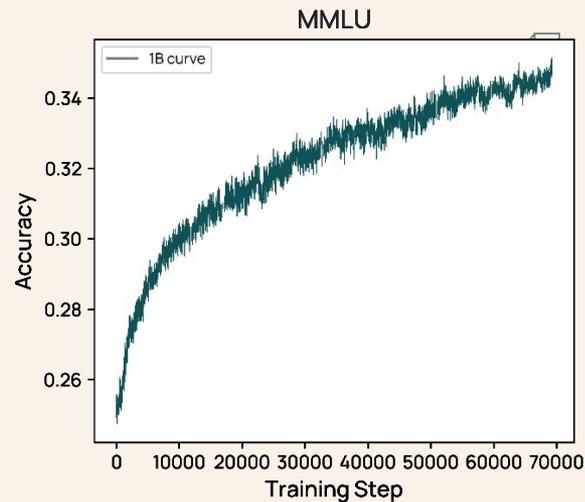
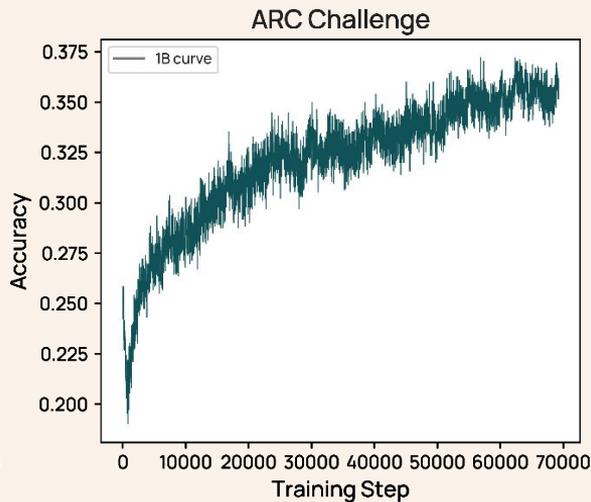
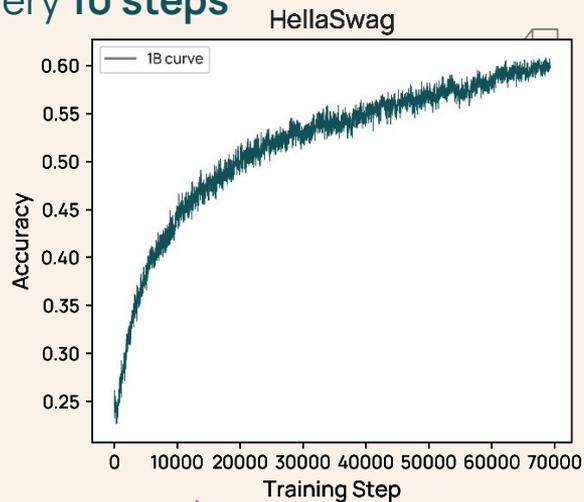


DataDecide: How to Predict Best Pretraining Data with Small Experiments (ICML, 2025)  
Establishing Task Scaling Laws via Compute-Efficient Model Ladders (COLM, 2025)

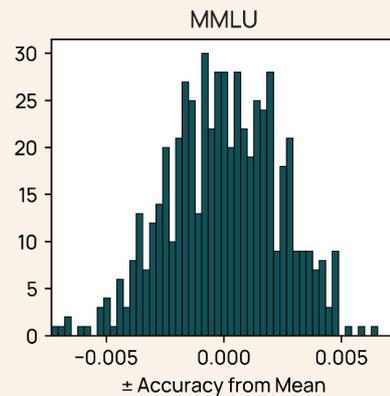
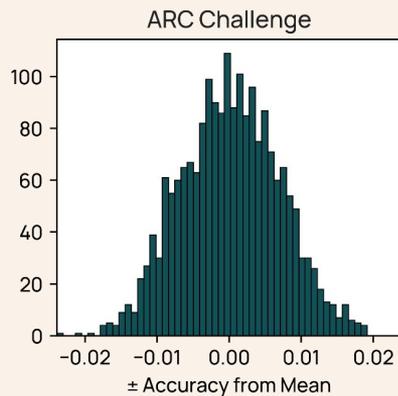
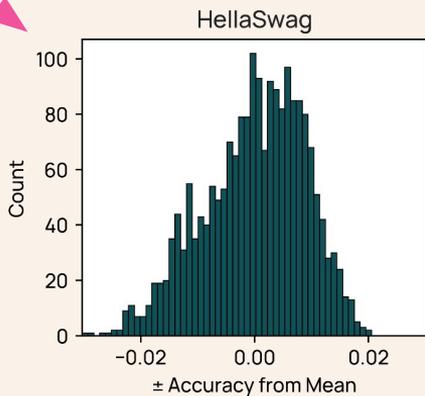


Why do so many predictions fail -  
but some don't?

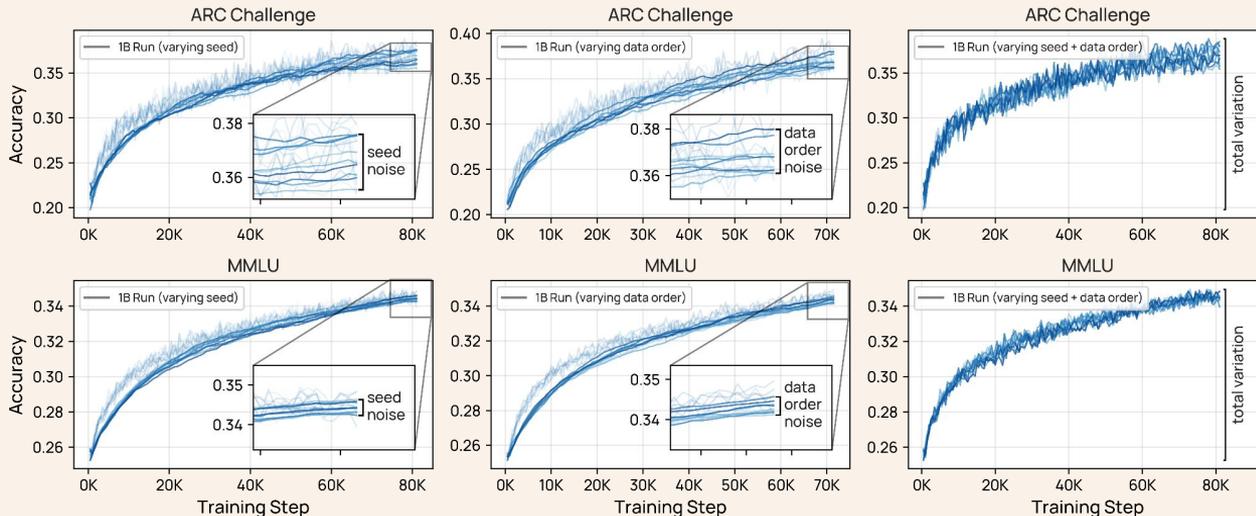
# 1B-5xC + eval every 10 steps



Final 20% of checkpoints

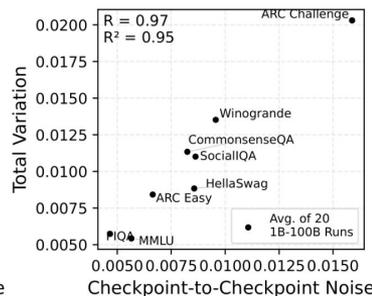
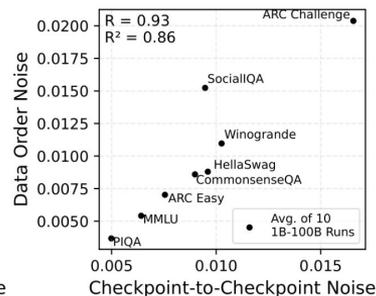
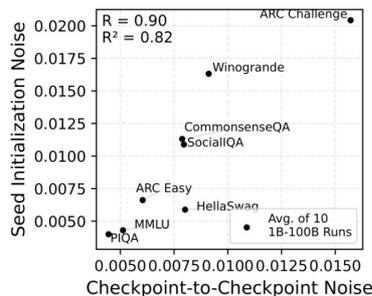


# Many sources of noise ...



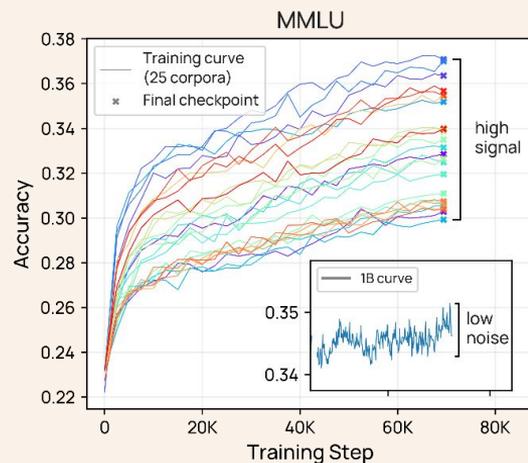
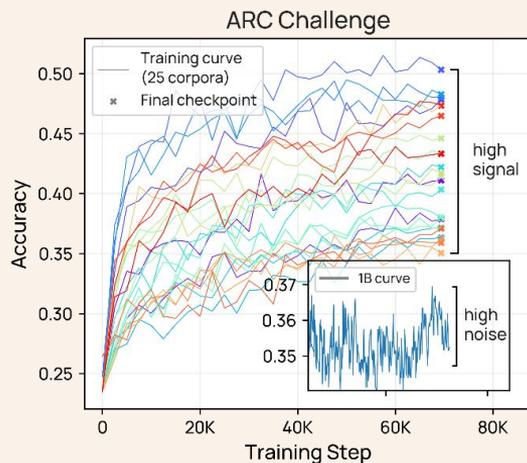
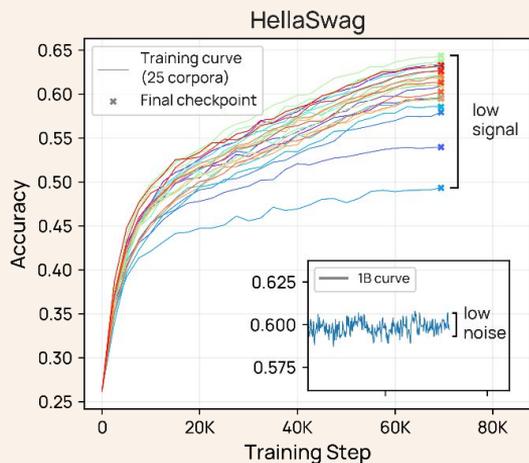
10 1B-5xC models varying:  
data order and random seed

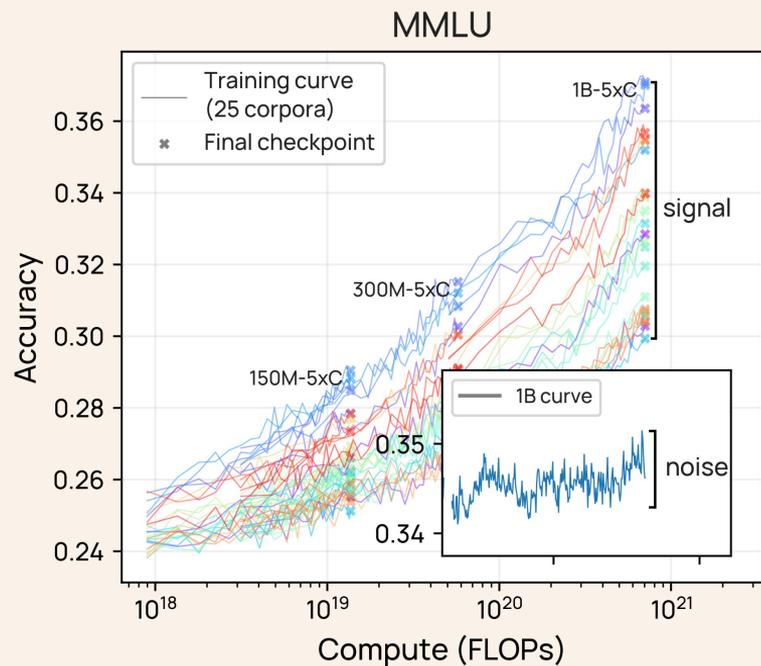
... seed noise correlates with step-to-step noise!



+ Total variation correlates w/ step-to-step noise

# Inter-checkpoint variance is not the whole story! We need to measure both **signal** and **noise**





**Signal** (max difference between model scores):

$$\text{Rel. Dispersion}(M) = \max_{j,k} |m_j - m_k| / \bar{m}$$

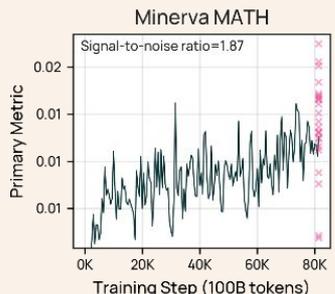
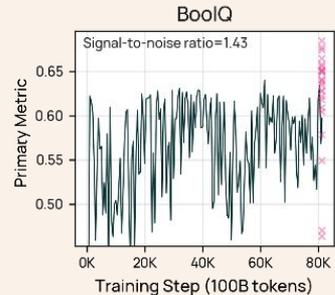
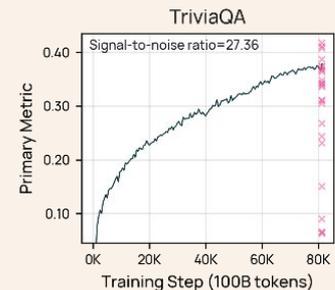
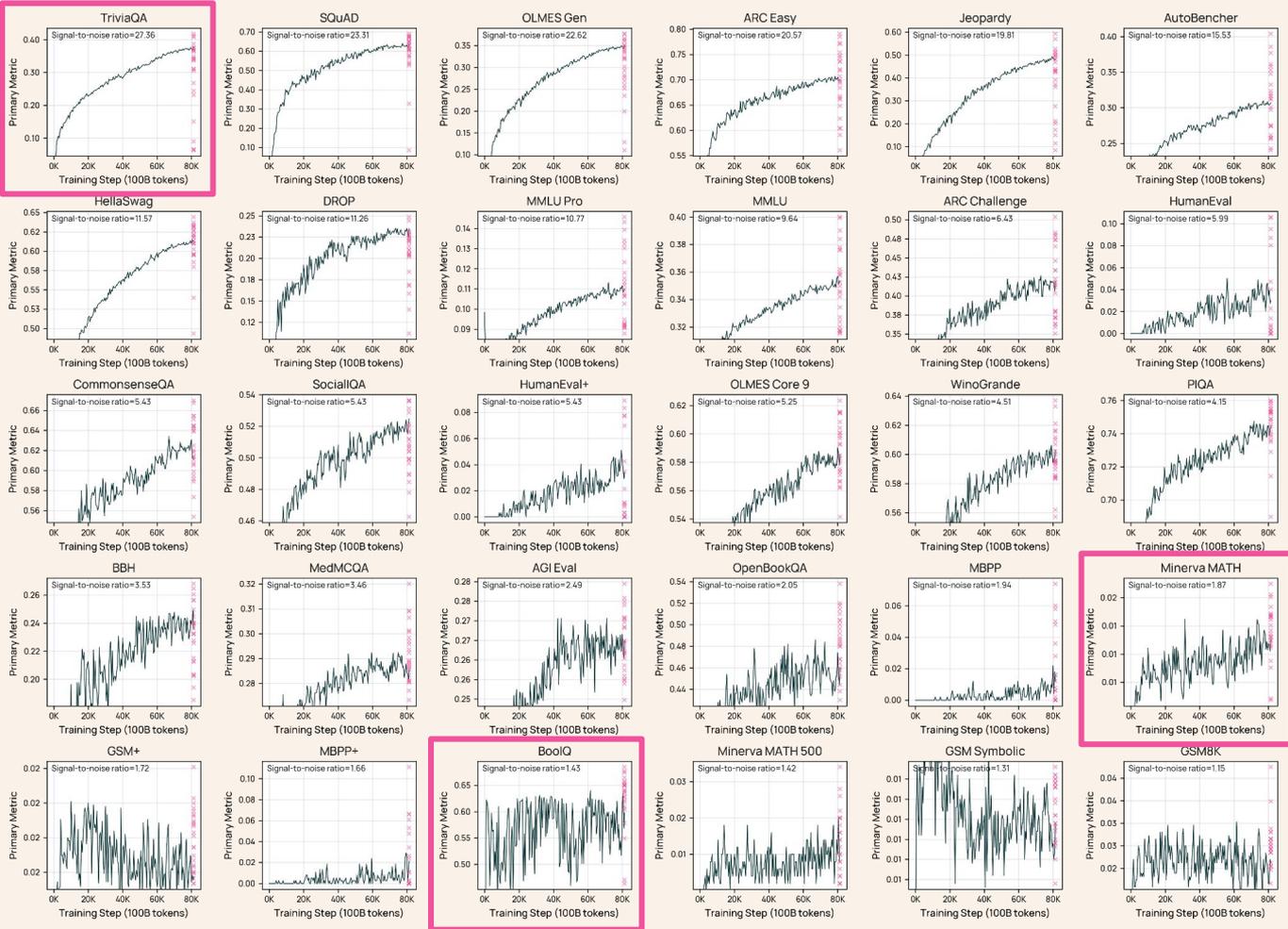
**Noise** (ckpt-to-ckpt noise using last n steps):

$$\text{Rel. Std.}(m) = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (m_i - \bar{m})^2 / \bar{m}}$$

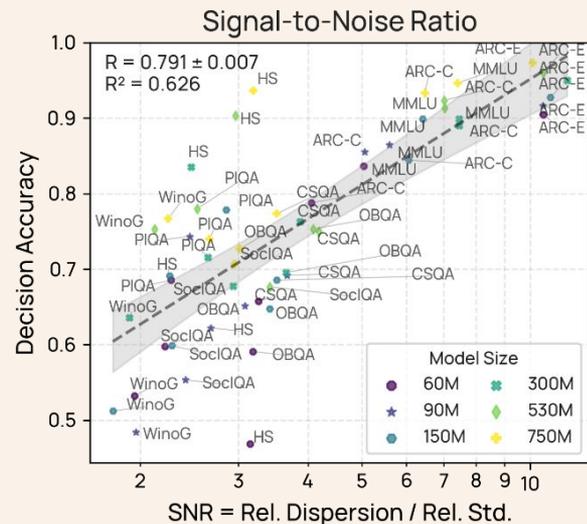
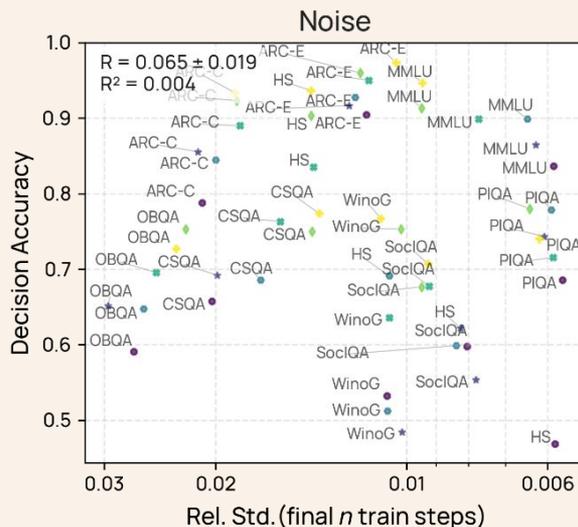
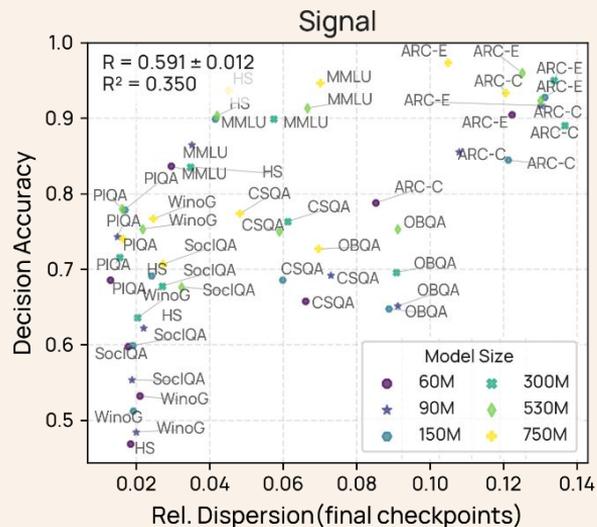
Signal-to-Noise Ratio =

$$\frac{\text{Rel. Dispersion}(\text{final train checkpoint})}{\text{Rel. Std.}(\text{final } n \text{ train checkpoints})}$$

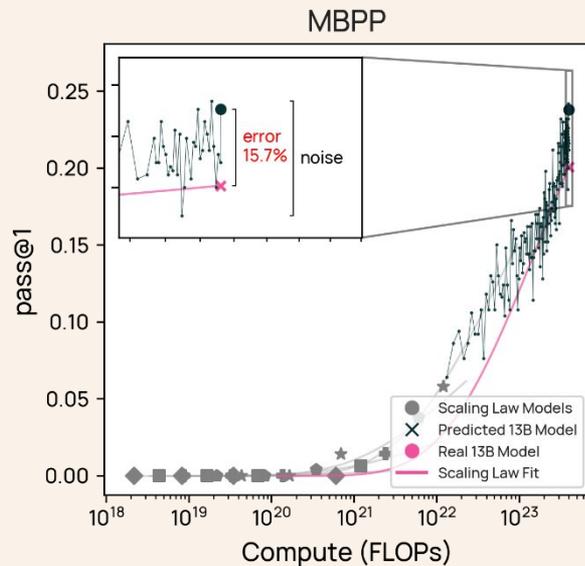
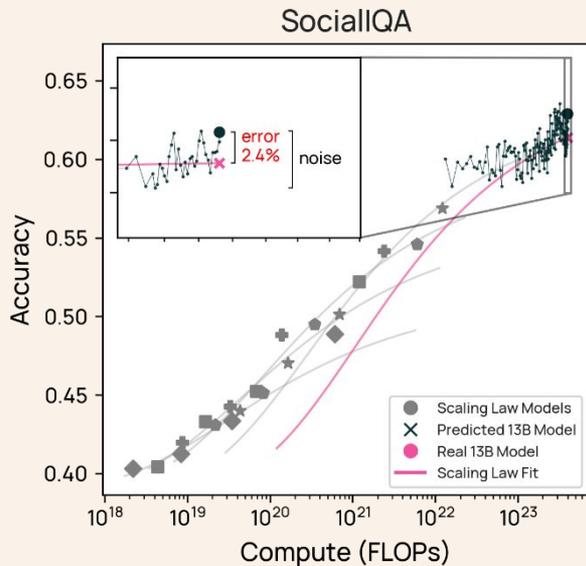
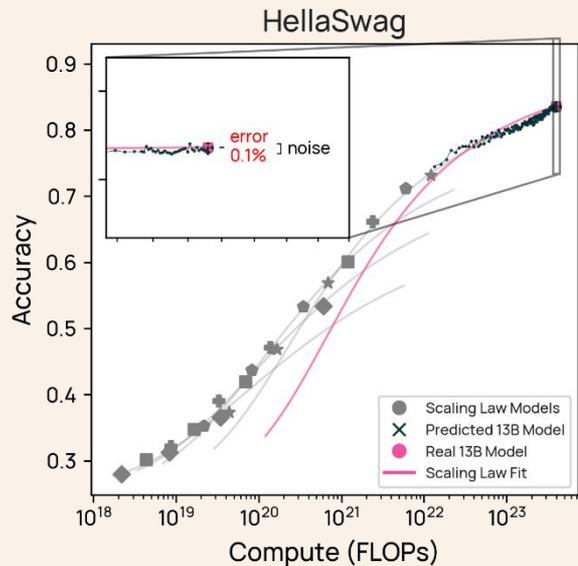
# ← SNR at the 1B scale



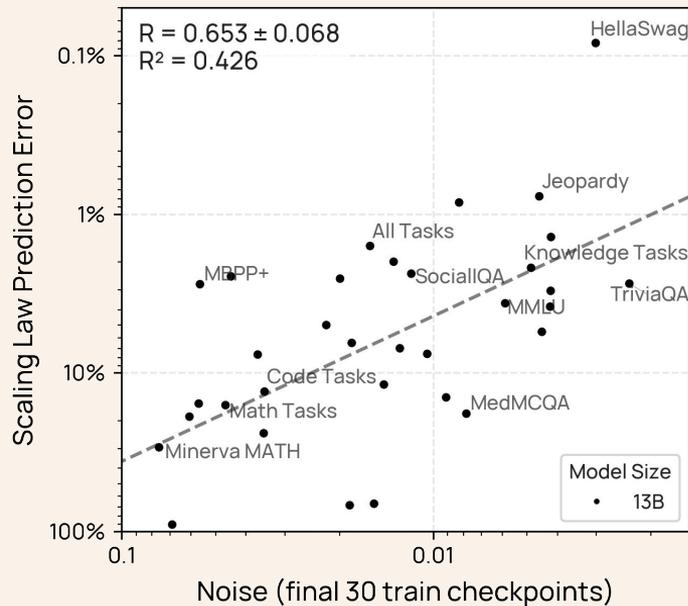
# Only **signal** or **noise** alone do not explain rank agreement from small to large scale... .. but the **signal-to-noise ratio** does!

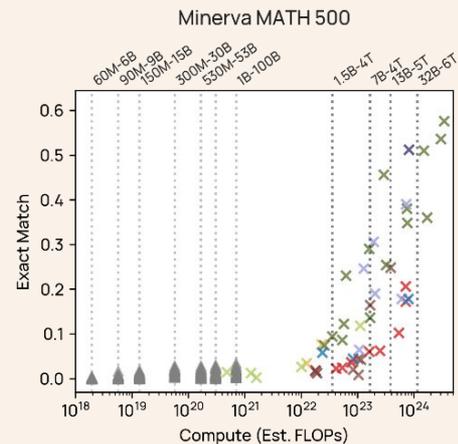
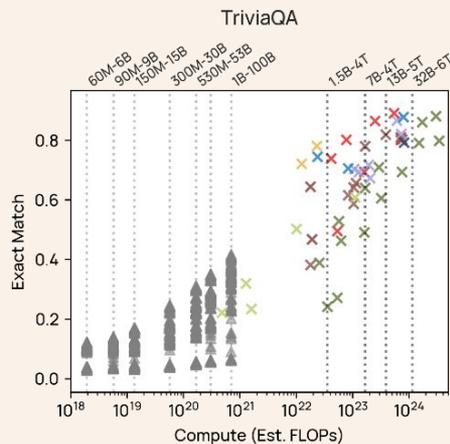
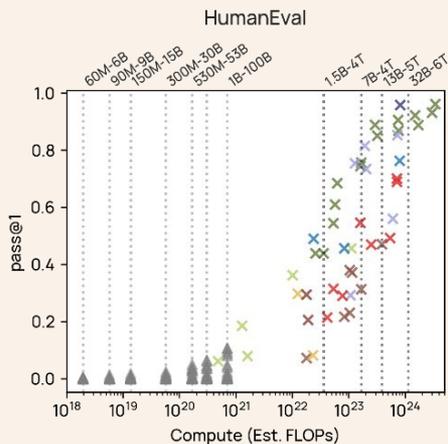
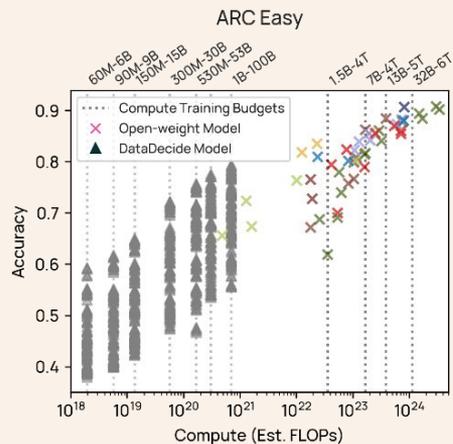


# Predicting task performance using scaling laws is sensitive to noise!



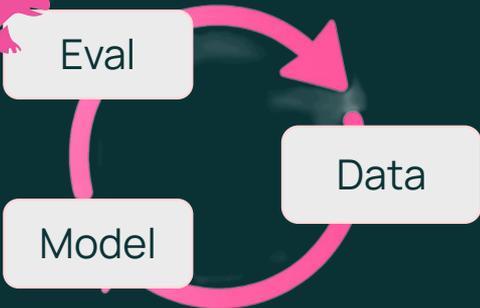
# Predicting task performance using **scaling laws** is sensitive to noise!





The signal-to-noise ratio is specifically useful as **signal of benchmarks changes** for large train compute budgets

We are here!



## Olmo 3

Olmo Team\*

Allyson Ettlinger<sup>\*1</sup> Amanda Bertsch<sup>\*1,3</sup> Bailey Kuehl<sup>\*1</sup> David Graham<sup>\*1</sup>  
 David Heineman<sup>\*1</sup> Dirk Groeneveld<sup>\*1</sup> Faeze Brahman<sup>\*1</sup> Finbarr Timbers<sup>\*1</sup>  
 Hamish Ivison<sup>\*1,2</sup> Jacob Morrison<sup>\*1,2</sup> Jake Poznanski<sup>\*1</sup> Kyle Lo<sup>\*1,2</sup> Luca Soldaini<sup>\*1</sup>  
 Matt Jordan<sup>\*1</sup> Mayee Chen<sup>\*1,4</sup> Michael Noukhovitch<sup>\*1,5,6</sup> Nathan Lambert<sup>\*1</sup>  
 Pete Walsh<sup>\*1</sup> Pradeep Dasigi<sup>\*1</sup> Robert Berry<sup>\*1</sup> Saumya Malik<sup>\*1</sup> Saurabh Shah<sup>\*1</sup>  
 Scott Geng<sup>\*1,2</sup> Shane Arora<sup>\*1</sup> Shashank Gupta<sup>\*1</sup> Taira Anderson<sup>\*1</sup> Teng Xiao<sup>\*1</sup>  
 Tyler Murray<sup>\*1</sup> Tyler Romero<sup>\*1</sup> Victoria Graf<sup>\*1,2</sup>

Akari Asai<sup>1,3</sup> Akshita Bhagia<sup>1</sup> Alexander Wettig<sup>1</sup> Alisa Liu<sup>2</sup> Aman Rangapur<sup>1</sup>  
 Chloe Anastasiades<sup>1</sup> Costa Huang<sup>1</sup> Dustin Schwenk<sup>1</sup> Harsh Trivedi<sup>1</sup> Ian Magnusson<sup>1,2</sup>  
 Jaron Lochner<sup>1</sup> Jiacheng Liu<sup>1</sup> Lester James V. Miranda<sup>1</sup> Maarten Sap<sup>1,3</sup> Malia Morgan<sup>1</sup>  
 Michael Schmitz<sup>1</sup> Michal Querquin<sup>1</sup> Michael Wilson<sup>1</sup> Regan Huff<sup>1</sup> Ronan Le Bras<sup>1</sup>  
 Rui Xin<sup>2</sup> Rulin Shao<sup>2</sup> Sam Sijonasberg<sup>1</sup> Shannon Zhai<sup>1</sup> Shen<sup>1</sup> Shuyue Stella Li<sup>1</sup>  
 Tucker Wilde<sup>1</sup> Valentina Pyatkin<sup>1</sup> Will Merrill<sup>1</sup> Yapei Chang<sup>2</sup> Yuling Gu<sup>1</sup> Zhiyuan Zeng<sup>1,2</sup>

Ashish Sabharwal<sup>1</sup> Luke Zettlemoyer<sup>2</sup> Pang Wei Koh<sup>1,2</sup>  
 Ali Farhadi<sup>1,2</sup> Noah A. Smith<sup>\*1,2</sup> Hannaneh Hajishirzi<sup>\*1,2</sup>

<sup>1</sup>Allen Institute for AI <sup>2</sup>University of Washington <sup>3</sup>Carnegie Mellon University <sup>4</sup>Stanford University <sup>5</sup>Mila  
<sup>6</sup>Université de Montréal <sup>7</sup>Princeton University <sup>8</sup>Massachusetts Institute of Technology <sup>9</sup>University of Maryland

\*OLMO 3 was a team effort; authors sorted alphabetically. \*marks core contributors. See author contributions here.

- 🟡 **Olmo 3 Base:** Olmo-3-1025-7B Olmo-3-1125-32B
- 🟡 **Olmo 3 Think:** Olmo-3-7B-Think Olmo-3-13-32B-Think
- 🟡 **Olmo 3 Instruct:** Olmo-3-7B-Instruct Olmo-3-1-32B-Instruct
- 🟡 **Olmo 3 RL Zero:** Olmo-3-7B-RL-Zero-(MathCode|IF|General|Mix) Olmo-3-1-7B-RL-Zero-(Math|Code)
- 🟢 **Base Data:** Pretrain: Dolma 3 Mix Midtrain: Dolma 3 Dolmino Mix Long-ctx: Dolma 3 Longino Mix
- 🟢 **Think Data:** Dolci-Think-(SFT|DPO|RL)-7B Dolci-Think-(SFT|DPO|RL)-32B
- 🟢 **Instruct Data:** Dolci-Instruct-(SFT|DPO|RL)
- 🟢 **RL-Zero Data:** Dolci-RL-Zero-(Math|Code|IF|General)-7B Dolci-RL-Zero-Mix-7B

### 3.3 Experimental Design and Evaluation

Model development requires many iterative data and training decisions. However, benchmarks are not perfect decision-making tools: different evaluations are only sensitive for making development decisions across specific ranges of scale and capability (Magnusson et al., 2025). Models trained at small compute scales are known to exhibit random-chance performance on math, code, and multiple-choice question answering (MCQA) tasks (Wei et al., 2022; Gu et al., 2024b), and benchmark noise can reduce the ability to trust small differences in scores (Heineman et al., 2025). To address these problems, we develop **OLMOBASEEVAL**, a collection of benchmark suites to support decision-making during base model development. **OLMOBASEEVAL** features the

We increase **OLMO 3**’s family of state-of-the-art, fully-open language models at the 7B and 32B parameter scales. **OLMO 3** model construction targets long-context reasoning, function calling, coding, instruction following, general chat, and knowledge recall. This release includes the entire model flow, i.e., the full lifecycle of the family of models, including every stage, checkpoint, data point, and dependency used to build it. Our flagship model, **OLMO 3.1 THINK 32B**, is the strongest fully-open thinking model released to-date.

# OLMo 2 → Olmo 3 base evals:

- + capabilities (math, code)
- + coverage of science QA, Basic Skills
- + coverage across formats (Gen2MC)

## 3 new methods for Olmo 3 Base eval:

- (1) clustering tasks,
- (2) scaling analysis,
- (3) signal-to-noise analysis

task	split	# inst (total)	# shots	metric	reference
<b>Multiple-choice tasks</b>					
ARC-Challenge (ARC_C)	Test	1172	5	pmi	(Clark et al., 2018)
BoolQ	Val	1000 (3270)	5	none	(Clark et al., 2019)
HellaSwag (HSwag)	Val	1000 (10042)	5	char	(Zellers et al., 2019)
MMLU <sup>†</sup>	Test	14042	5	char	(Hendrycks et al., 2021a)
WinoGrande (WinoG)	Val	1267	5	none	(Sakaguchi et al., 2020)
<b>Generative tasks</b>					
DROP	Val	1000 (9536)	5	F1	(Dua et al., 2019)
Natural Questions (NatQs)	Val	1000 (3610)	5	F1	(Kwiatkowski et al., 2019)
<b>Held-out tasks</b>					
AGIEval English	Test	2646	1	MCF	(Zhong et al., 2024)
GSM8K	Test	1319	8 (CoT)	EM	(Cobbe et al., 2021)
MMLU-Pro	Test	12032	5	MCF	(Wang et al., 2024)
TriviaQA	Val	7993	5	F1	(Joshi et al., 2017)

**Table 20** Details of OLMES benchmarks used in OLMo 2 evaluation, with standardized choices of dataset split, number of instances to use, along with total number if sampling was used. For multiple-choice tasks, when using the Cloze/Completion Formulation (CF), the “metric” column specifies which normalization scheme to use. Following the OLMES standard, we evaluate each model using both the MCF (Multiple-Choice Formulation) and CF formulations, and the best performing one is used. For efficiency reasons, we limit MMLU and held-out multiple-choice evaluations to MCF only as all the relevant models strongly prefer that format for these tasks.

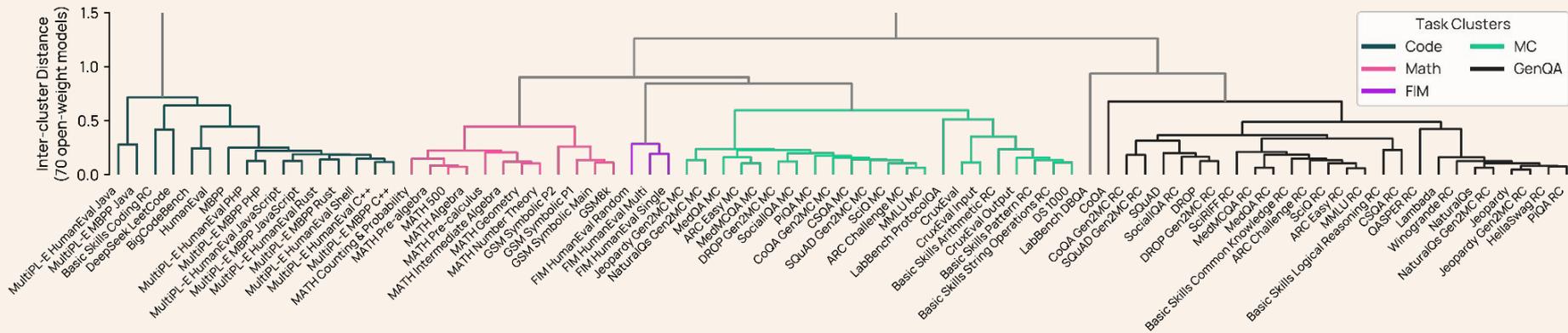


task	ICL	format	metric	temp	top-p	max toks	p@k (n)	# sub	
<b>Base Main Suite</b>									
Math	GSM8K* (2021)	8 <sup>α</sup>	CoT EM	pass@k	0.6	0.6	512	1, 4 (8)	-
	GSM Symbolic* (2024)	8 <sup>α</sup>	CoT EM	pass@k	0.6	0.6	512	1, 4 (8)	3
	Minerva MATH* (2022)	4 <sup>α</sup>	CoT EM	pass@k	0.6	0.6	1024	1, 4 (4)	7
	MATH 500* (2022; 2023)	4 <sup>α</sup>	CoT EM	pass@k	0.6	0.6	1024	1, 16 (32)	-
Code	HumanEval* (2021)	3	Code Exec	pass@k	0.6	0.6	512	1, 16 (32)	-
	MBPP* (2021)	3	Code Exec	pass@k	0.6	0.6	512	1, 16 (32)	-
	BigCodeBench* (2024)	3	Code Exec	pass@k	0.6	0.6	1280	1 (5)	-
	DS 1000* (2022)	3	Code Exec	pass@k	0.6	0.6	1024	1 (5)	-
	Deepseek LeetCode* (2024)	0	Code Exec	pass@k	0.6	0.6	512	1, 16 (32)	-
	MultiPL-E HumanEval* (2022)	0	Code Exec	pass@k	0.6	0.6	1024	1, 16 (32)	6
	MultiPL-E MBPP* (2022)	0	Code Exec	pass@k	0.6	0.6	1024	1, 16 (32)	6
FIM	HumEval FIM Single* (2022)	0	FIM	pass@1	0.8	0.95	512	1 (10)	-
	HumEval FIM Random* (2022)	0	FIM	pass@1	0.8	0.95	512	1 (5)	-
	HumEval FIM Multi* (2022)	0	FIM	pass@1	0.8	0.95	512	1 (1)	-
STEM QA	ARC (2018)	5	MC	Acc	-	-	-	-	2
	MMLU STEM (2021b)	5	MC	Acc	-	-	-	-	19
	MedMCQA* (2022)	5	MC	Acc	-	-	-	-	-
	MedQA* (2021)	5	MC	Acc	-	-	-	-	-
	SciQ* (2017)	5	MC	Acc	-	-	-	-	-
	MMLU Humanities (2021b)	5	MC	Acc	-	-	-	-	13
	MMLU Social Sci. (2021b)	5	MC	Acc	-	-	-	-	12
Non-STEM QA	MMLU Other (2021b)	5	MC	Acc	-	-	-	-	14
	CSQA (2019)	5	MC	Acc	-	-	-	-	-
	PIQA (2020)	5	MC	Acc	-	-	-	-	-
	SocialIQA (2019)	5	MC	Acc	-	-	-	-	-
	DROP Gen2MC* (§A.3.2; 2019)	5	MC	Acc	-	-	-	-	-
	Jeopardy Gen2MC* (§A.3.2; 2024)	5	MC	Acc	-	-	-	-	-
	NaturalQs Gen2MC* (§A.3.2; 2019)	5	MC	Acc	-	-	-	-	-
	SQuAD Gen2MC* (§A.3.2; 2016)	5	MC	Acc	-	-	-	-	-
GenQA	CoQA Gen2MC* (§A.3.2; 2019)	0 <sup>†</sup>	MC	Acc	-	-	-	-	-
	Basic Skills* (§A.3.2)	5	MC	Acc	-	-	-	-	6
	HellaSwag (2019)	5	RC <sub>per-char</sub>	Acc	-	-	-	-	-
	WinoGrande (2020)	5	RC <sub>none</sub>	Acc	-	-	-	-	-
	Lambda (2016)	0	RC <sub>per-char</sub>	Acc	-	-	-	-	-
	Basic Skills* (§A.3.2)	5	RC <sub>per-token</sub>	Acc	-	-	-	-	6
	DROP (2019)	5	GenQA	F1	0	1	100	-	-
	Jeopardy (2024)	5	GenQA	F1	0	1	50	-	-
	NaturalQs (2019)	5	GenQA	F1	0	1	50	-	-
	SQuAD (2016)	5	GenQA	F1	0	1	50	-	-
CoQA (2019)	0 <sup>†</sup>	GenQA	F1	0	1	50	-	-	
<b>Base Held-out Suite</b>									
MMLU Pro (2024a)	5	MC	Acc	-	-	-	-	-	13
LBPP* (2024)	0	Code Exec	pass@k	0.6	0.6	4096	1 (32)	-	
Deepmind Math* (2019)	5	CoT EM	pass@k	0.6	0.6	2048	1 (1)	-	
BigBench Hard (2022)	3	CoT EM	Acc	0.6	0.6	512	1 (1)	55	

**Table 43** Details of the OLMo 3 base evaluation suite. Tasks were formatted as multiple-choice (MC), rank choice (RC), following the setup in Gu et al. (2024b), short-form generative (GenQA), chain-of-thought with exact-match scoring (CoT EM), code execution (Code Exec) or fill-in-the-middle coding (FIM). \* = new additions to the base OLMo 2 suite (OLMo et al., 2024); <sup>†</sup> = few-shot examples are built-in the task; <sup>α</sup> = human-written few-shot examples.

1

# How to handle large # of benchmarks? Group tasks into “clusters” (Math, Code, GenQA) and track multi-task averages



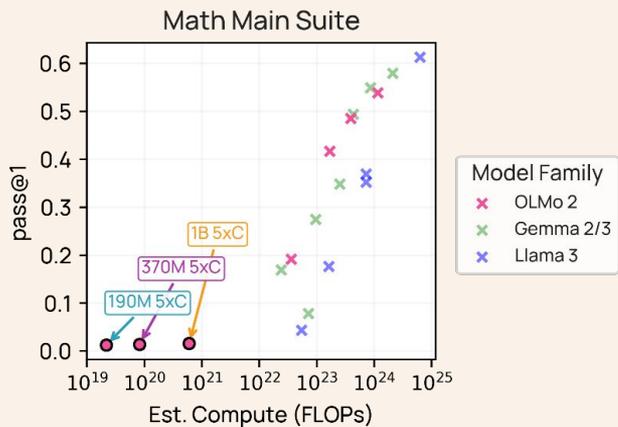
Task averages simplify decision making (e.g. in midtraining)

The image displays a large spreadsheet with multiple columns and rows, representing performance metrics for various models and tasks. A red arrow points from the text to a specific row in the spreadsheet. A blue box highlights a summary table for 'OLMOBASEVAL'.

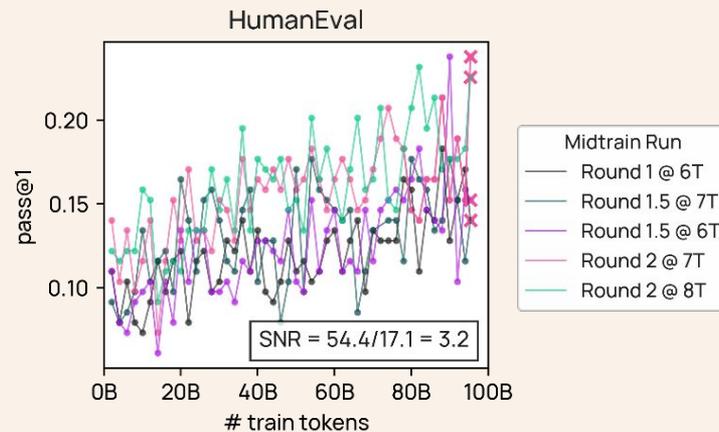
Mix	OLMOBASEVAL							SFT Exps
	Avg	MC STEM	MC Non-STEM	GenQA	Math	Code	FIM	Avg
Round 1	49.7	64.3	75.2	68.3	47.4	23.4	28.4	35.2
Round 3	50.7	64.9	75.7	68.1	48.7	24.4	31.9	35.3
Round 5	53.1	65.3	76.1	70.8	57.1	27.7	29.4	37.3

2 + 3

## Two additional problems:

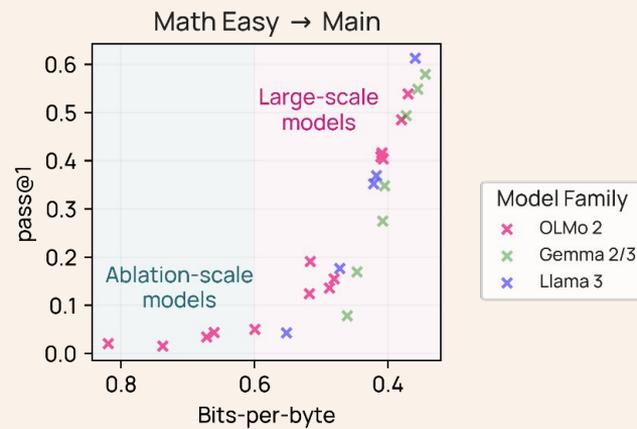
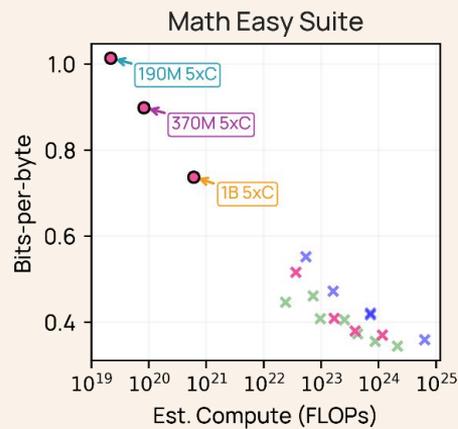
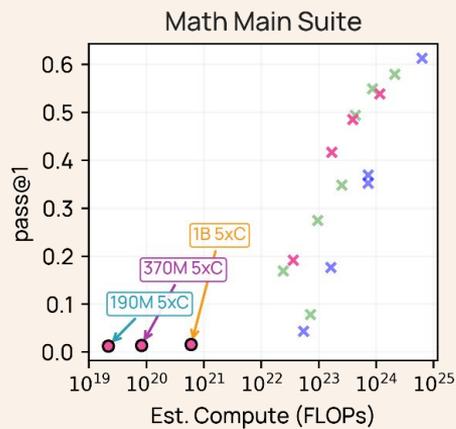


**Signal at small scales**  
(e.g. 30M to 1B)



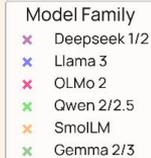
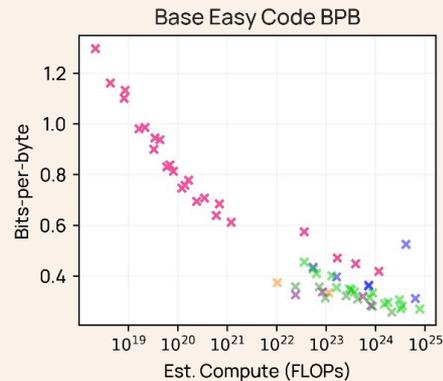
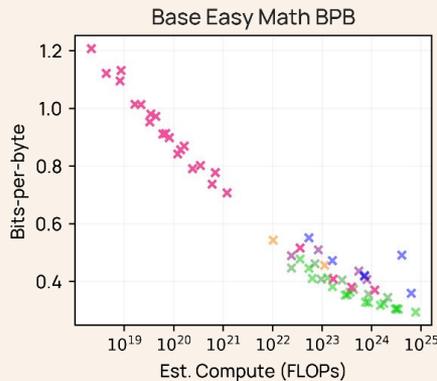
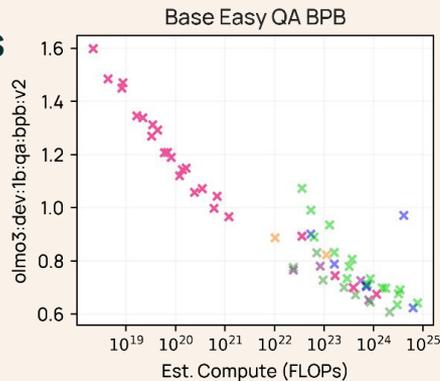
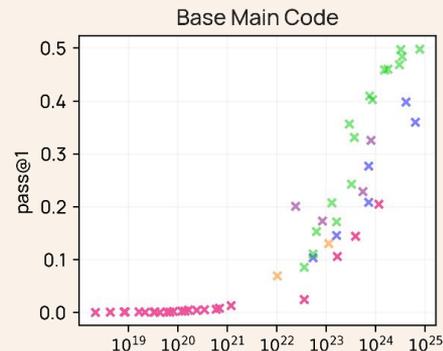
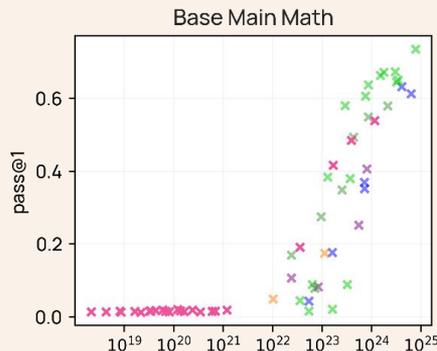
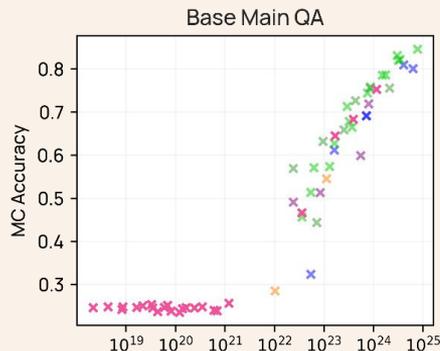
**Noise at large scales**  
(e.g. midtraining)

## ② Small-scale metrics:



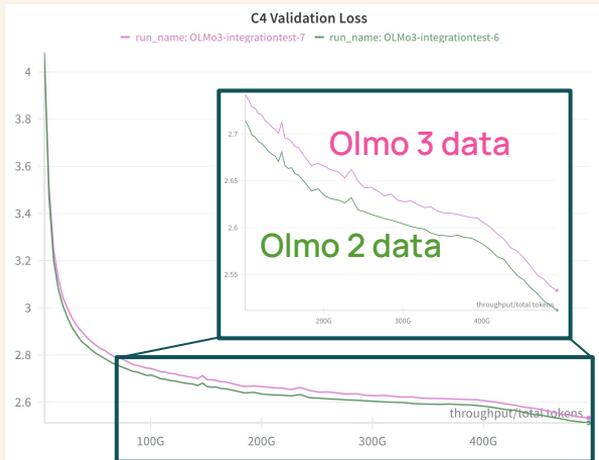
## ② Small-scale metrics:

Separate decision-making suite using BPB over continuations

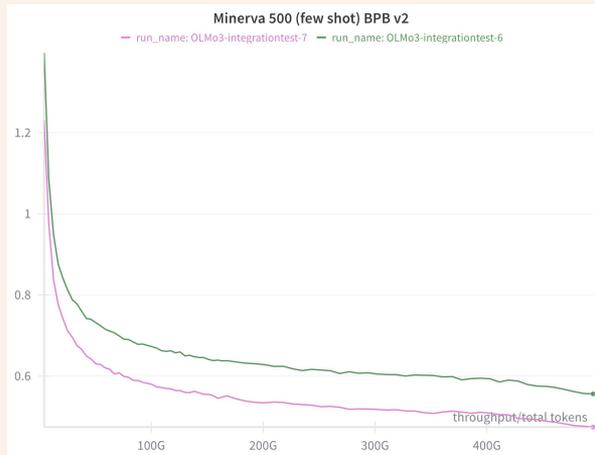


## ② Small-scale metrics:

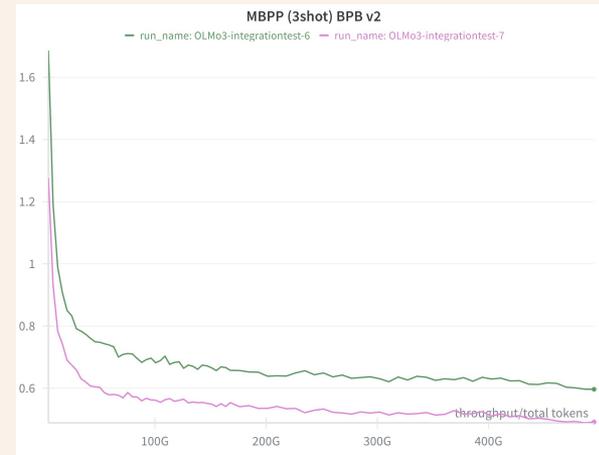
### C4 Loss



### Math 500 BPB



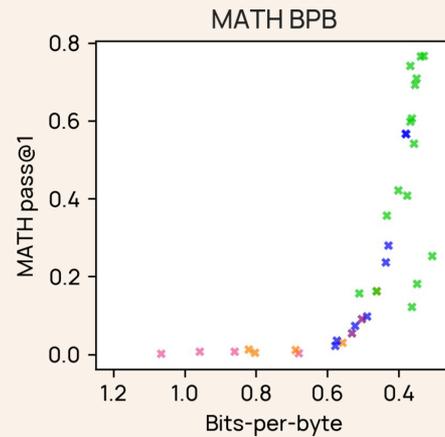
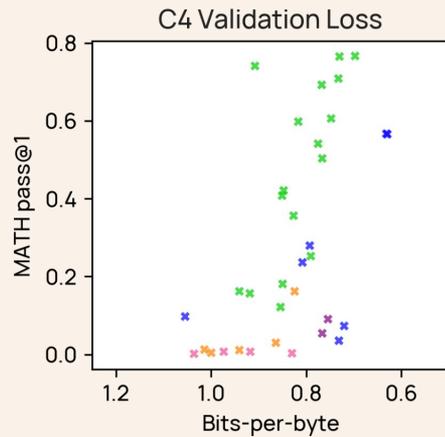
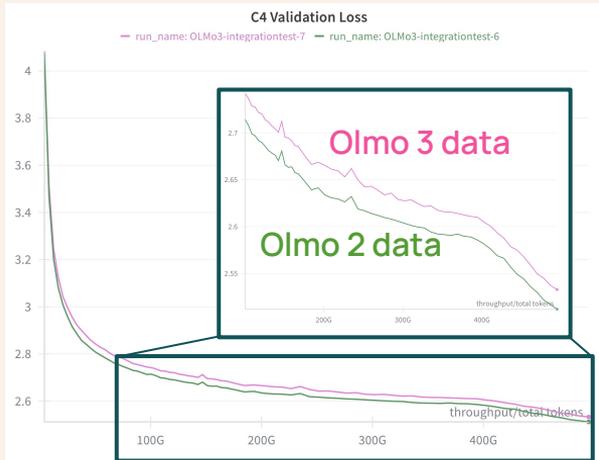
### MBPP BPB (Python Coding)



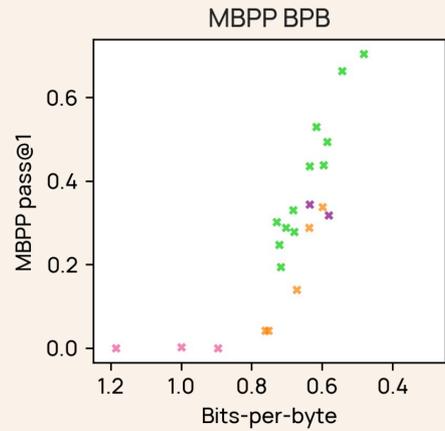
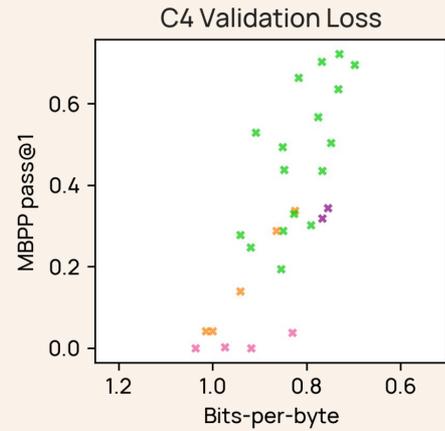
At 7B 500B tokens, **Olmo 3 data** has a worse fit than **Olmo 2 data** on C4 loss, but better on math, code text

## ② Small-scale metrics:

### C4 Loss

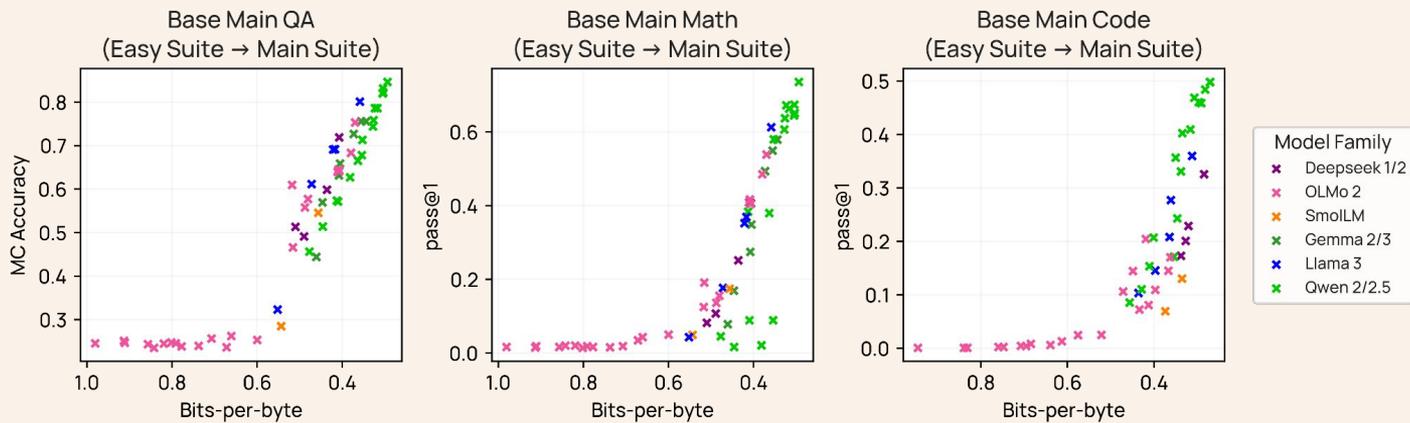


- Model Family
- ✖ Deepseek 1/2
  - ✖ OLMo 2
  - ✖ SmoLM
  - ✖ Gemma 2/3
  - ✖ Llama 3
  - ✖ Qwen 2/2.5

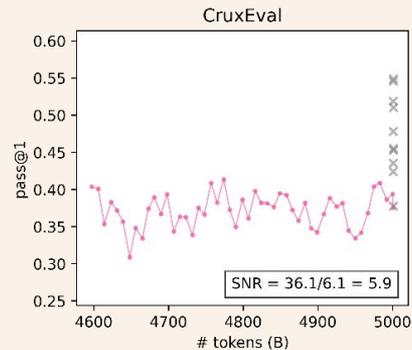
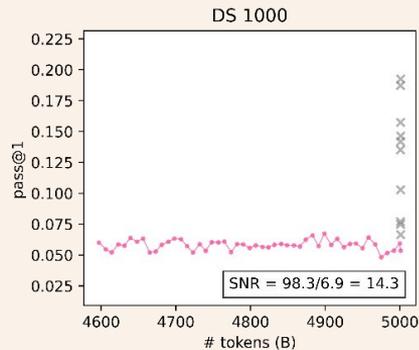
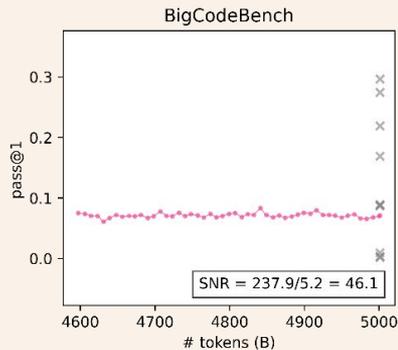
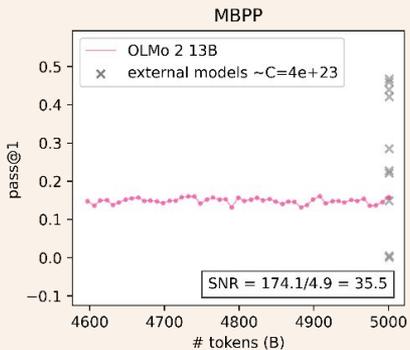
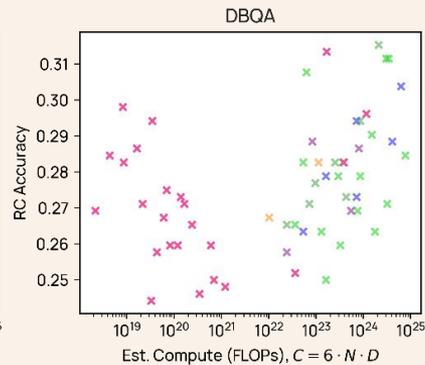
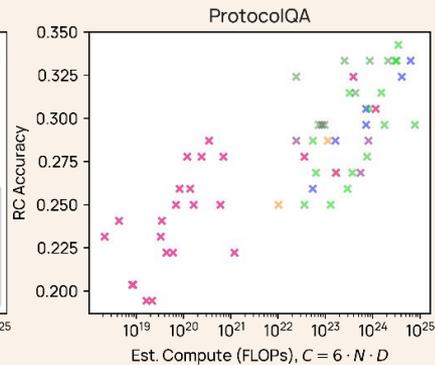
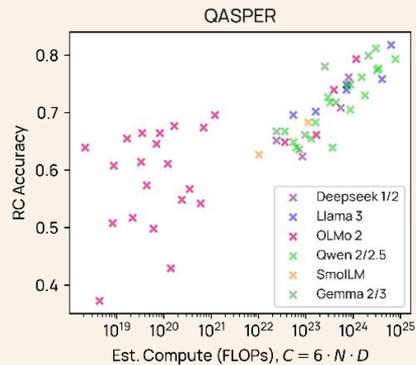
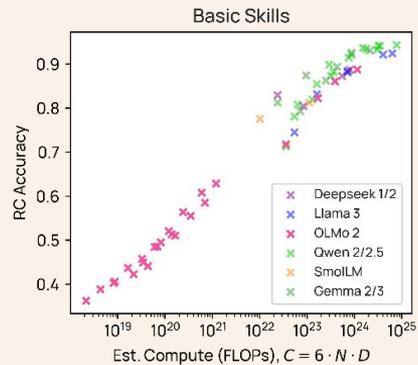


- Model Family
- ✖ Deepseek 1/2
  - ✖ OLMo 2
  - ✖ SmoLM
  - ✖ Gemma 2/3
  - ✖ Llama 3
  - ✖ Qwen 2/2.5

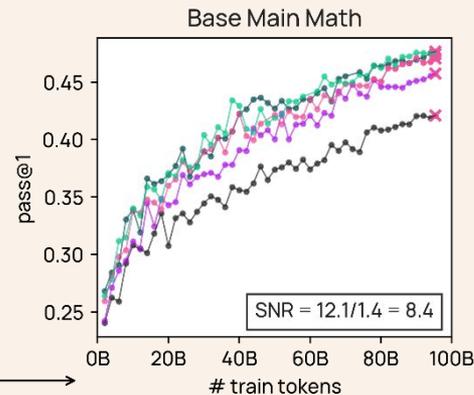
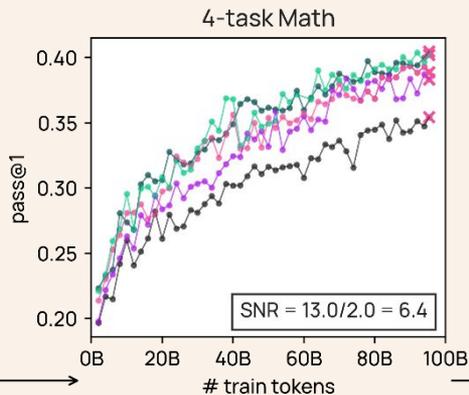
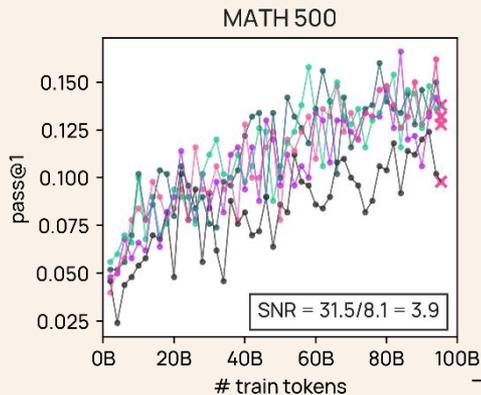
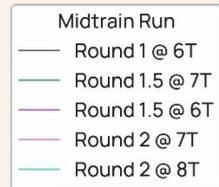
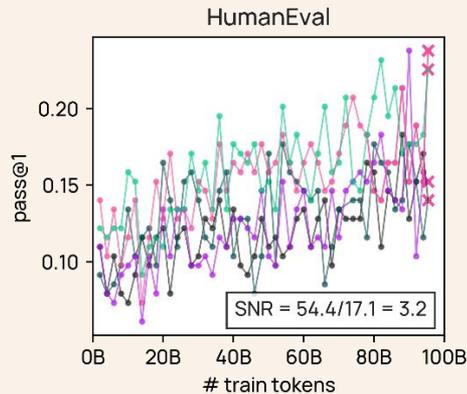
## ② Small-scale metrics:



# 3 SNR in Olmo 3 Eval

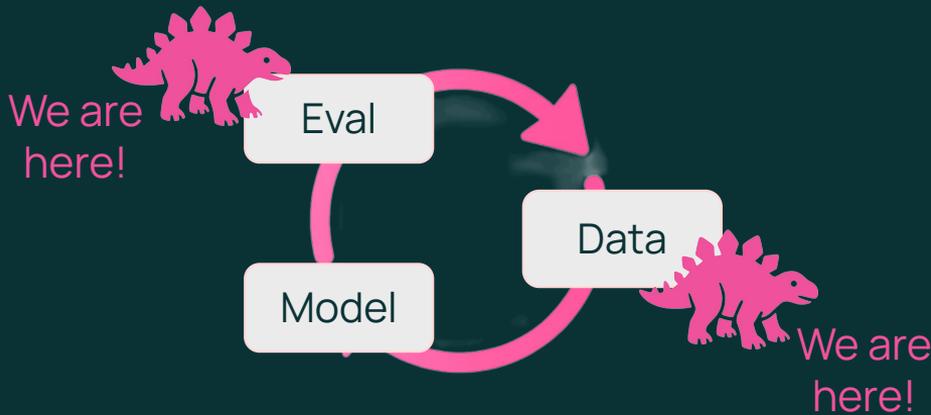


# 3 SNR in Olmo 3 Eval



Aggregate into multi-task average, filter noisy tasks

Tune generation configuration



# Olmix: A Framework for Data Mixing Throughout LM Development

Mayee F. Chen<sup>1,2</sup> Tyler Murray<sup>1</sup> David Heineman<sup>1</sup> Matt Jordan<sup>1</sup> Hannaneh Hajishirzi<sup>1,3</sup>  
 Christopher Ré<sup>2</sup> Luca Soldaini<sup>1</sup> Kyle Lo<sup>1,3</sup>

<sup>1</sup>Allen Institute for AI <sup>2</sup>Stanford University <sup>3</sup>University of Washington

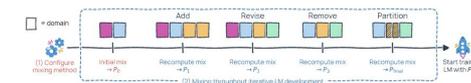
Code: [Olmix](#) Data: [Olmix](#) Contact: [mfchen@cs.stanford.edu](mailto:mfchen@cs.stanford.edu) [{lucas,kyle}@allenai.org](mailto:{lucas,kyle}@allenai.org)

## Abstract



Data mixing—determining the ratios of data from different domains—is a first-order concern for training language models (LMs). While existing mixing methods show promise, they fall short when applied during real-world LM development. We present OLMIX, a framework that addresses two such challenges. First, the configuration space for developing a mixing method is not well understood—design choices across existing methods lack justification or consensus and overlook practical issues like data constraints. We conduct a comprehensive empirical study of this space, identifying which design choices lead to a strong mixing method. Second, in practice, the domain set evolves throughout LM development as datasets are added, removed, partitioned, and revised—a problem setting largely unaddressed by existing works, which assume fixed domains. We study how to efficiently recompute the mixture after the domain set is updated, leveraging information from past mixtures. We introduce mixture reuse, a mechanism that reuses existing ratios and recomputes ratios only for domains affected by the update. Over a sequence of five domain-set updates mirroring real-world LM development, mixture reuse matches the performance of fully recomputing the mix after each update with 74% less compute and improves over training without mixing by 11.6% on downstream tasks.

## 1 Introduction

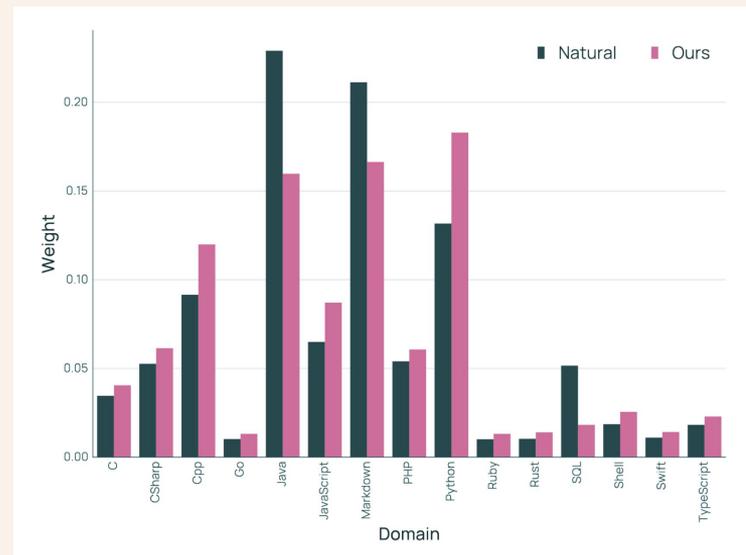
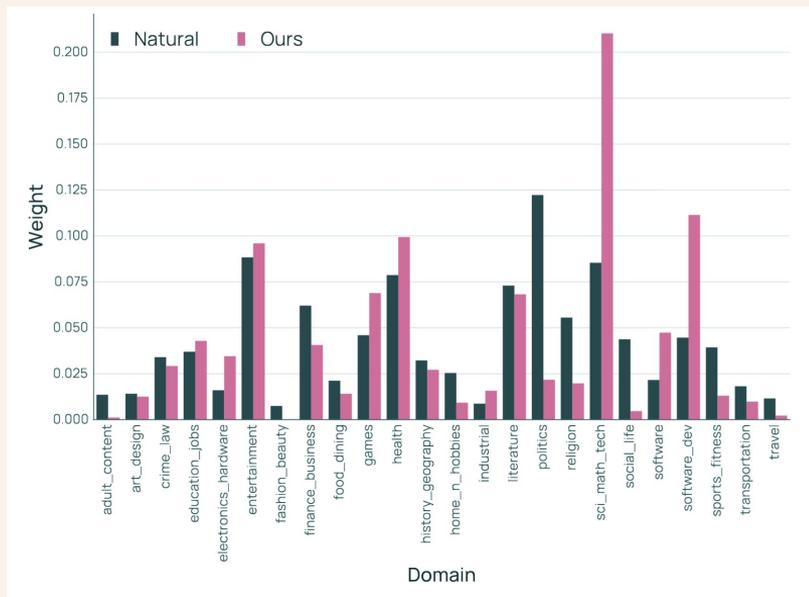


**Figure 1** Two problems with data mixing encountered during LM development: (1) How to best configure your mixing method? (2) How to efficiently mix under evolving domain sets?

Modern language models (LMs) are trained on datasets composed of many domains, such as web text, code, and PDFs. The composition of these domains is crucial for strong downstream performance, making data mixing a first-order component of LM development (Grazziotin et al., 2024; Chen et al., 2024; Ohno et al., 2025). However, finding a good mix is non-trivial: practitioners often resort to manual weight tuning or exhaustive search, which can require many training runs—possibly thousands of GPU hours—to assess performance. This has resulted in a growing literature on data mixing methods that aim to find strong mixtures systematically with less compute (Liu et al., 2023; Fan et al., 2024; Chen et al., 2025). Many mixing methods that achieve promising results follow a common *offline mixing schema* (Liu et al., 2025b,a; Ye et al., 2025) that consists of three steps: 1. train a set of smaller proxy models on different mixtures (a “swarm”), 2.

# What data mixing?

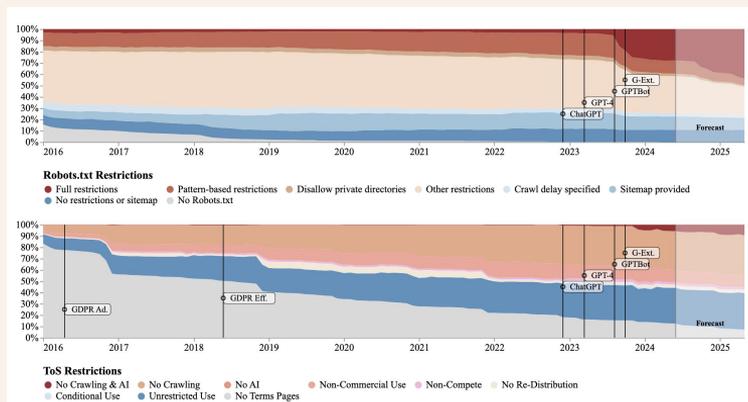
Olmo 3 (2025)



# Why data mixing?

## Token scale

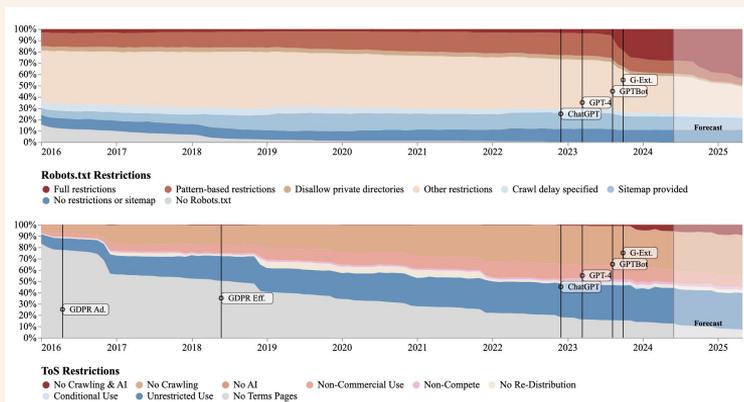
### Consent in Crisis (Neurips 2024)



# Why data mixing?

## Token scale

### Consent in Crisis (Neurips 2024)



## Proportion / Ratios

### Gopher (2021)

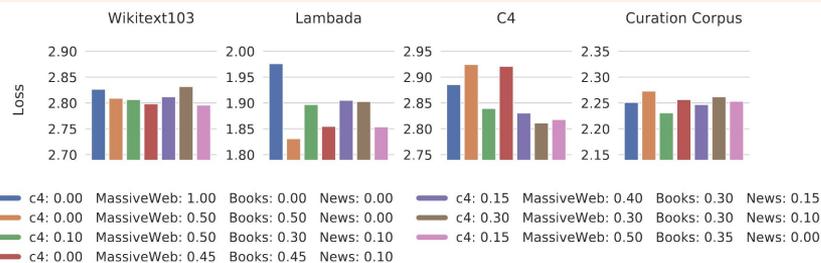
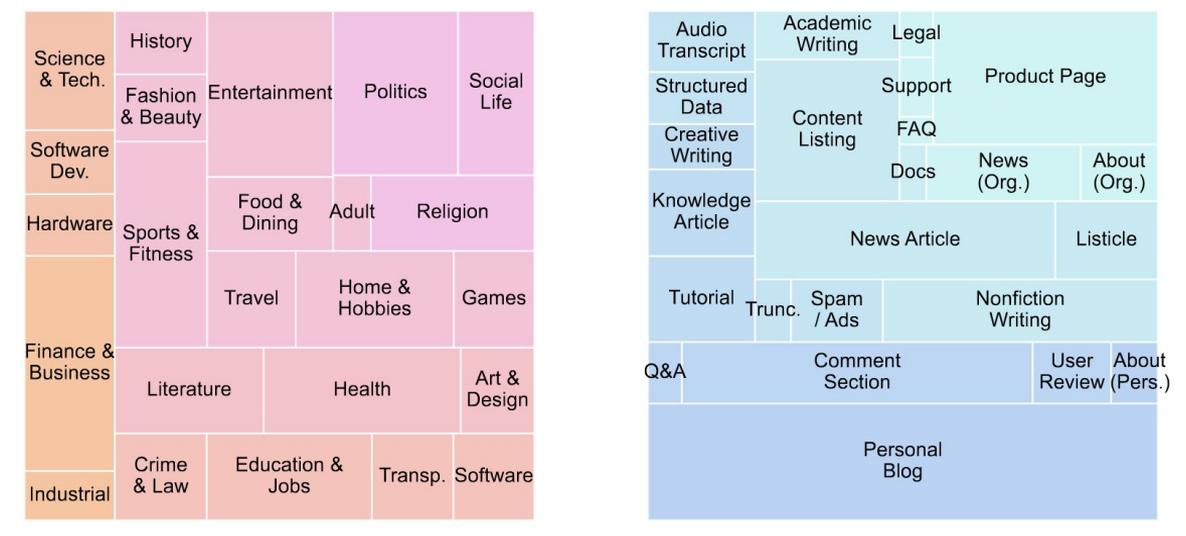


Figure A4 | Downstream performance for different *MassiveText* subset sampling weights. The configuration (in green) with 10% C4, 50% *MassiveWeb*, 30% Books, and 10% News performs well across all tasks and achieves the best performance on Curation Corpus—we therefore choose those sampling weights in our main *Gopher* training experiments.

# Problem: Search over data mixtures is combinatorial

WebOrganizer (ICML 2024)



# How data mixing?

## REGMIX: DATA MIXTURE AS REGRESSION FOR LANGUAGE MODEL PRE-TRAINING

**Qian Liu**<sup>1\*</sup> **Xiaosen Zheng**<sup>2\*</sup> **Niklas Muennighoff**<sup>3,4</sup> **Guangtao Zeng**<sup>5</sup>  
**Longxu Dou**<sup>1</sup> **Tianyu Pang**<sup>1</sup> **Jing Jiang**<sup>2</sup> **Min Lin**<sup>1</sup>  
<sup>1</sup>Sea AI Lab <sup>2</sup>SMU <sup>3</sup>Contextual AI <sup>4</sup>Stanford University <sup>5</sup>SUTD  
liuqian.sea@gmail.com; xszheng.2020@phdcs.smu.edu.sg

## BiMIX: BIVARIATE DATA MIXING LAW FOR LANGUAGE MODEL PRETRAINING

**Ce Ge & Zhijian Ma**  
Alibaba Group  
Beijing, China  
{gece.gc, zhijian.mzj}@alibaba-inc.com

**Dayuan Chen**  
Alibaba Group  
Hangzhou, China  
daoyuanchen.cdy@alibaba-inc.com

**Yaliang Li\* & Bolin Ding**  
Alibaba Group  
Bellevue, USA  
{yaliang.li, bolin.ding}@alibaba-inc.com

## Data Mixing Laws: Optimizing Data Mixtures by Predicting Language Modeling Performance

**Jiasheng Ye**<sup>1,\*</sup> **Peiju Liu**<sup>1,\*</sup> **Tianxiang Sun**<sup>1</sup> **Jun Zhan**<sup>1</sup> **Yunhua Zhou**<sup>2, †</sup> **Xipeng Qiu**<sup>1, †</sup>  
{jsye23, pjliu23}@m.fudan.edu.cn zhouyunhua@pjlab.org.cn xpqiu@fudan.edu.cn  
<sup>1</sup>Fudan University <sup>2</sup>Shanghai AI Laboratory

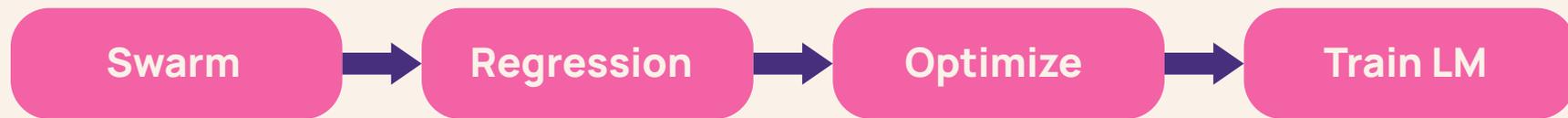
---

## Nemotron-CLIMB: CLustering-based Iterative Data Mixture Bootstrapping for Language Model Pre-training

---

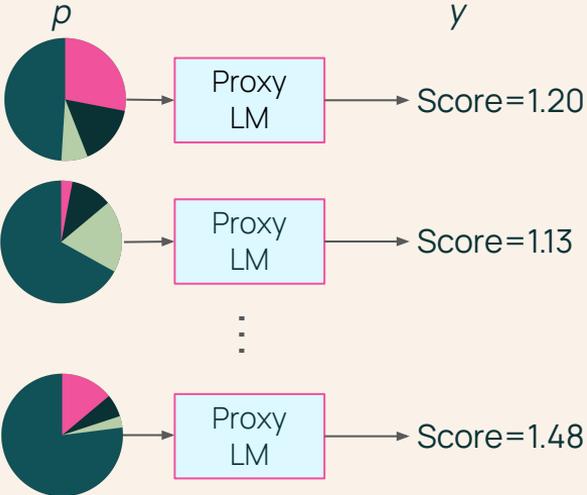
**Shizhe Diao**<sup>1</sup>, **Yu Yang**<sup>1</sup>, **Yonggan Fu**<sup>1</sup>, **Xin Dong**<sup>1</sup>, **Dan Su**<sup>1</sup>, **Markus Kliegl**<sup>1</sup>, **Zijia Chen**<sup>1</sup>,  
**Peter Belcak**<sup>1</sup>, **Yoshi Suhara**<sup>1</sup>, **Hongxu Yin**<sup>1</sup>, **Mostafa Patwary**<sup>1</sup>, **Yingyan (Celine) Lin**<sup>2</sup>,  
**Jan Kautz**<sup>1</sup>, **Pavlo Molchanov**<sup>1</sup>  
<sup>1</sup>NVIDIA <sup>2</sup>Georgia Institute of Technology

# How data mixing?



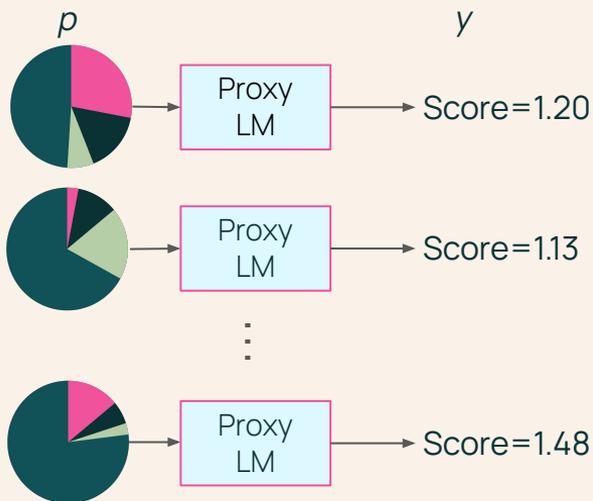
# How data mixing?

1. **Swarm:** Train  $K$  small models with randomly sampled mixtures  $p$



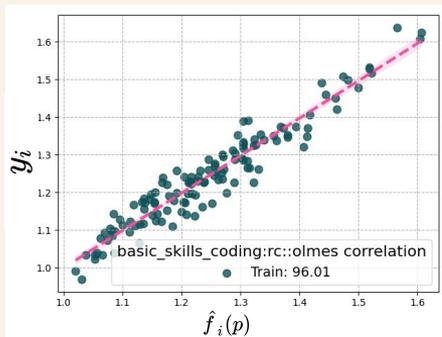
# How data mixing?

1. **Swarm:** Train  $K$  small models with randomly sampled mixtures  $p$



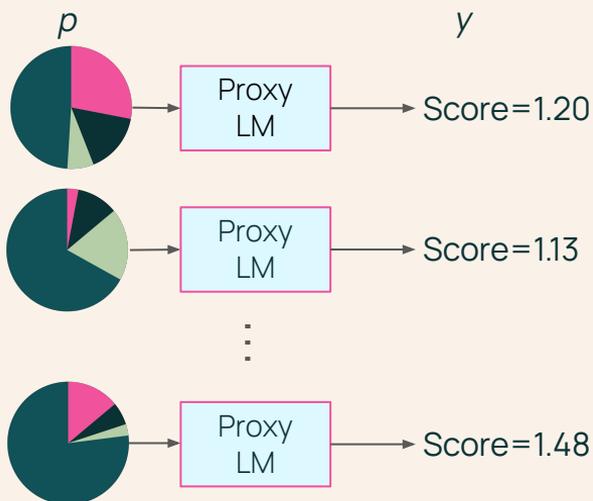
2. **Regression:** Fit function to predict LM performance given mixture  $p$

$$\hat{f}_i(p) \approx y_i$$

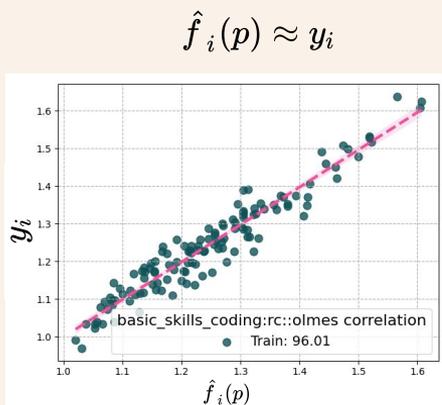


# How data mixing?

1. **Swarm:** Train  $K$  small models with randomly sampled mixtures  $p$

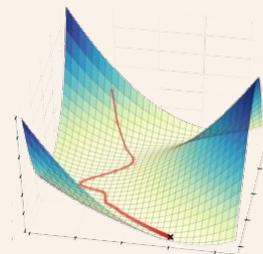
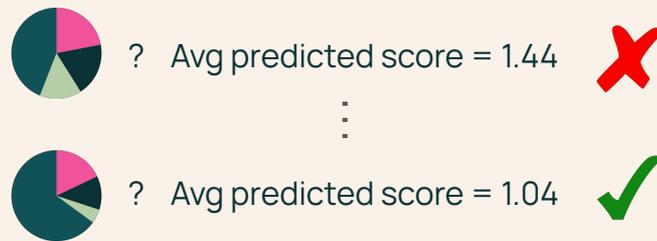


2. **Regression:** Fit function to predict LM performance given mixture  $p$



3. **Optimize:** Use fit function to solve for optimal mix  $p^*$

$$\underset{p \in \Delta^{m-1}}{\text{minimize}} \frac{1}{n} \sum_{i=1}^n \hat{f}_i(p)$$



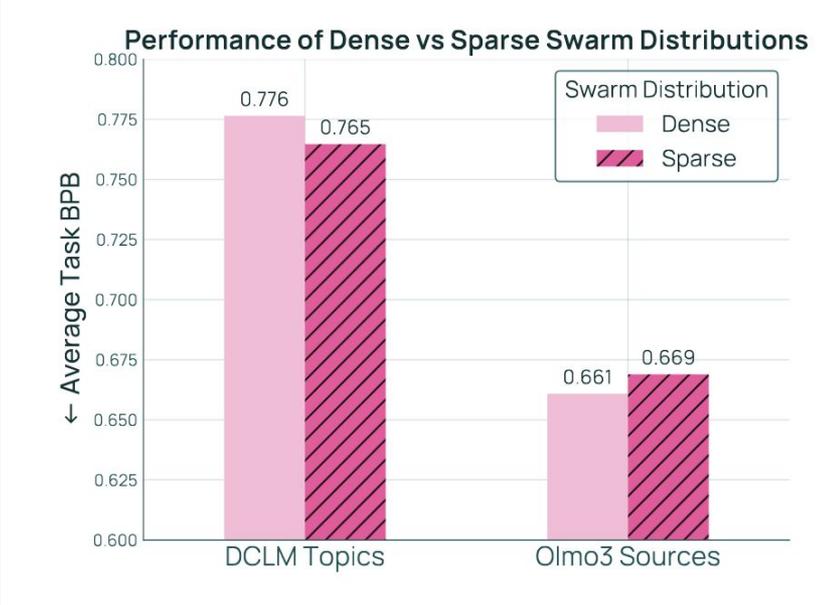
# How data mixing?

Design Choice	RegMix (Liu et al., 2025a)	DML (Ye et al., 2025)	AutoScale (Kang et al., 2025)	BiMix (Ge et al., 2025b)	ADMIRE-BayesOpt (Chen et al., 2025b)	CLIMB (Diao et al., 2025)
<b>Swarm Construction</b>						
Proxy model size	1M	70, 160, 305, 410M	Target	280M	1M, 60M	350M
Swarm size (vs $m$ domains)	512 ( $m = 17$ )	20 ( $m = 7$ )	$2m + 1$	4	101 ( $m = 17$ )	112 ( $m = 21$ )
Swarm distribution	Dirichlet with natural prior	Exponential grid	Exponential grid	Entropy-weighted	Dynamic	Dirichlet with natural prior
<b>Regression Model</b>						
Regression model family	LightGBM	Log-Linear	Power Law	Power Law	Gaussian Process	LightGBM
Regression granularity	Aggregated	Aggregated	Per-Task	Per-Task	Aggregated	Aggregated
<b>Mixture Optimization</b>						
Data repetition constraints	No	No	No	No	No	No
Optimization solver	Search	Search	Gradient Descent	Exact Solver	Search	Search

# Problem 1: No “standard” config

Design Choice	RegMix (Liu et al., 2025a)	DML (Ye et al., 2025)	AutoScale (Kang et al., 2025)	BiMix (Ge et al., 2025b)	ADMIRE-BayesOpt (Chen et al., 2025b)	CLIMB (Diao et al., 2025)
<b>Swarm Construction</b>						
Proxy model size	1M	70, 160, 305, 410M	Target	280M	1M, 60M	350M
Swarm size (vs $m$ domains)	512 ( $m = 17$ )	20 ( $m = 7$ )	$2m + 1$	4	101 ( $m = 17$ )	112 ( $m = 21$ )

# Problem 1: No “standard” config



# Problem 1: No “standard” config

## Swarm construction:

*RQ1* What is the smallest proxy model size (the number of parameters,  $S_{\text{small}}$ ) such that decision-making generalizes to larger target models?

*RQ2* How many proxy runs  $K$  do we need to learn a good mix on  $m$  domains?

*RQ3* How should we specify the distribution  $\mathcal{P}$  to sample the mixes for the proxy runs?

## Regression model:

*RQ4* Is there an optimal family of regression models ( $\mathcal{F}$ ) for predicting mix performance?

*RQ5* At what granularity should we fit the regression models in order to construct  $\hat{f}(p)$ ?

## Mix optimization:

*RQ6* How do we mix under finite data constraints?

*RQ7* How do we solve the optimization problem?

# Problem 2: Data changes during LM development

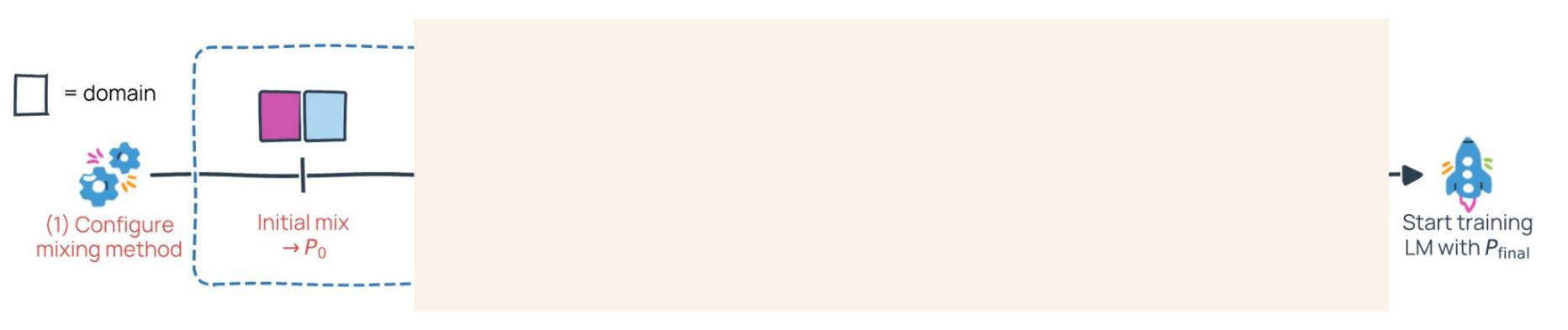
 = domain



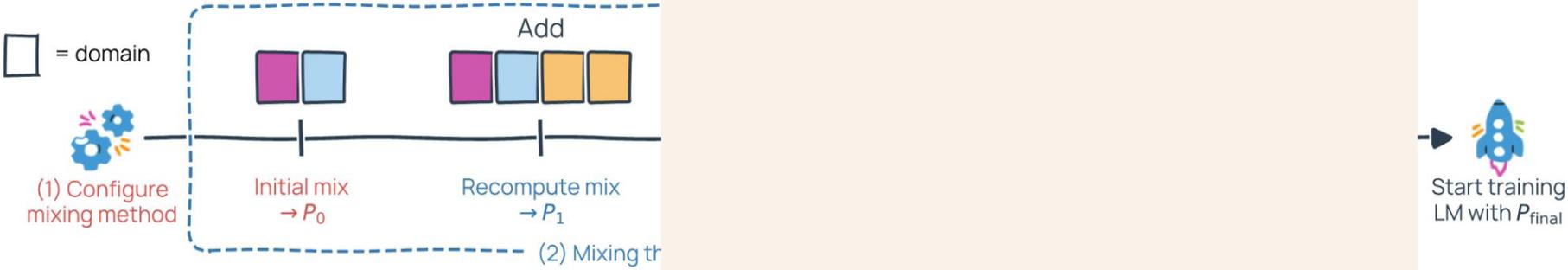
(1) Configure mixing method



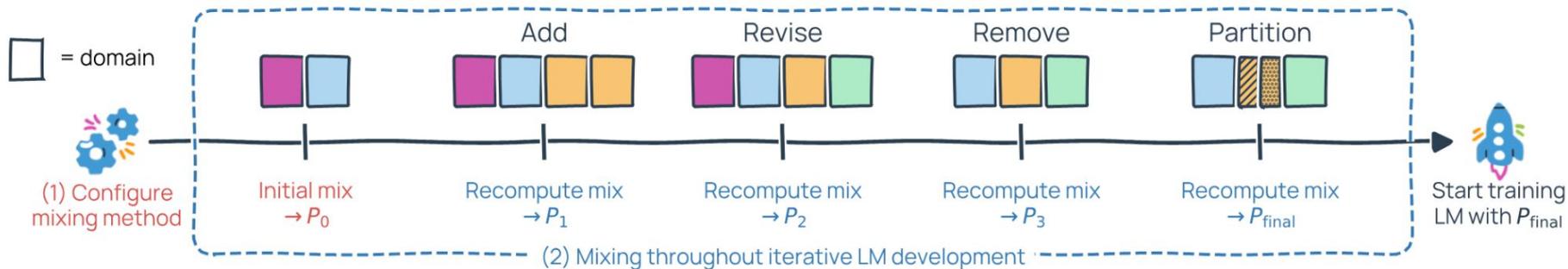
# Problem 2: Data changes during LM development



# Problem 2: Data changes during LM development



# Problem 2: Data changes during LM development



# Problem 2: Data changes during LM development

DCLM (Neurips 2025)

## 4.5 Dataset mixing

Often, Common Crawl (CC) is combined with other data sources that are considered high-quality [63, 70, 168, 170] (e.g., Wikipedia, StackExchange, and peS2o [156]). Since DCLM participants can include additional data sources in our mixing track, we examined the potential benefits of adding high-quality sources to training sets derived from Common Crawl only. We compare a model trained on 100% filtered CC data to models trained with the mixing ratios from Llama 1 and RedPajama: 67% CC, and 33% from Wikipedia, Books, Stack exchange, arXiv, and Github. For the CC component, we consider different variants: a subset of our DCLM-BASELINE, RedPajama's CC portion, RefinedWeb, and C4. The results in Table 6 show that mixing improves performance for the lower-performing CC subsets (C4, RedPajama-CC, and RefinedWeb). In the case of DCLM-BASELINE however, mixing actually hurts performance on average, which suggests it can be counterproductive given performant filtering. For additional mixing results, see Appendix M.

# Problem 2: Data changes during LM development

Olmo 3(2025)

## Add + Remove

- DCLM → Olmo 3 Common Crawl
- Stack v2 → StackEdu
- OpenWebMath → FineMath

## Transform

- DCLM quality filter → FineWebEdu quality filter

## Partition

- S2ORC PDFs → Classify by field of study

# Problem 2: Data changes during LM development

Olmo 3(2025)

## Add + Remove

- DCLM → Olmo 3 Common Crawl
- Stack v2 → StackEdu
- OpenWebMath → FineMath

## Transform

- DCLM quality filter → FineWebEdu quality filter

## Partition

- S2ORC PDFs → Classify by field of study



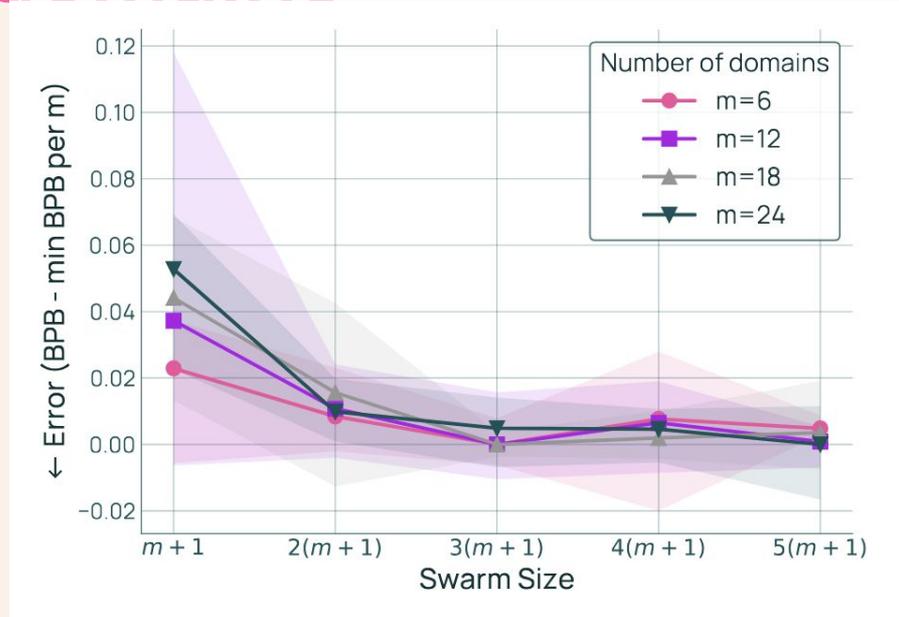
# Olmix: Data mixing over the LM development cycle

# Finding 1: Find smallest model size where performance ranking correlates well with larger models

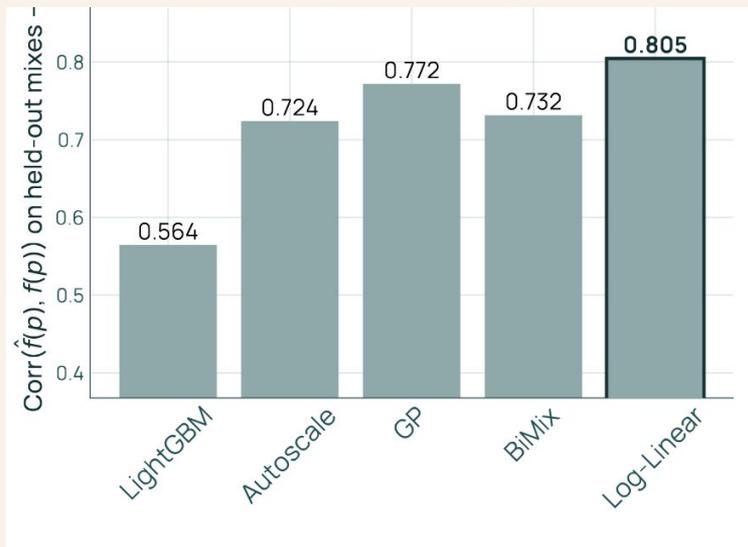
Regmix paper →



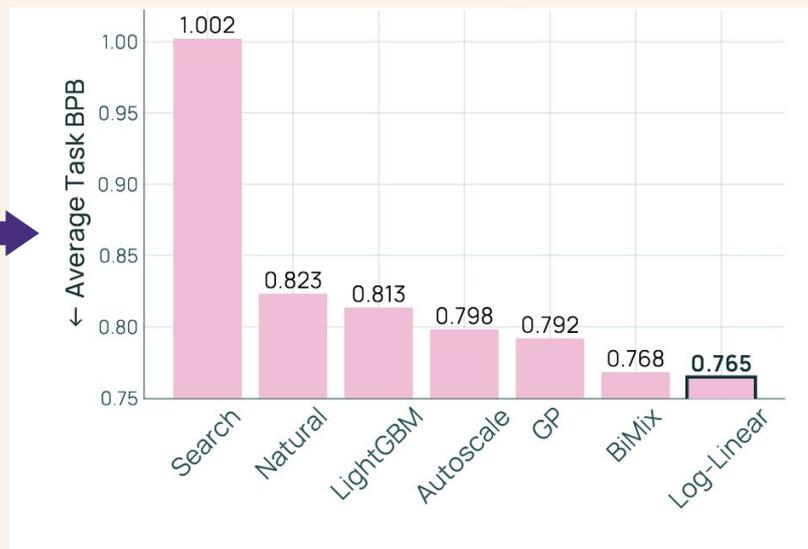
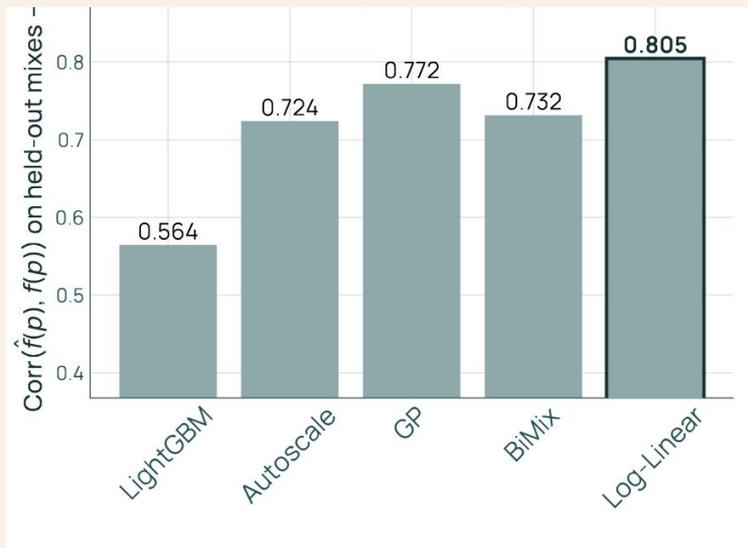
# Finding 2: Swarm size scales linearly with number of domains



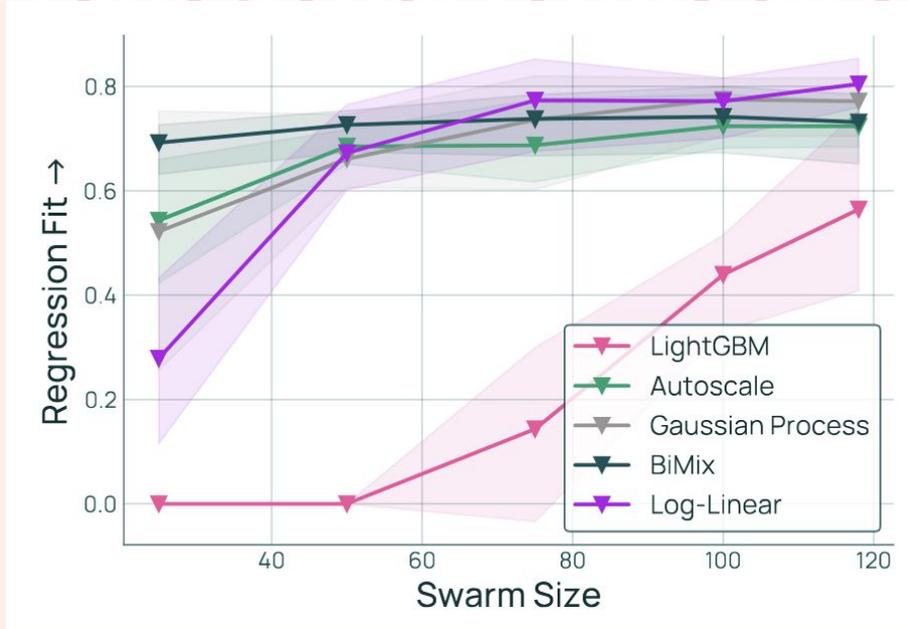
# Finding 3: Regression fit is decently indicative of downstream performance



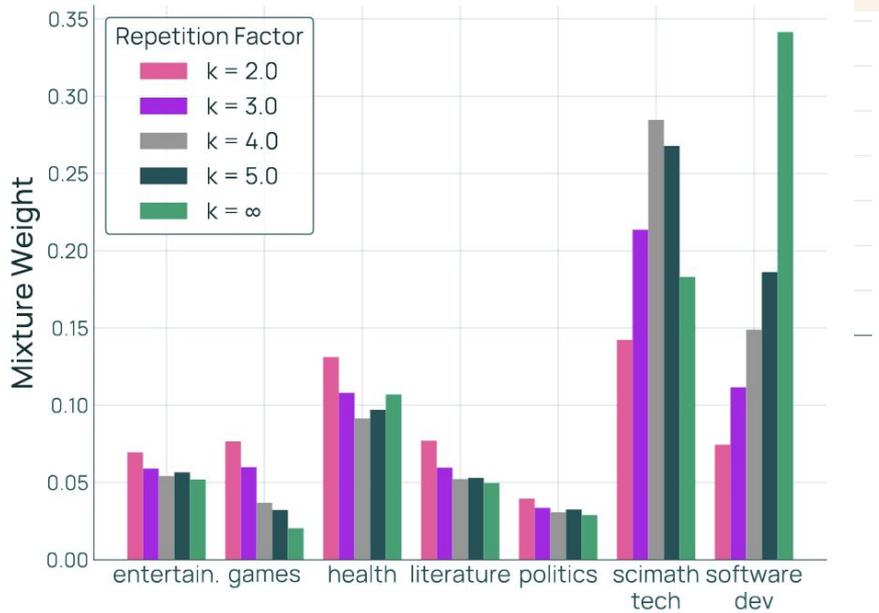
# Finding 3: Regression fit is decently indicative of downstream performance



# Finding 3b: Given sufficient compute, regression functional forms all similar



# Finding 4: Data mixing dramatically sensitive if forced to repeat data



# Finding 5: Historical data can be optimally reused to minimize experimental cost over time

Swarm

$$p_1 = [0.3, 0.7]$$

$$p_2 = [0.4, 0.6]$$

$$p_3 = [0.8, 0.2]$$

...

$$p_N = [0.1, 0.9]$$



$$p^* = [0.66, 0.34]$$



# Finding 5: Historical data can be optimally reused to minimize experimental cost over time

Swarm

$$p_1 = [0.3, 0.7]$$

$$p_2 = [0.4, 0.6]$$

$$p_3 = [0.8, 0.2]$$

...

$$p_N = [0.1, 0.9]$$

Add a  
domain



$$p^* = [0.66, 0.34]$$



# Finding 5: Historical data can be optimally reused to minimize experimental cost over time



**Swarm**

$$p_1 = [0.3, 0.7]$$

$$p_2 = [0.4, 0.6]$$

$$p_3 = [0.8, 0.2]$$

...

$$p_N = [0.1, 0.9]$$



$$p^* = [0.66, 0.34]$$

**Add a  
domain**

**Full recomputation**

$$p_1 = [0.3, 0.5, 0.2]$$

$$p_2 = [0.4, 0.4, 0.2]$$

$$p_3 = [0.8, 0.1, 0.1]$$

...

$$p_N = [0.1, 0.5, 0.4]$$



$$p^*$$



# Finding 5: Historical data can be optimally reused to minimize experimental cost over time

Swarm

$$p_1 = [0.3, 0.7, 0.0]$$

$$p_2 = [0.4, 0.6, 0.0]$$

$$p_3 = [0.8, 0.2, 0.0]$$

...

$$p_N = [0.1, 0.9, 0.0]$$



~~$$p^* = [0.66, 0.34]$$~~

Add a  
domain

“Swarm” reuse

$$p_1 = [0.3, 0.5, 0.2]$$

$$p_2 = [0.4, 0.4, 0.2]$$

$$p_3 = [0.8, 0.1, 0.1]$$

...

$$p_N = [0.1, 0.5, 0.4]$$



$$p^*$$



# Finding 5: Historical data can be optimally reused to minimize experimental cost over time

Swarm

~~$p_1 = [0.3, 0.7]$   
 $p_2 = [0.4, 0.6]$   
 $p_3 = [0.8, 0.2]$   
...  
 $p_N = [0.1, 0.9]$~~



$p^* = [0.66, 0.34]$

Add a domain

“Mix” reuse

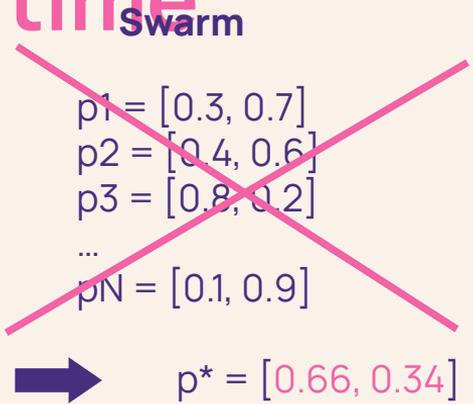
$p_1 = [0.3, 0.7]$   
 $p_2 = [0.4, 0.6]$   
 $p_3 = [0.8, 0.2]$   
...  
 $p_N = [0.1, 0.9]$



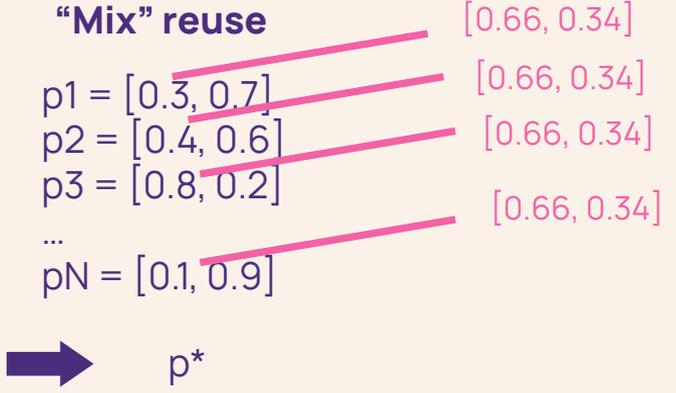
$p^*$



# Finding 5: Historical data can be optimally reused to minimize experimental cost over time

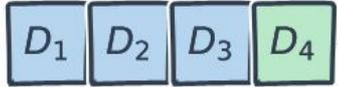


Add a domain

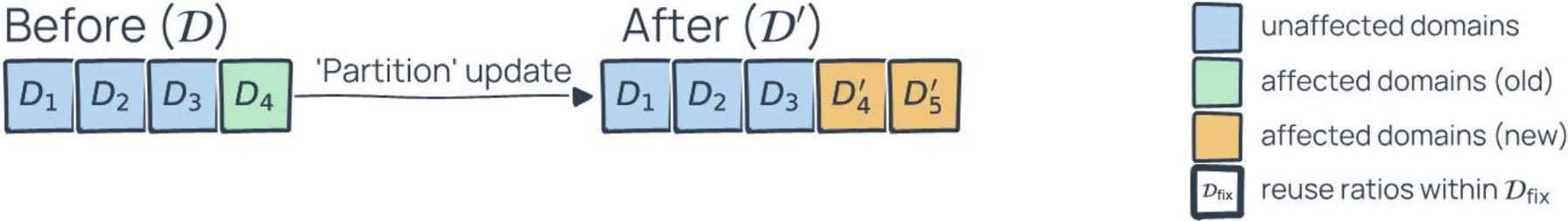


# Finding 5: Mixture reuse

Before ( $\mathcal{D}$ )

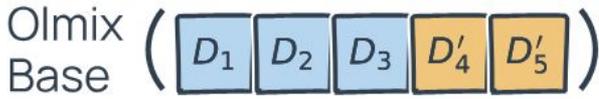


# Finding 5: Mixture reuse



# Finding 5: Mixture reuse

Full recomputation



Higher cost



Higher performance

Potentially lower performance



# Finding 5: Mixture reuse

Full recomputation

Olmix Base (  $D_1$   $D_2$   $D_3$   $D'_4$   $D'_5$  )

Partial mixture reuse

$D_{\text{fix}} = D_1 D_3$

Olmmix Base (  $D_{\text{fix}}$   $D_2$   $D'_4$   $D'_5$  )

Higher cost

Higher performance

Potentially lower performance

# Finding 5: Mixture reuse

Full recomputation

Olmix Base (  $D_1$   $D_2$   $D_3$   $D'_4$   $D'_5$  )

Partial mixture reuse

$D_{\text{fix}} = D_1 D_3$

Olmix Base (  $D_{\text{fix}}$   $D_2$   $D'_4$   $D'_5$  )

Full mixture reuse

$D_{\text{fix}} = D_1 D_2 D_3$

Olmix Base (  $D_{\text{fix}}$   $D'_4$   $D'_5$  )

Higher cost

Lower cost

Higher performance

Potentially lower performance



# Finding 5: Historical data can be optimally reused to minimize experimental cost over time



# Finding 5: Historical data can be optimally reused to minimize experimental cost over time

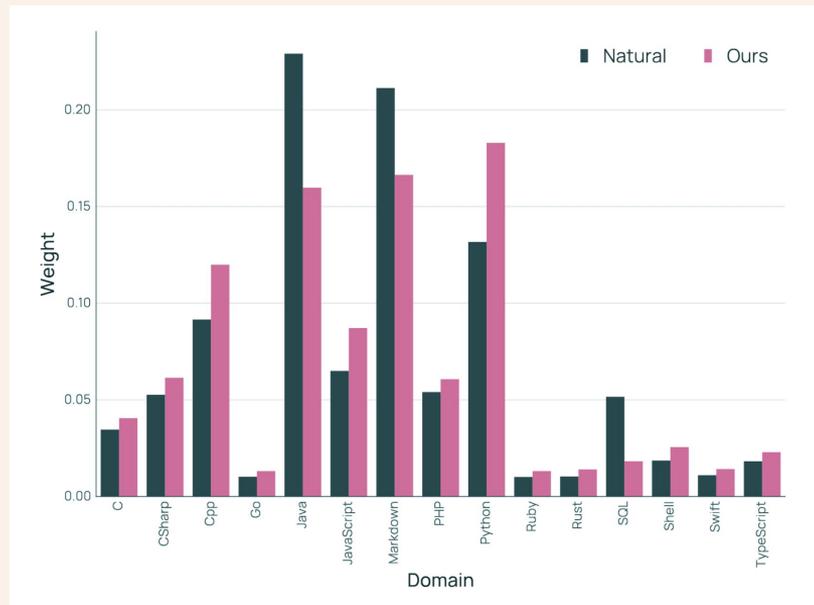
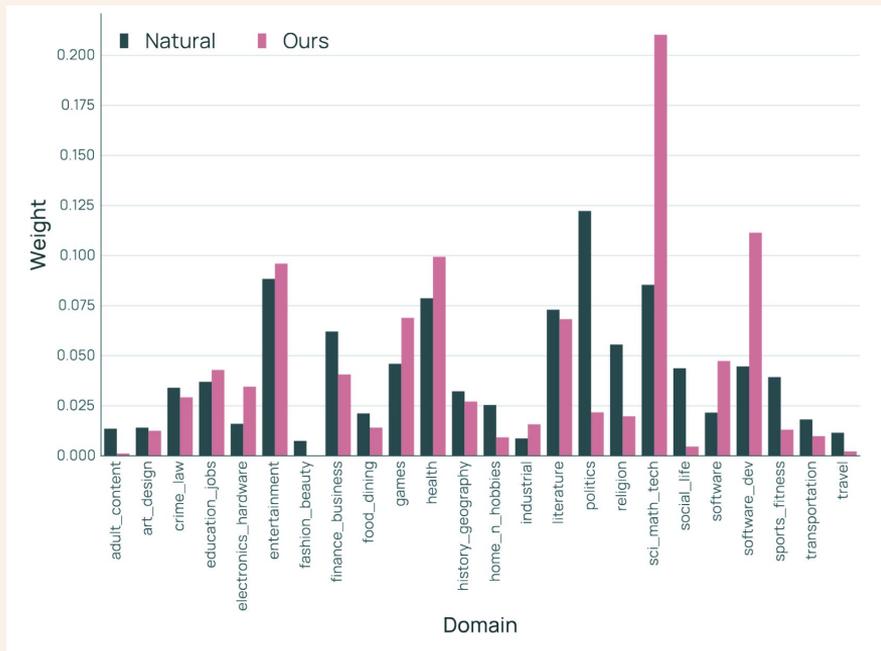
**Theorem 4.1** (Performance gap bound). *The performance gap is bounded by*

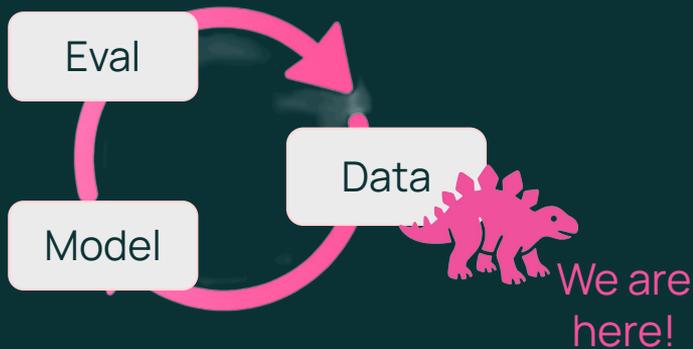
$$F(q^*(\tilde{p}_{\mathcal{D}_{fix}})) - F(q^*) \leq C \|\tilde{p}_{\mathcal{D}_{fix}} - q_{\mathcal{D}_{fix}}^*\|,$$
$$C = \mathcal{E}(\|\tilde{p}_{\mathcal{D}_{fix}} - q_{\mathcal{D}_{fix}}^*\|)(\kappa(\alpha_{fix}, \alpha_{comp}) + \|\alpha_{fix}\|)$$

**Theorem 4.2.** *Assume that  $\tilde{p}$  is the solution to (2) on  $\mathcal{D}$ . When new domains are added, the reuse gap is bounded by*

$$\|\tilde{p}_{\mathcal{D}_{fix}} - q_{\mathcal{D}_{fix}}^*\| \leq \mathcal{E}(1 - \rho^*)\kappa(\alpha_{fix}, \alpha_{comp})(1 - \rho^*)$$

# Data mixing





## Olmo 3

Olmo Team\*

Allyson Ettlinger<sup>\*1</sup> Amanda Bertsch<sup>\*1,3</sup> Bailey Kuehl<sup>\*1</sup> David Graham<sup>\*1</sup>  
 David Heineman<sup>\*1</sup> Dirk Groeneveld<sup>\*1</sup> Faeze Brahman<sup>\*1</sup> Finbarr Timbers<sup>\*1</sup>  
 Hamish Ivison<sup>\*1,2</sup> Jacob Morrison<sup>\*1,2</sup> Jake Poznanski<sup>\*1</sup> Kyle Lo<sup>\*1,2</sup> Luca Soldaini<sup>\*1</sup>  
 Matt Jordan<sup>\*1</sup> Mayee Chen<sup>\*1,4</sup> Michael Noukhovitch<sup>\*1,5,6</sup> Nathan Lambert<sup>\*1</sup>  
 Pete Walsh<sup>\*1</sup> Pradeep Dasigi<sup>\*1</sup> Robert Berry<sup>\*1</sup> Saumya Malik<sup>\*1</sup> Saurabh Shah<sup>\*1</sup>  
 Scott Geng<sup>\*1,2</sup> Shane Arora<sup>\*1</sup> Shashank Gupta<sup>\*1</sup> Taira Anderson<sup>\*1</sup> Teng Xiao<sup>\*1</sup>  
 Tyler Murray<sup>\*1</sup> Tyler Romero<sup>\*1</sup> Victoria Graf<sup>\*1,2</sup>

Akari Asai<sup>1,3</sup> Akshita Bhagia<sup>1</sup> Alexander Wettig<sup>1</sup> Alisa Liu<sup>2</sup> Aman Rangapur<sup>1</sup>  
 Chloe Anastasiades<sup>1</sup> Costa Huang<sup>1</sup> Dustin Schwenk<sup>1</sup> Harsh Trivedi<sup>1</sup> Ian Magnusson<sup>1,2</sup>  
 Jaron Lochner<sup>1</sup> Jiacheng Liu<sup>1</sup> Lester James V. Miranda<sup>1</sup> Maarten Sap<sup>1,3</sup> Malia Morgan<sup>1</sup>  
 Michael Schmitz<sup>1</sup> Michal Querquin<sup>1</sup> Michael Wilson<sup>1</sup> Regan Huff<sup>1</sup> Ronan Le Bras<sup>1</sup>  
 Rui Xin<sup>2</sup> Rulin Shao<sup>2</sup> Sam Sijmonsberg<sup>2</sup> Shannon Zaijiang Shen<sup>1</sup> Shuyue Stella LF  
 Tucker Wilde<sup>1</sup> Valentina Pyatkin<sup>1</sup> Will Merrill<sup>1</sup> Yapei Chang<sup>2</sup> Yuling Gu<sup>1</sup> Zhiyuan Zeng<sup>1,2</sup>

Ashish Sabharwal<sup>1</sup> Luke Zettlemoyer<sup>2</sup> Pang Wei Koh<sup>1,2</sup>  
 Ali Farhadi<sup>1,2</sup> Noah A. Smith<sup>\*1,2</sup> Hannaneh Hajishirzi<sup>\*1,2</sup>

<sup>1</sup>Allen Institute for AI <sup>2</sup>University of Washington <sup>3</sup>Carnegie Mellon University <sup>4</sup>Stanford University <sup>5</sup>Mila  
<sup>6</sup>Université de Montréal <sup>7</sup>Princeton University <sup>8</sup>Massachusetts Institute of Technology <sup>9</sup>University of Maryland

\*OLMO 3 was a team effort; authors sorted alphabetically. \*marks core contributors. See author contributions here.

- 🟡 **Olmo 3 Base:** [Olmo-3-1025-7B](#) [Olmo-3-1125-32B](#)
- 🟡 **Olmo 3 Think:** [Olmo-3-7B-Think](#) [Olmo-3\(12-1\)-32B-Think](#)
- 🟡 **Olmo 3 Instruct:** [Olmo-3-7B-Instruct](#) [Olmo-3-1-32B-Instruct](#)
- 🟡 **Olmo 3 RLZero:** [Olmo-3-7B-RL-Zero-\(Math|Code|IF|General|Mix\)](#) [Olmo-3-1-7B-RL-Zero-\(Math|Code\)](#)
- 🟡 **Base Data:** [Pretrain: Dolma 3 Mix](#) [Midtrain: Dolma 3 Dolmino Mix](#) [Long-ctx: Dolma 3 Longino Mix](#)
- 🟡 **Think Data:** [Dolci-Think-\(SFT|DP0|RL\)-7B](#) [Dolci-Think-\(SFT|DP0|RL\)-32B](#)

### 3.4 Stage 1: Pretraining

We first train **OLMO 3 BASE** on **DOLMA 3 MIX**, our 6T token pretraining data mix. While **DOLMA 3 MIX** is comprised of largely the same types of data sources used in other open pretraining recipes ([Soldaini et al., 2024](#); [Bakoch et al., 2025](#); [OLMo et al., 2024](#)), we demonstrate three key novelties:

📍 [Contact: olmo@allenai.org](mailto:olmo@allenai.org)

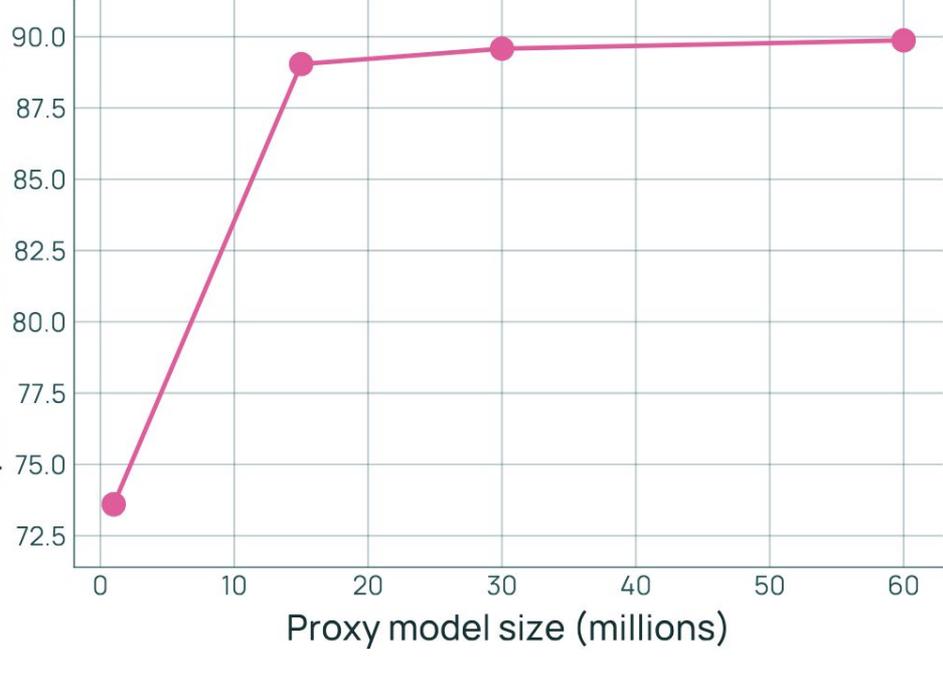
### Abstract

🔗 **Ai2**

We introduce **OLMO 3**, a family of state-of-the-art, fully-open language models at the 7B and 32B parameter scales. **OLMO 3** model construction targets long-context reasoning, function calling, coding, instruction following, general chat, and knowledge recall. This release includes the entire **model flow**, i.e., the full lifecycle of the family of models, including every stage, checkpoint, data point, and dependency used to build it. Our flagship model, **OLMO 3.1 THINK 32B**, is the strongest fully-open thinking model released to-date.

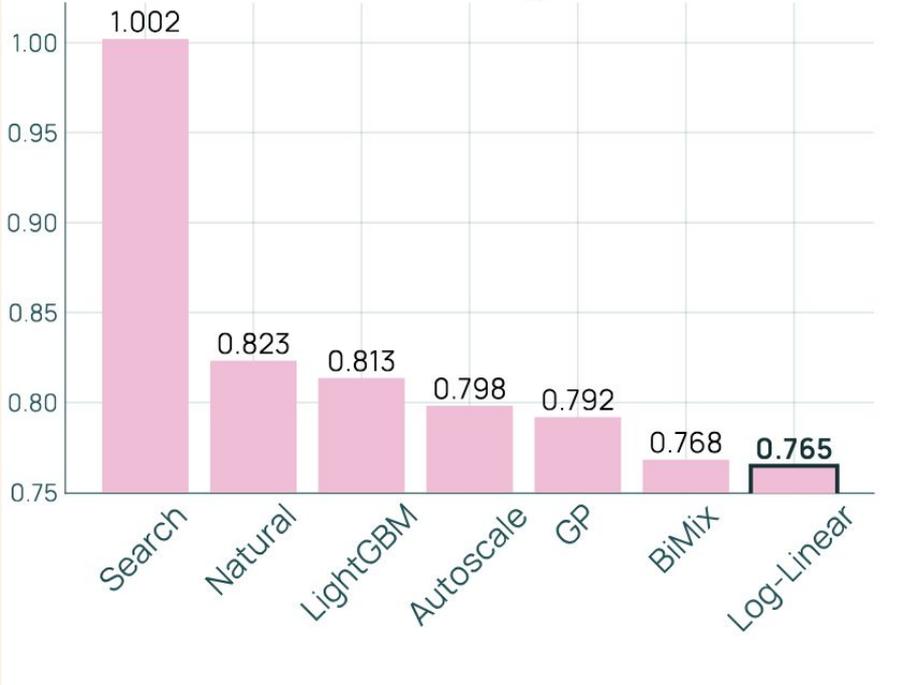
# Data mixing: Infrastructure

Correlation between proxy and large model performances

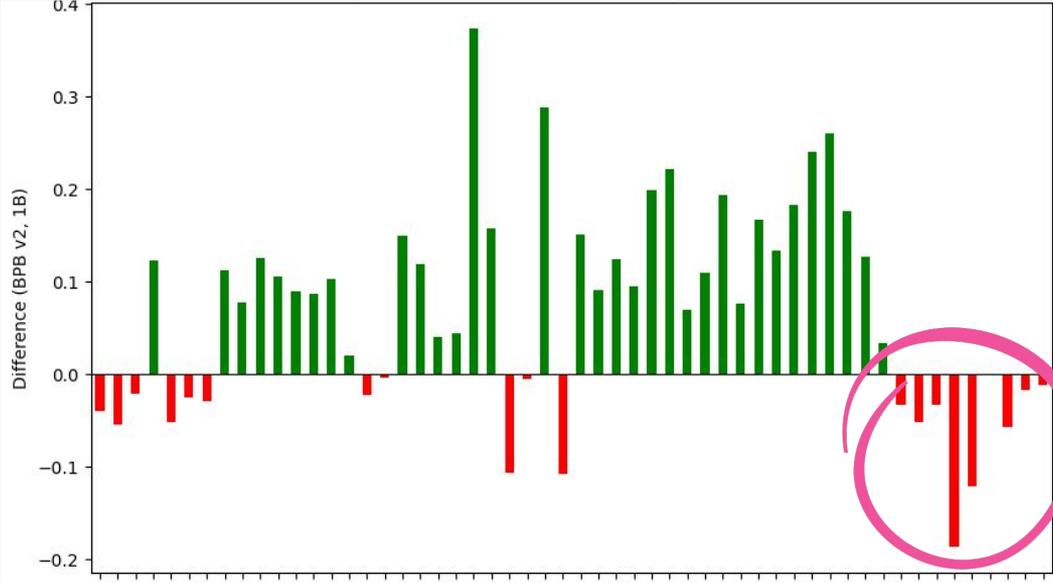


# Data mixing: Infrastructure

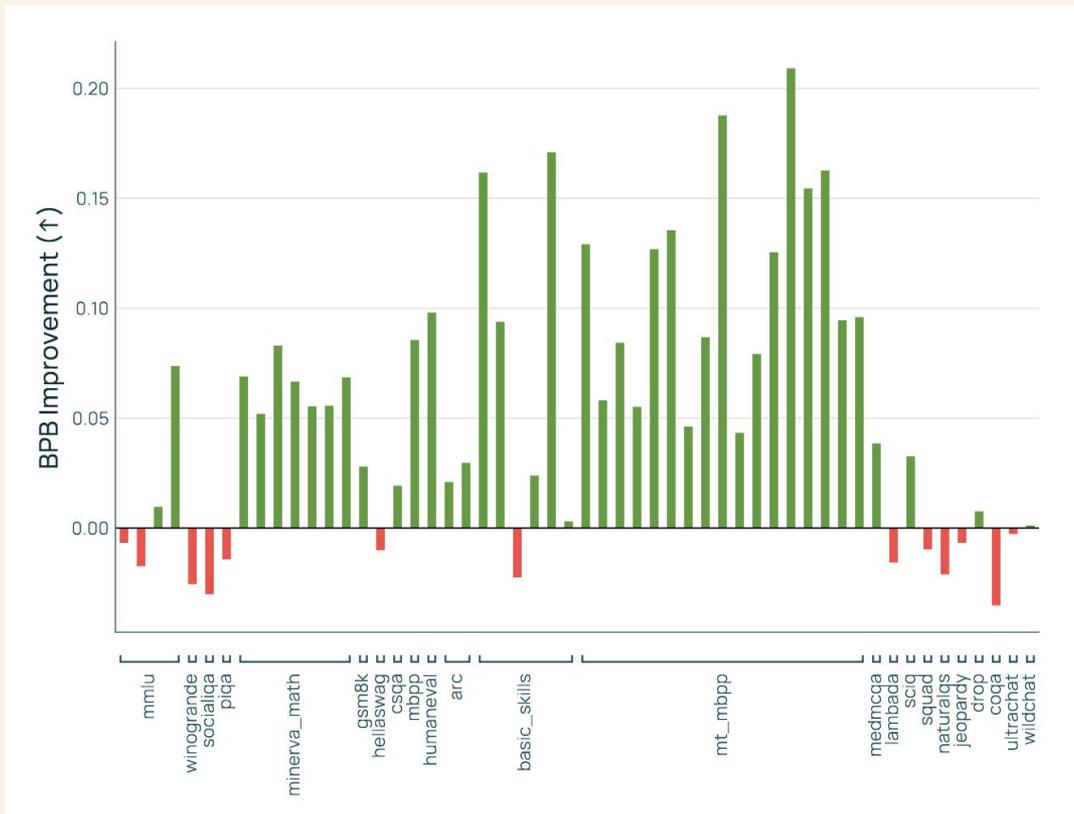
Performance  
(BPB, lower is better)



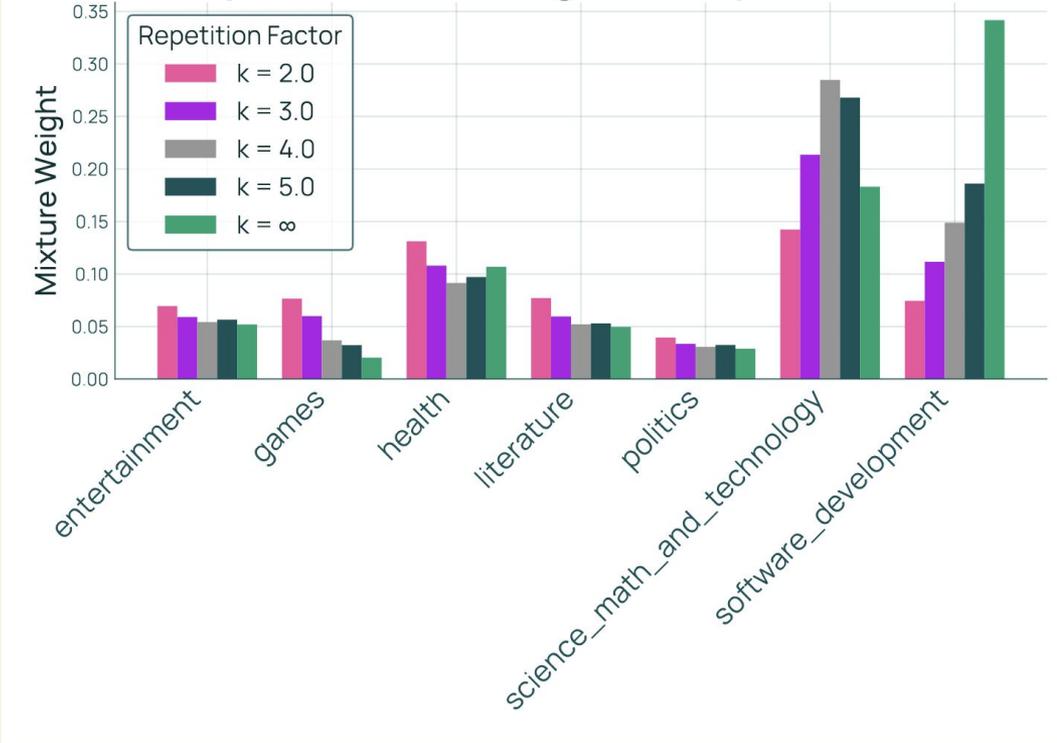
# Data mixing: Operations



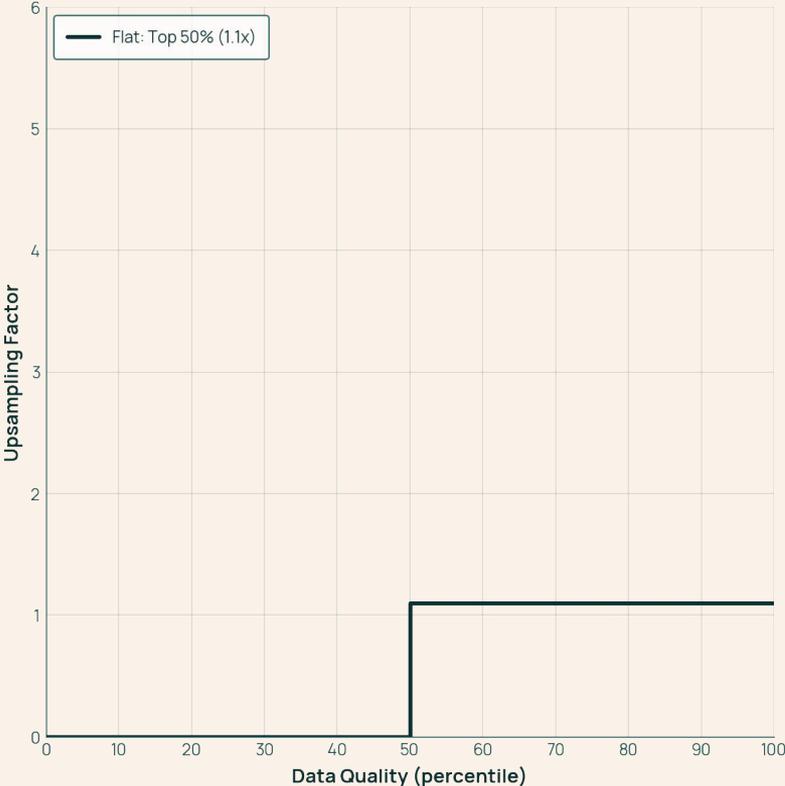
# Data mixing: Operations



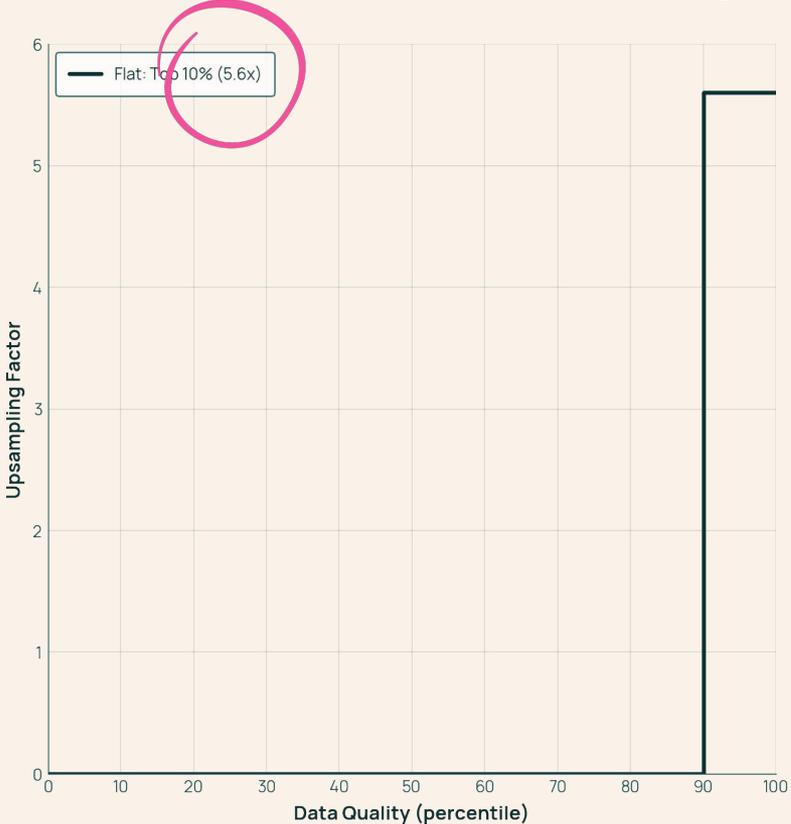
# Data mixing under data constraints



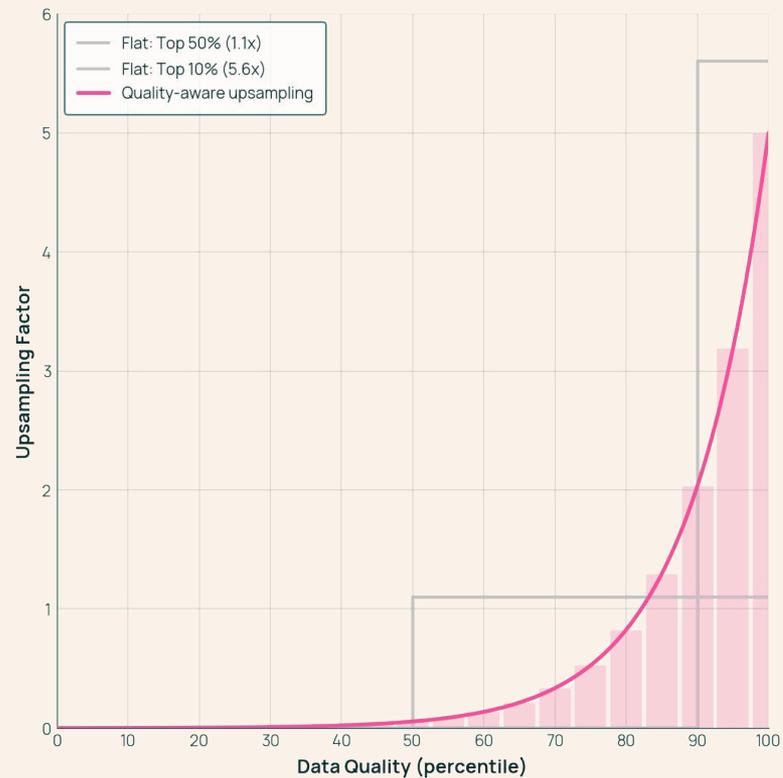
# Data constraints influence upsampling



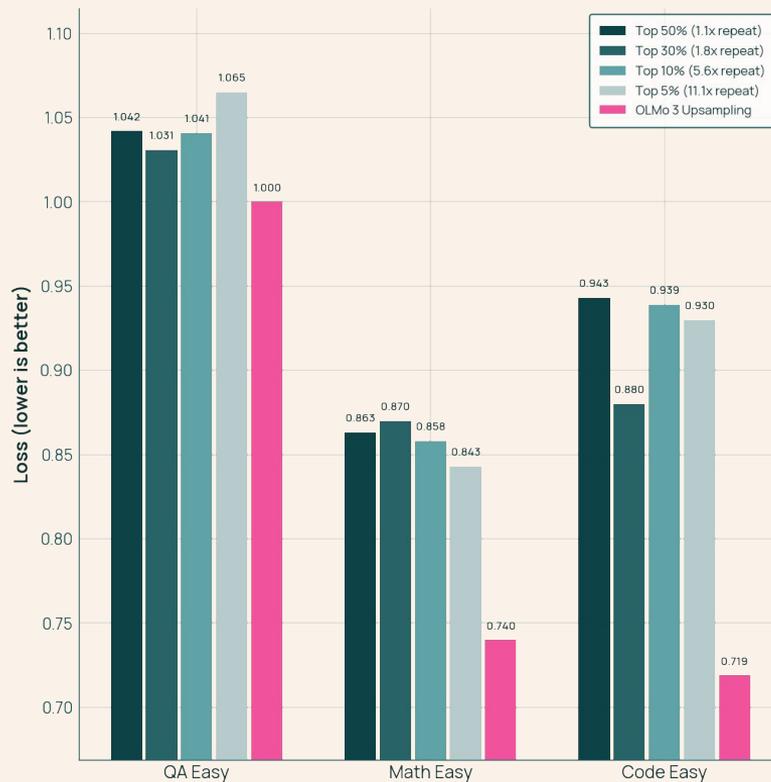
# Data constraints influence upsampling



# Quality-aware upsampling



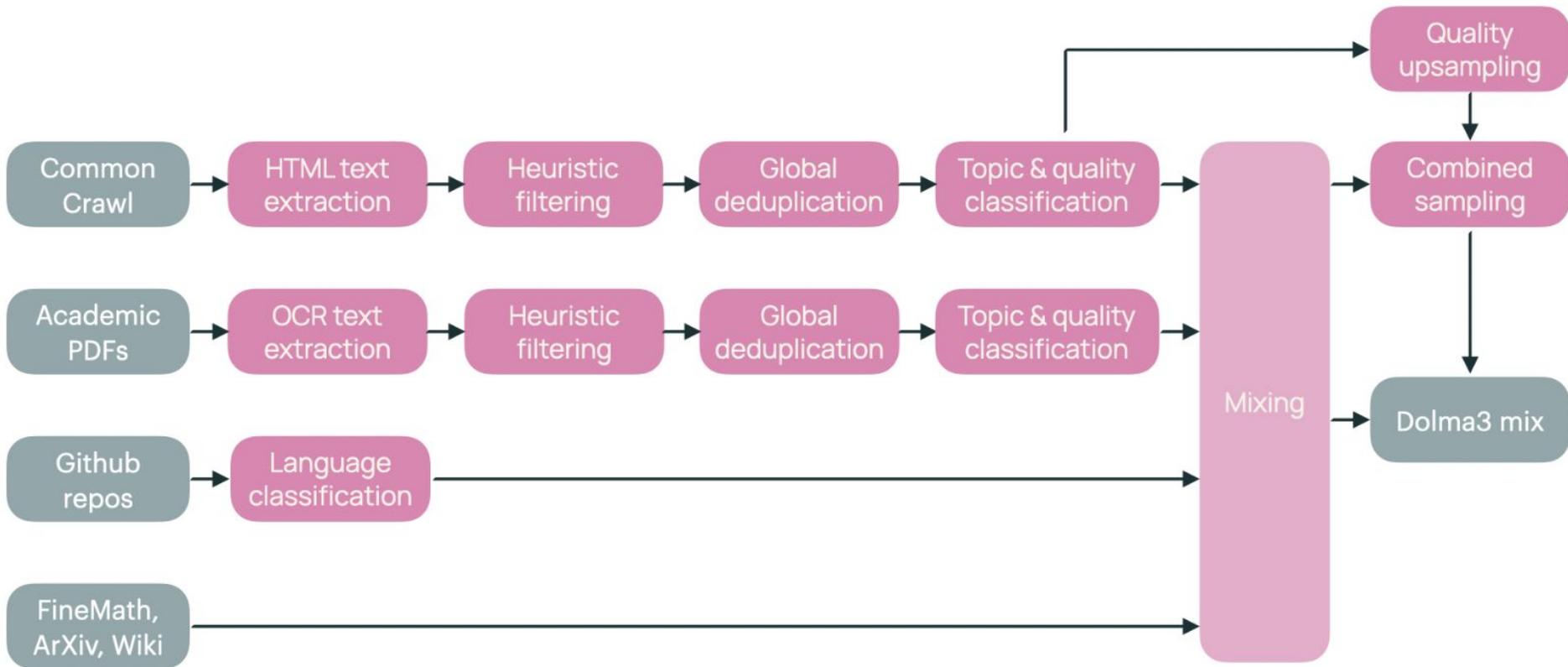
# Quality aware upsampling



# Pretraining: Dolma 3

Source	Type	9T Pool		6T Mix	
		Tokens	Documents	Tokens	Ratio
Common Crawl	Web pages	8.14T	9.67B	4.51T	76.1%
olmOCR Science PDFs	Academic documents	972B	101M	804.9B	13.6%
StackEdu (Rebalanced)	GitHub code	137B	167M	408.9B	6.9%
arXiv	Papers with LaTeX	21.4B	3.95M	50.8B	0.9%
FineMath 3+	Math web pages	34.1B	21.4M	151.9B	2.6%
Wikipedia & Wikibooks	Encyclopedic	3.69B	6.67M	2.5B	0.04%
<b>Total</b>		<b>9.31T</b>	<b>9.97B</b>	<b>5.93T</b>	<b>100%</b>

Extraction, filtering, deduplication, quality classification



# Filtering, Deduplication, Fine-Grained Organization

Common Crawl



255B Docs



39B Docs



10 B Docs

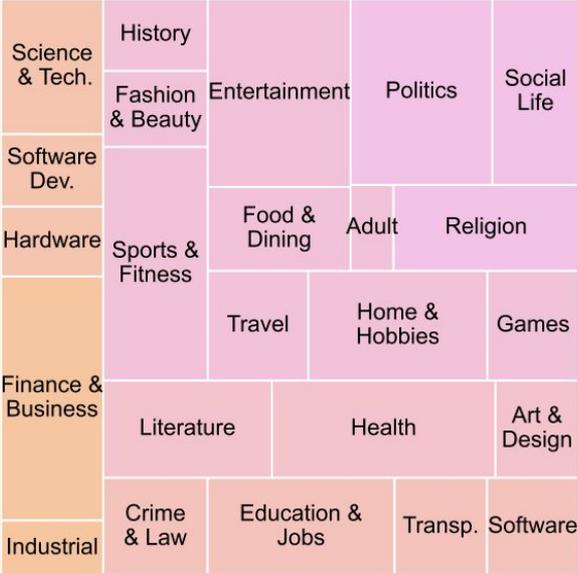
Web Organizer

Heuristic Filtering

- PII filtering
- Language filtering
- URL removal

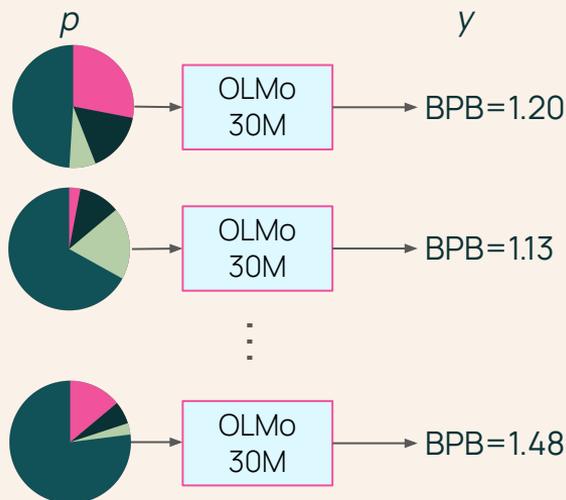
Deduplication

- Exact, global deduplication (new tooling!)
- Min-hash (fuzzy)
- Suffix array for boilerplate

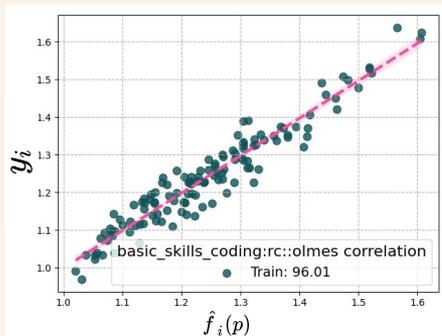


# From Pools to a Mix

1. **Swarm runs**: train  $K$  30M models with randomly sampled mixtures  $p$

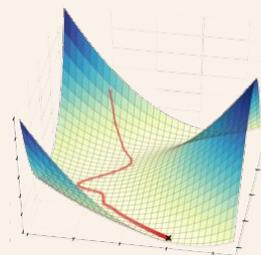
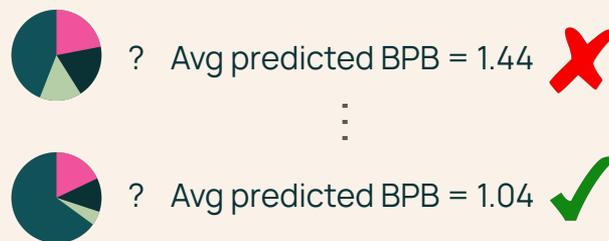


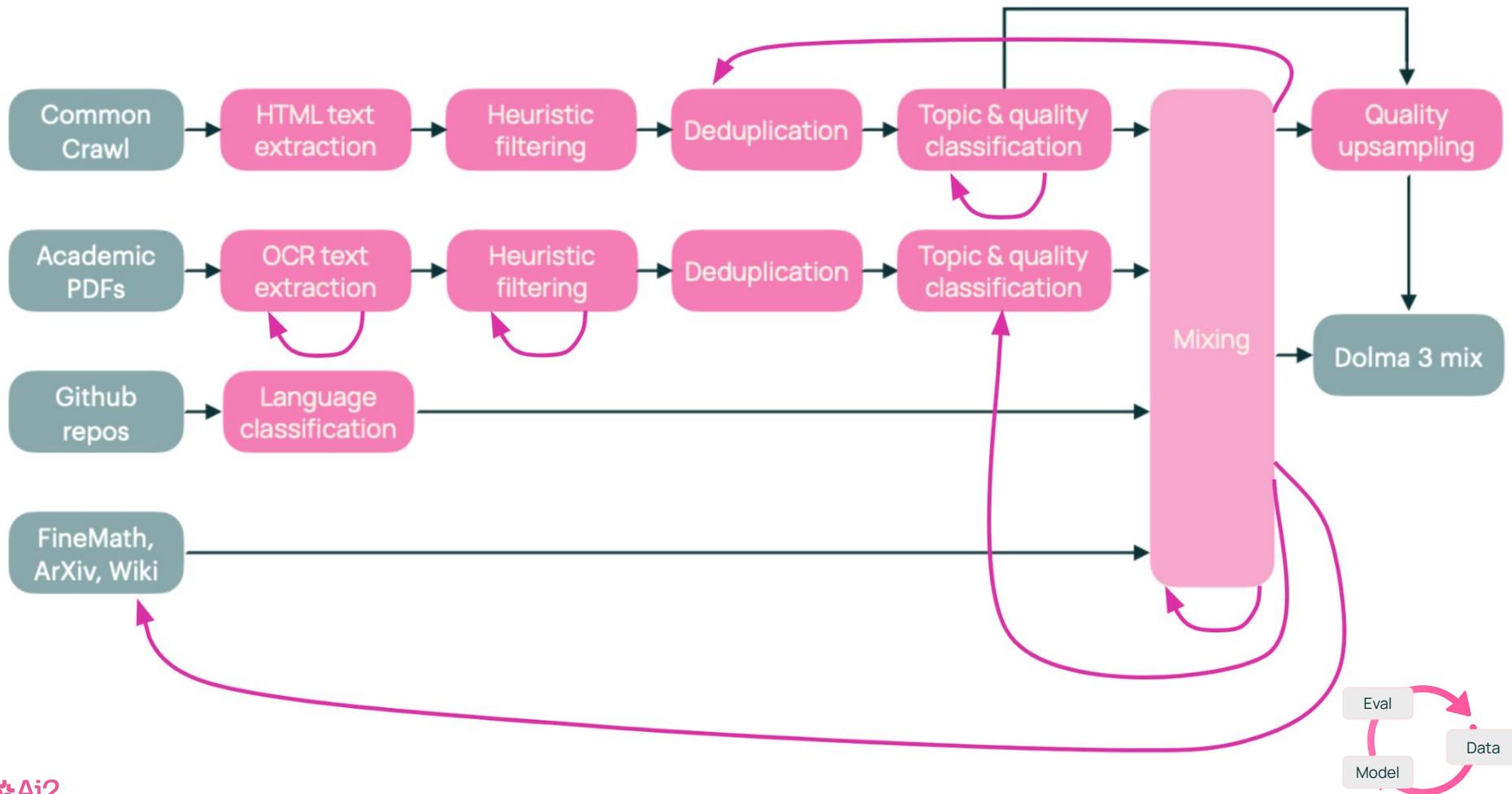
2. **Fit a regression model** for each benchmark task  $\hat{f}_i(p) \approx y_i$



3. **Solve optimization problem** to get optimal mix  $p^*$

$$\text{minimize}_{p \in \Delta^{m-1}} \frac{1}{n} \sum_{i=1}^n \hat{f}_i(p)$$



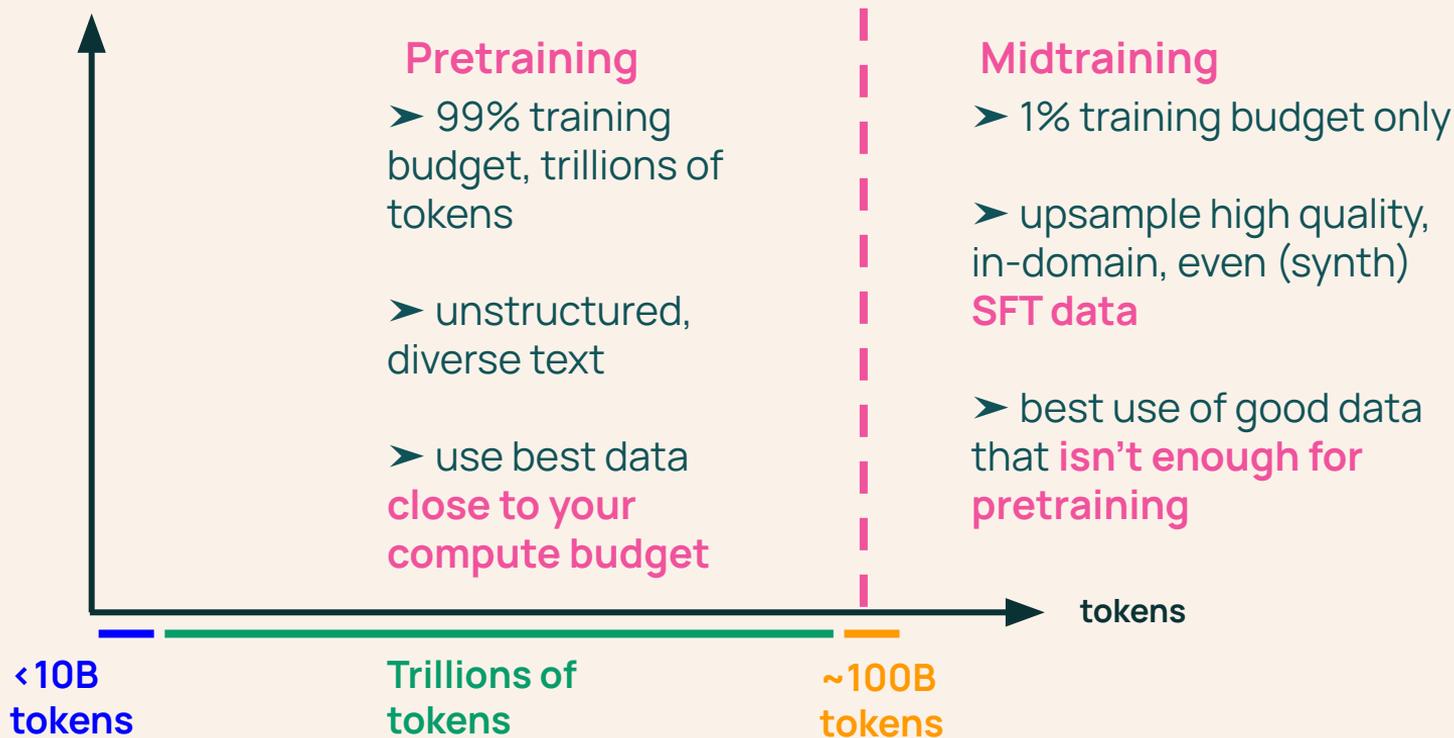


# Pretraining: Dolma 3

Source	Type	9T Pool		6T Mix	
		Tokens	Documents	Tokens	Ratio
Common Crawl	Web pages	8.14T	9.67B	4.51T	76.1%
olmOCR Science PDFs	Academic documents	972B	101M	804.9B	13.6%
StackEdu (Rebalanced)	GitHub code	137B	167M	408.9B	6.9%
arXiv	Papers with LaTeX	21.4B	3.95M	50.8B	0.9%
FineMath 3+	Math web pages	34.1B	21.4M	151.9B	2.6%
Wikipedia & Wikibooks	Encyclopedic	3.69B	6.67M	2.5B	0.04%
<b>Total</b>		<b>9.31T</b>	<b>9.97B</b>	<b>5.93T</b>	<b>100%</b>

Upsampled; higher quality increases probability of sampling a document

# Tension between Scale and Quality



# Dolma 3 Dolmino: Midtraining Mix

Math  
problem-solving  
through **code**  
and/or **discussion**

Instruction  
datasets

STEM PDFs and  
high-quality web  
pages

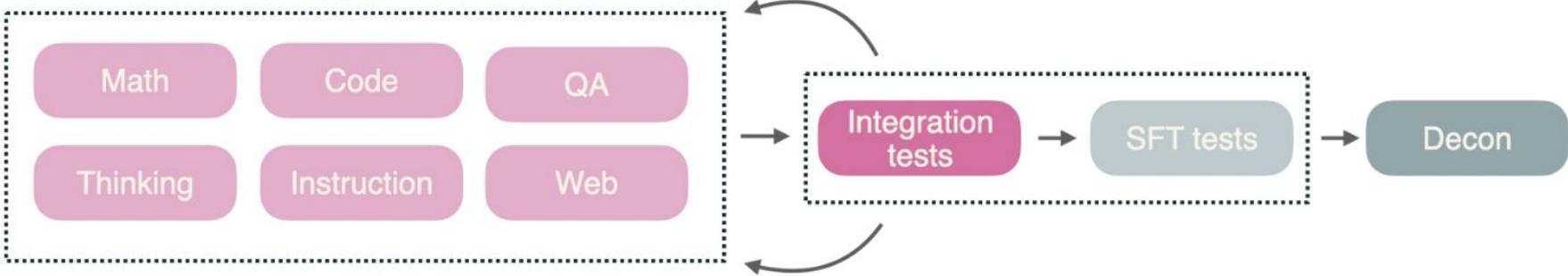


Type	Source	2T Pool		100B Mix	
		Tokens	Docs	Tokens	Docs
Math (synth)	TinyMATH Mind**	899M	1.42M	898M (0.9%)	1.52M
Math (synth)	TinyMATH PoT**	241M	729K	241M (0.24%)	758K
Math (synth)	CraneMath*	5.62B	6.55M	5.62B (5.63%)	7.24M
Math (synth)	MegaMatt*	3.88B	6.79M	1.73B (1.73%)	3.23M
Math (synth)	Dolmino Math^^	10.7B	21M	10.7B (10.7%)	22.3M
Code	StackEdu (FIM)^	21.4B	32M	10.0B (10.0%)	16.2M
Python (synth)	CraneCode*	18.8B	19.7M	10.0B (10.0%)	11.7M
QA (synth)	Reddit To Flashcards**	21.6B	370M	5.90B (5.9%)	101M
QA (synth)	Wiki To RCQA**	4.22B	22.3M	3.0B (3.0%)	16.3M
QA (synth)	Nemotron Synth QA^	487B	972M	5.0B (5.0%)	10.6M
Thinking (synth)	Math Meta-Reasoning**	1.05B	984K	381M (0.38%)	401K
Thinking (synth)	Code Meta-Reasoning**	1.27B	910K	459M (0.46%)	398K
Thinking (synth)	Program-Verifiable**	438M	384K	159M (0.16%)	158K
Thinking (synth)	OMR Rewrite FullThoughts^	850M	291K	850M (0.85%)	394K
Thinking (synth)	QWQ Reasoning Traces^	4.77B	438K	1.87B (1.87%)	401K
Thinking (synth)	General Reasoning Mix^	2.48B	668K	1.87B (1.87%)	732K
Thinking (synth)	Gemini Reasoning Traces^	246M	55.2K	246M (0.25%)	85.1K
Thinking (synth)	Llama Nemotron Reasoning Traces^	20.9B	3.91M	1.25B (1.25%)	368K
Thinking (synth)	OpenThoughts2 Reasoning Traces^	5.6B	1.11M	1.25B (1.25%)	402K
Instruction (synth)	Tulu 3 SFT^^	1.61B	1.95M	1.1B (1.1%)	1.45M
Instruction (synth)	Dolmino 1 Flan^^	16.8B	56.9M	5.0B (5.0%)	14.8M
PDFs	OLMOCR Science PDFs (High Q.)^	240B	28.7M	4.99B (5.0%)	1.20M
Web pages	STEM-Heavy Crawl^	5.21B	5.16M	4.99B (5.0%)	5.53M
Web pages	Common Crawl (High Q.)^	1.32T	965M	22.4B (22.5%)	18.3M
<b>Total</b>		<b>2.19T</b>	<b>2.52B</b>	<b>99.95B (100%)</b>	<b>236M</b>

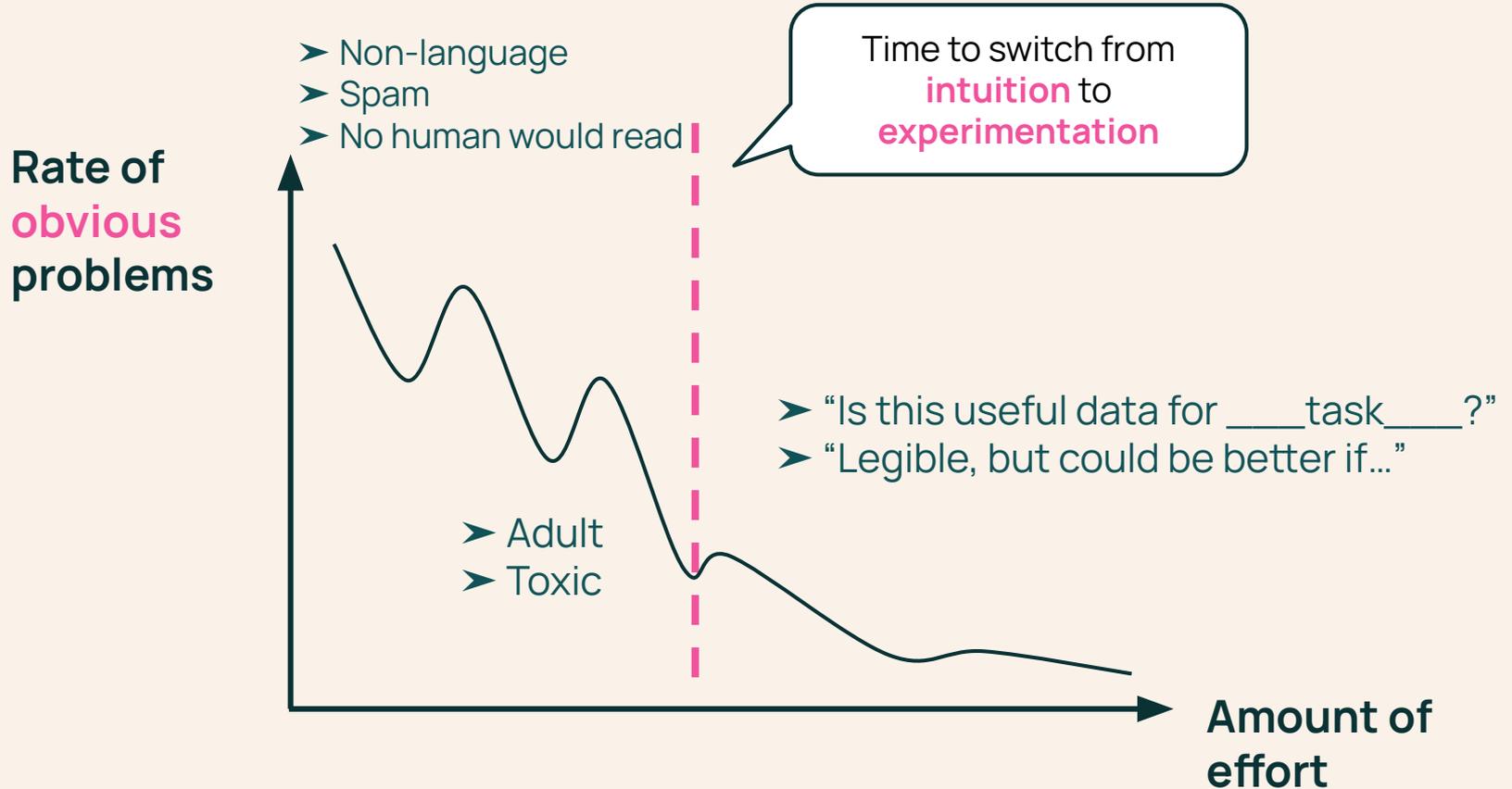
Demonstration of  
**diverse question  
structures**, rewritten  
from natural  
knowledge-rich data

**Math and code**  
problem-solving using  
**human-inspired  
meta-reasoning  
strategies/traces**

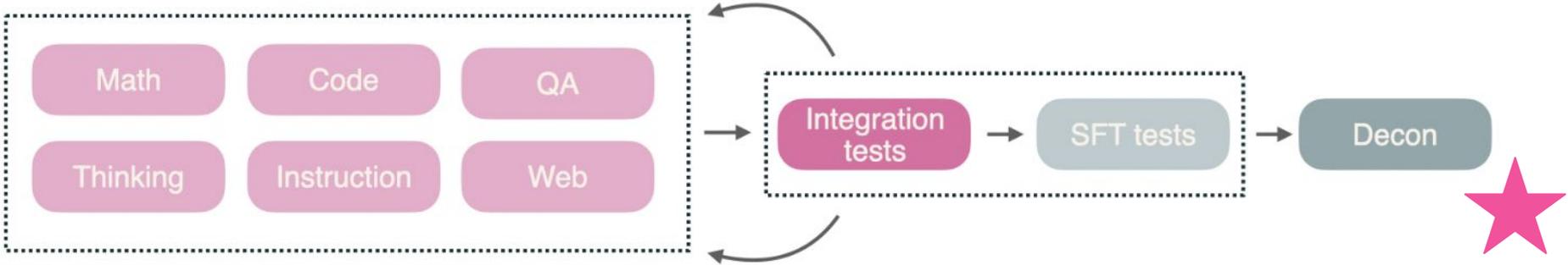
# Distributed Exploration & Centralized Evaluation



# Inspect Data Often, Favor Fast, Scalable Improvements



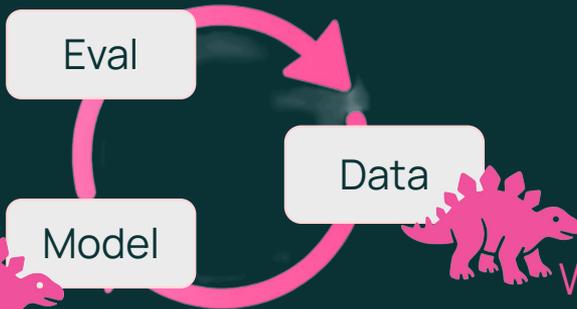
# Distributed Exploration & Centralized Evaluation



# Decontamination in Dolmino

Midtraining Data Sources		Evaluated splits:														All														Occurrences Of Contamination				
		Val/Test							All							All							All											
Dolmino 1 Flan	2e4	6e3	0	0	0	809	124	0	127	10	0	0	35	0	0	6e3	68	3e3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Tulu 3 SFT	1e4	270	0	14	0	85	42	2	25	5	0	0	3	0	0	287	6e3	893	554	1e3	1e3	689	260	14										
Nemotron Synth QA	6e3	692	10	2e3	79	2	23	167	0	80	74	17	15	31	22	519	689	27	1e3	41	64	189	2	0										
Dolmino Math	4e3	0	2e3	49	1e3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0										
Common Crawl (High Q.)	2e3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2e3	50	0	256	0	1	119	0	0										
StackEdu (FIM)	876	0	0	0	0	0	0	0	0	0	0	0	0	0	0	792	0	0	24	0	1	58	0	1										
Gemini Reasoning Traces	606	0	513	31	0	0	0	0	0	0	0	43	0	19	0	0	0	0	0	0	0	0	0	0										
OLMOCR Science PDFs (High Q.)	554	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97	4	1	390	0	19	33	10	0										
Sponge	308	0	0	0	0	0	0	0	0	0	0	0	0	0	0	38	1	0	190	0	0	79	0	0										
General Reasoning Mix	113	0	6	68	3	0	0	0	0	0	0	5	0	0	0	0	0	0	4	0	27	0	0	0										
Total contam		Evaluated splits:														All																		
Perf Δ		1.7	2.0	-1.2	-1.6	13.9	0.4	-0.4	-2.4	0.6	-0.1	-0.7	-0.0	0.6	0.9	-1.4	0.0	1.4	-0.3	1.8	1.1	-0.4	1.1	-0.3										
% contam		27%	50%	4%	100%	9%	2%	2%	2%	2%	0%	5%	1%	24%	3%	6%	3%	2%	13%	3%	2%	6%	0%	0%										
		SQUAD	Minerva	MMLU (MC)	GSM8K	DROP	CoQA (MC)	HumEval (@16)	DROP (MC)	LAMBADA	MedMCOA (MC)	MedQA En (MC)	SQuAD (MC)	LeetCode (@16)	M-E-HumEval (@16)	Jeopardy	HellaSwag	CoQA	ARC (MC)	PIQA (MC)	CSQA (MC)	SciQ (MC)	Winogrande	SocialQA (MC)										

We are here!



We are here!

## Olmo 3

Olmo Team\*

Allyson Ettlinger<sup>\*1</sup> Amanda Bertsch<sup>\*1,3</sup> Bailey Kuehl<sup>\*1</sup> David Graham<sup>\*1</sup>  
David Heineman<sup>\*1</sup> Dirk Groeneveld<sup>\*1</sup> Faeze Brahman<sup>\*1</sup> Finbarr Timbers<sup>\*1</sup>  
Hamish Ivison<sup>\*1,2</sup> Jacob Morrison<sup>\*1,2</sup> Jake Poznanski<sup>\*1</sup> Kyle Lo<sup>\*1,2</sup> Luca Soldaini<sup>\*1</sup>  
Matt Jordan<sup>\*1</sup> Mayee Chen<sup>\*1,4</sup> Michael Noukhotovitch<sup>\*1,5,6</sup> Nathan Lambert<sup>\*1</sup>  
Pete Walsh<sup>\*1</sup> Pradeep Dasigi<sup>\*1</sup> Robert Berry<sup>\*1</sup> Saumya Malik<sup>\*1</sup> Saurabh Shah<sup>\*1</sup>  
Scott Geng<sup>\*1,2</sup> Shane Arora<sup>\*1</sup> Shashank Gupta<sup>\*1</sup> Taira Anderson<sup>\*1</sup> Teng Xiao<sup>\*1</sup>  
Tyler Murray<sup>\*1</sup> Tyler Romero<sup>\*1</sup> Victoria Graf<sup>\*1,2</sup>

Akari Asai<sup>1,3</sup> Akshita Bhagia<sup>1</sup> Alexander Wettig<sup>1</sup> Alisa Liu<sup>2</sup> Aman Rangapur<sup>1</sup>  
Chloe Anastasiades<sup>1</sup> Costa Huang<sup>1</sup> Dustin Schwenk<sup>1</sup> Harsh Trivedi<sup>1</sup> Ian Magnusson<sup>1,2</sup>  
Jaron Lochner<sup>1</sup> Jiacheng Liu<sup>1</sup> Lester James V. Miranda<sup>1</sup> Maarten Sap<sup>1,3</sup> Malia Morgan<sup>1</sup>  
Michael Schmitz<sup>1</sup> Michal Querquin<sup>1</sup> Michael Wilson<sup>1</sup> Regan Huff<sup>1</sup> Ronan Le Bras<sup>1</sup>  
Rui Xin<sup>1</sup> Rulin Shao<sup>1</sup> Sam Stojanberg<sup>1</sup> Shannon Zhai<sup>1</sup> Shuyue Stella Li<sup>1</sup>  
Tucker Wilde<sup>1</sup> Valentina Pyatkin<sup>1</sup> Will Merrill<sup>1</sup> Yapei Chang<sup>1</sup> Yuling Gu<sup>1</sup> Zhiyuan Zeng<sup>1,2</sup>

Ashish Sabharwal<sup>1</sup> Luke Zettlemoyer<sup>1</sup> Pang Wei Koh<sup>1,2</sup>  
Ali Farhadi<sup>1,2</sup> Noah A. Smith<sup>\*1,2</sup> Hannaneh Hajishirzi<sup>\*1,2</sup>

<sup>1</sup>Allen Institute for AI <sup>2</sup>University of Washington <sup>3</sup>Carnegie Mellon University <sup>4</sup>Stanford University <sup>5</sup>Mila  
<sup>6</sup>Université de Montréal <sup>7</sup>Princeton University <sup>8</sup>Massachusetts Institute of Technology <sup>9</sup>University of Maryland

\*OLMO 3 was a team effort; authors sorted alphabetically. \*marks core contributors. See author contributions here.

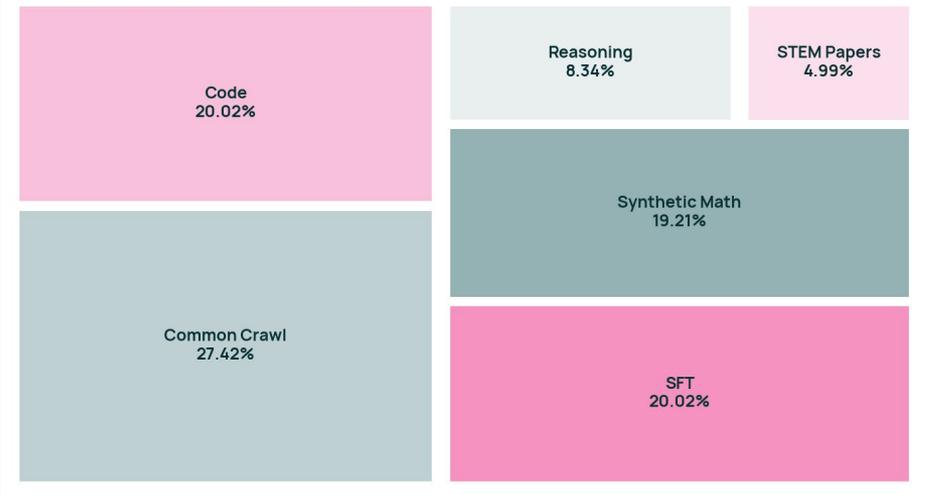
- 🟡 **Olmo 3 Base:** `Olmo-3-1025-7B` `Olmo-3-1125-32B`
- 🟡 **Olmo 3 Think:** `Olmo-3-7B-Think` `Olmo-3(12-1)-32B-Think`
- 🟡 **Olmo 3 Instruct:** `Olmo-3-7B-Instruct` `Olmo-3-1-32B-Instruct`
- 🟡 **Olmo 3 RL Zero:** `Olmo-3-7B-RL-Zero-(MathCode|IF|General|Mix)` `Olmo-3-1-7B-RL-Zero-(MathCode)`
- 🟢 **Base Data:** `Pretrain: Dolma 3 Mix` `Midtrain: Dolma 3 Dolmino Mix` `Long-ctx: Dolma 3 Longino Mix`
- 🟢 **Think Data:** `Dolci-Think-(SFT|DPO|RL)-7B` `Dolci-Think-(SFT|DPO|RL)-32B`
- 🟢 **Instruct Data:** `Dolci-Instruct-(SFT|DPO|RL)`
- 🟢 **RL-Zero Data:** `Dolci-RL-Zero-(MathCode|IF|General)-7B` `Dolci-RL-Zero-Mix-7B`
- 🔗 **Training Code:** `olmo-core` (pretrain) `Open Instruct` (posttrain)
- 🔗 **Data Code:** `dasnap-rs` (data processing) `dupliocus` (deduplication) `dolma3` (data recipes)
- 🔗 **Eval Code:** `OLMOS` (eval suite) `laccos` (eval decontamination)
- 📄 **Training Logs:** `Olmo-3-7B-(Base|Think|Instruct|RL-Zero)` `Olmo-3-32B-(Base|Think|Instruct)`
- 📄 **Demo:** `32B Think` `32B Instruct` `7B Think` `7B Instruct`
- 📄 **Contact:** `olmo@allenai.org`

### Abstract

🔗 Ai2

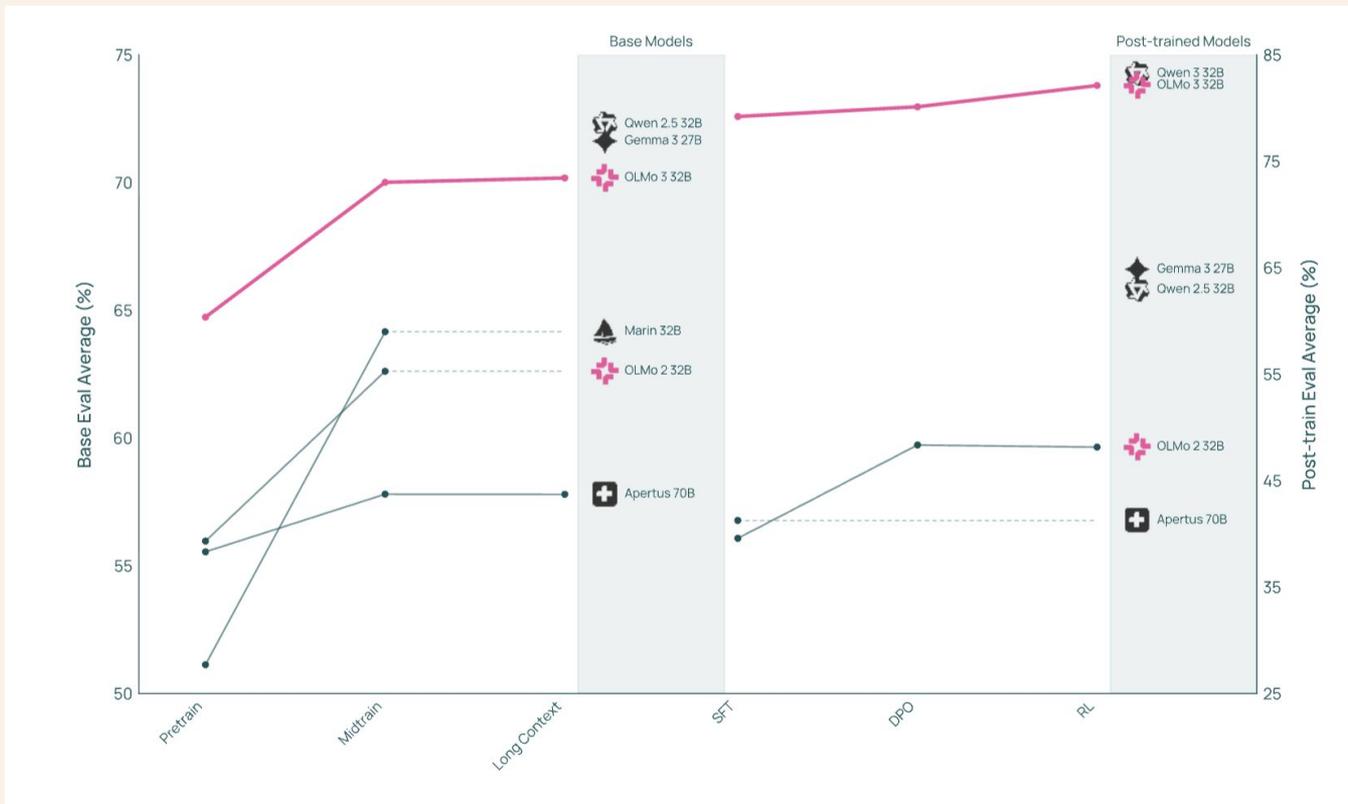
We introduce **OLMO 3**, a family of state-of-the-art, fully-open language models at the 7B and 32B parameter scales. OLMO 3 model construction targets long-context reasoning, function calling, coding, instruction following, general chat, and knowledge recall. This release includes the entire model flow, i.e., the full lifecycle of the family of models, including every stage, checkpoint, data point, and dependency used to build it. Our flagship model, **OLMO 3.1 THINK 32B**, is the strongest fully-open thinking model released to-date.

# Midtraining Olmo 3



Boost data that closely match post-train capability targets

# Olmo 3 Model Flow



# Thank you!



Twitter  
Bluesky  
Email

@heinemandavidj

@davidheineman.com

davidh@allenai.org



... and many more (ordered arbitrarily)



PAUL G. ALLEN SCHOOL  
OF COMPUTER SCIENCE & ENGINEERING



## Signal and Noise: A Framework for Reducing Uncertainty in Language Model Evaluation

David Heineman<sup>1</sup> Valentin Hofmann<sup>1\*</sup> Ian Magnusson<sup>1\*</sup> Yuling Gu<sup>1</sup>  
Noah A. Smith<sup>1\*</sup> Hannaneh Hajishirzi<sup>1\*</sup> Kyle Lo<sup>1\*</sup> Jesse Dodge<sup>1</sup>

<sup>1</sup>Allen Institute for Artificial Intelligence

<sup>2</sup>Paul G. Allen School of Computer Science

## Olmix: A Framework for Data Mixing Throughout LM Development

Mayee Chen<sup>1,2</sup> Tyler Murray<sup>1</sup> David Heineman<sup>1</sup> Matt Jordan<sup>1</sup> Hannaneh Hajishirzi<sup>1,3</sup>  
Christopher Ré<sup>2</sup> Luca Soldaini<sup>1</sup> Kyle Lo<sup>1,3</sup>

<sup>1</sup>Allen Institute for AI <sup>2</sup>Stanford University <sup>3</sup>University of Washington

Code: [Olmix](#) Data: [Olmix](#) Contact: [mfchen@cs.stanford.edu](mailto:mfchen@cs.stanford.edu) [{lucas,kyle}@allenai.org](mailto:{lucas,kyle}@allenai.org)

### Abstract

Data mixing—determining the retraining language models (LMs). V applied during real-world LM development. First, the configuration design choices across existing methods like data constraints. We conduct design choices lead to a strong mix LM development as datasets are addressed by existing works, with the mixture over the domain set is mixture reuse, a mechanism that re by the update. Over a sequence of mixture reuse matches the performance and improves over training

### 1 Introduction

Language model development is expensive decisions such as what architecture

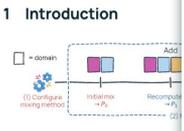


Figure 1 Two problems with data mixing: (1) How to mix data? (2) How to efficiently mix under

Modern language models (LMs) are trained

### Olmo 3

Olmo Team\*

Allyson Ettinger<sup>1</sup> Amanda Bertsch<sup>1,3</sup> Bailey Kuehl<sup>1,3</sup> David Graham<sup>1</sup>  
David Heineman<sup>1</sup> Dirk Groeneveld<sup>1</sup> Faæze Brahman<sup>1</sup> Finbar Timbers<sup>1</sup>  
Hannah Wong<sup>1,2</sup> Jacob Morrison<sup>1,3</sup> Jake Poznanski<sup>1</sup> Kyle Lo<sup>1,2</sup> Luca Soldaini<sup>1</sup>  
Matt Jordan<sup>1</sup> Mayee Chen<sup>1,4</sup> Michael Neukirch<sup>1,5,6</sup> Nathan Lambert<sup>1</sup>  
Pete Walsh<sup>1</sup> Pradeep Dasigi<sup>1</sup> Robert Berry<sup>1</sup> Saumya Malik<sup>1</sup> Saurabh Shah<sup>1</sup>  
Scott Geng<sup>1,2</sup> Shane Arora<sup>1</sup> Shashank Gupta<sup>1</sup> Taira Anderson<sup>1</sup> Teng Xiao<sup>1</sup>  
Tyler Murray<sup>1</sup> Tyler Romero<sup>1</sup> Victoria Graf<sup>1,2</sup>

Akari Asai<sup>1</sup> Akshita Bhagia<sup>1</sup> Alexander Wettig<sup>1</sup> Alisa Liu<sup>1</sup> Aman Rangapur  
Chloe Anastasiades<sup>1</sup> Costa Huang<sup>1</sup> Dustin Schwenk<sup>1</sup> Harsh Trivedi<sup>1</sup> Ian Magnusson<sup>1,2</sup>  
Jaron Lochner<sup>1</sup> Jiacheng Liu<sup>1</sup> Lester James V. Miranda<sup>1</sup> Maarten Sap<sup>1,2</sup> Malli Morgan<sup>1</sup>  
Michael Schmitz<sup>1</sup> Michael Guergin<sup>1</sup> Michael Wilson<sup>1</sup> Regan Huff<sup>1</sup> Roman Le Bras<sup>1</sup>  
Ruixi<sup>1</sup> Ruiluo Shao<sup>1</sup> Sam Sigonsberg<sup>1</sup> Shannon Zejiang Shen<sup>1</sup> Shuyue Stella Li<sup>1</sup>  
Tucker Wilde<sup>1</sup> Valentina Pytkin<sup>1</sup> Will Merrill<sup>1</sup> Yapei Chang<sup>1</sup> Yuling Gu<sup>1</sup> Zhiyuan Zeng<sup>1,2</sup>

Ashish Sabharwal<sup>1</sup> Luke Zettlemoyer<sup>2</sup> Pang Wei Koh<sup>1,2</sup>

All Farhadi<sup>1,2</sup> Noah A. Smith<sup>1,2</sup> Hannaneh Hajishirzi<sup>1,2</sup>

<sup>1</sup>Allen Institute for AI <sup>2</sup>University of Washington <sup>3</sup>Carleton Mellon University <sup>4</sup>Stanford University <sup>5</sup>Mila <sup>6</sup>Université de Montréal <sup>7</sup>Princeton University <sup>8</sup>Massachusetts Institute of Technology <sup>9</sup>University of Maryland

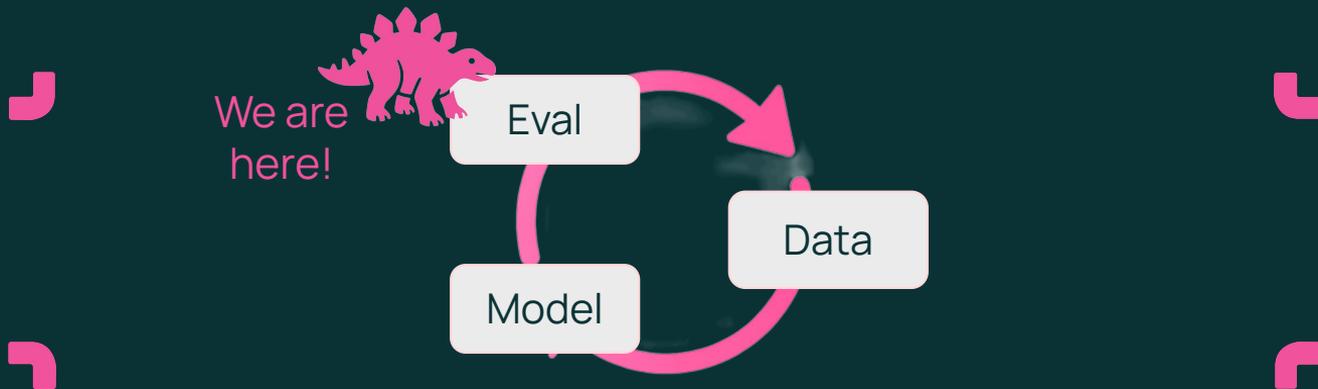
\*Olmo 3 was a team effort; authors sorted alphabetically. \*marks core contributors. See author contributions here.

- Olmo 3 Base: Olmo-3-102B-7B Olmo-3-112B-32B
  - Olmo 3 Think: Olmo-3-7B-Think Olmo-3(13.1)-32B-Think
  - Olmo 3 Instruct: Olmo-3-7B-Instruct Olmo-3.1-32B-Instruct
  - Olmo 3 RL-Zero: Olmo-3-7B-RL-Zero (Think/Code/IF/General/HiX) Olmo-3.1-7B-RL-Zero (Math/Code)
  - Base Data Pretrain: Dolma 3 Mix Midtrain: Dolma 3 Dolma Mix Long-Ctx: Dolma 3 Longmix HiX
  - Think Data: Dolci-Think (SFT/DP/RL)-7B Dolci-Think (SFT/DP/RL)-32B
  - Instruct Data: Dolci-Instruct (SFT/DP/RL)
  - RL-Zero Data: Dolci-RL-Zero (Math/Code/IF/General)-7B Dolci-RL-Zero-HiX-7B
- Training Code: [olmo-core](#) (pretrain) [Open Instruct](#) (posttrain)  
Data Code: [data-samp-re](#) (data processing) [duplication](#) (deduplication) [dolma3](#) (data recipes)  
Eval Code: [OLM3](#) (eval suite) [soco](#) (eval decomposition)
- Training Logs: [Olmo-3-7B](#) [\(Base/Think/Instruct/RL-Zero\)](#) [Olmo-3-32B](#) [\(Base/Think/Instruct\)](#)  
Demo: [32B Think](#) [32B Instruct](#) [7B Think](#) [7B Instruct](#)  
Contact: [olmo@allenai.org](mailto:olmo@allenai.org)

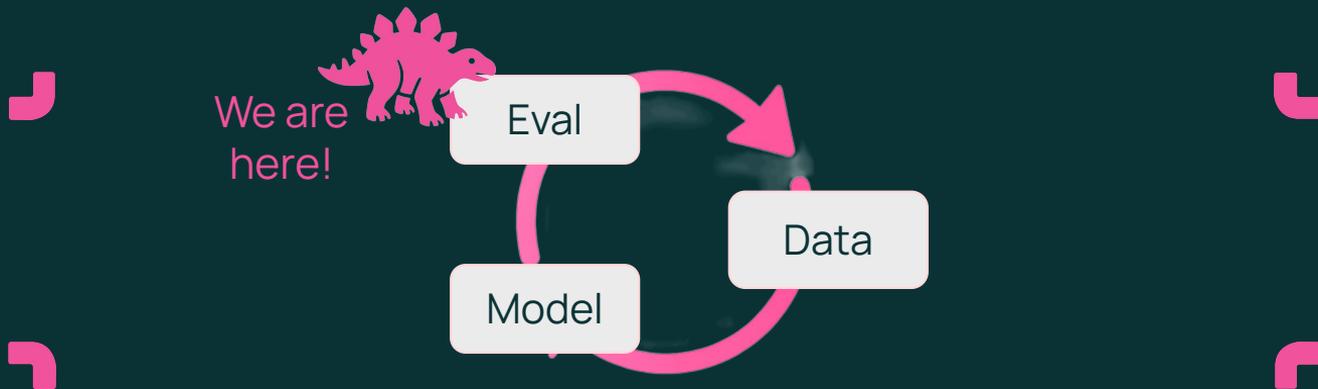
### Abstract

We introduce OLMO 3, a family of state-of-the-art, fully-open language models at the 7B and 32B parameter scales. OLMO 3 model construction targets long-context reasoning, function calling, coding, instruction following, general chat, and knowledge recall. This release includes the entire model flow, i.e., the full lifecycle of the family of models, including every stage: checkpoint, data point, and dependency used to build it. Our flagship model, OLMO 3.1 THINK 32B, is the strongest fully-open thinking model released to date.





- > Evaluation Methods
- > Evaluation in Olmo 3

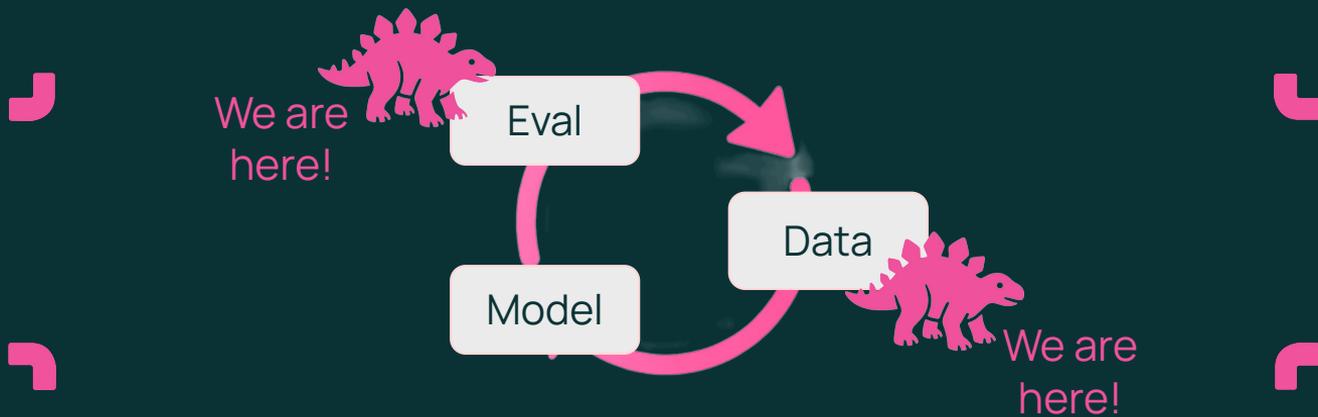


- > Evaluation Methods
- > Evaluation in Olmo 3

- › The Olmo 3 Models
- › Evaluation for LMs
- › Methods & Decision Making
- › Model Selection in Olmo 3
- › Data mixing with Olmix

- › The Olmo 3 Models
- › Evaluation for LMs
- › **Methods & Decision Making**
- › Model Selection in Olmo 3
- › Data mixing with Olmix

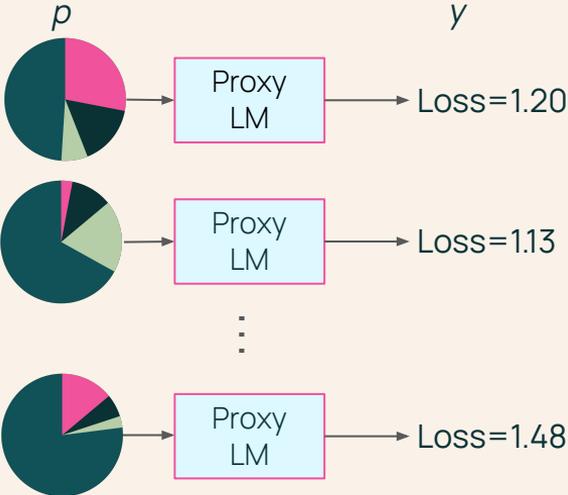
- › The Olmo 3 Models
- › Evaluation for LMs
- › Methods & Decision Making
- › Model Selection in Olmo 3
- › Data mixing with Olmix



- > Data Mixing with Olmix
- > Data for Olmo 3

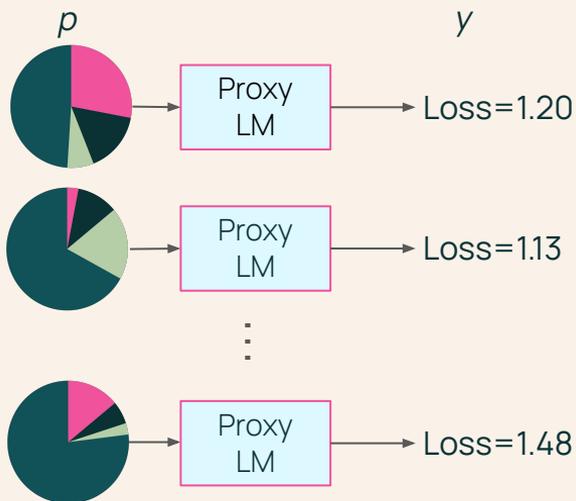
# Data mixing

1. **Swarm**: Train  $K$  small models with randomly sampled mixtures  $p$



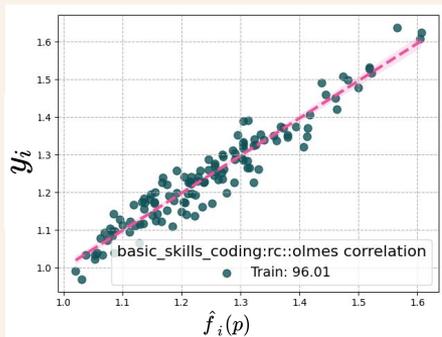
# Data mixing

1. **Swarm:** Train  $K$  small models with randomly sampled mixtures  $p$



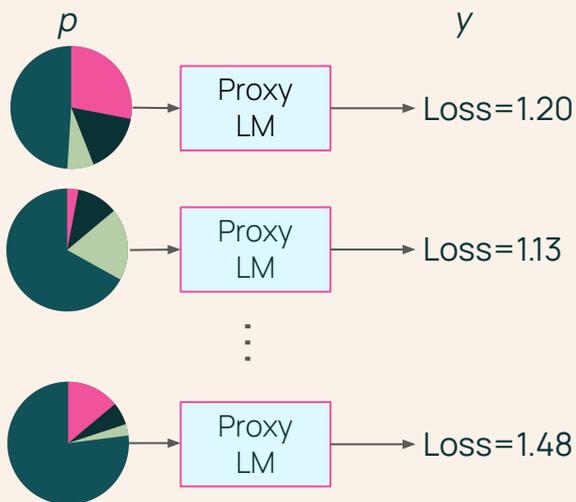
2. **Regression:** Fit function to predict LM performance given mixture  $p$

$$\hat{f}_i(p) \approx y_i$$

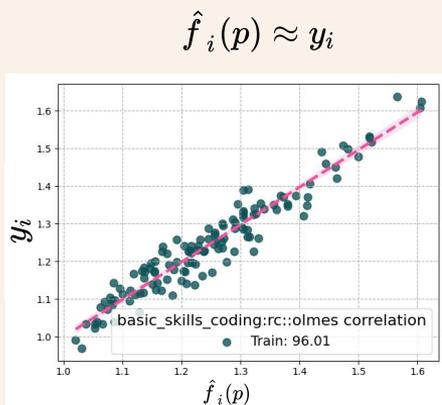


# Data mixing

1. **Swarm:** Train  $K$  small models with randomly sampled mixtures  $p$



2. **Regression:** Fit function to predict LM performance given mixture  $p$



3. **Optimize:** Use fit function to solve for optimal mix  $p^*$

$$\underset{p \in \Delta^{m-1}}{\text{minimize}} \frac{1}{n} \sum_{i=1}^n \hat{f}_i(p)$$



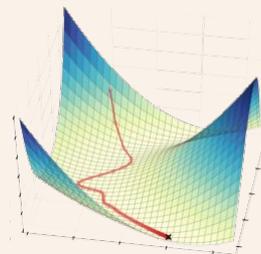
? Predicted loss = 1.44



⋮

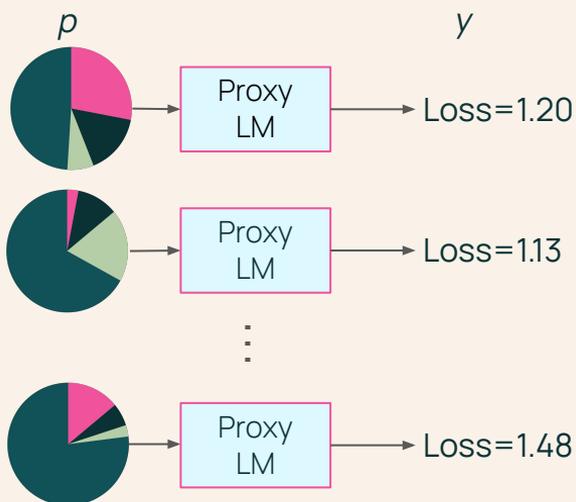


? Predicted loss = 1.04

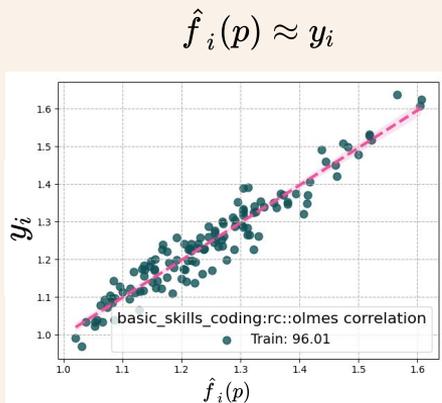


# Data mixing

1. **Swarm:** Train  $K$  small models with randomly sampled mixtures  $p$

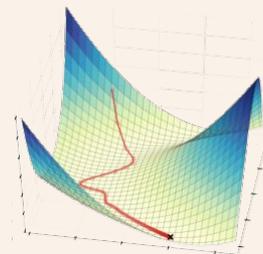
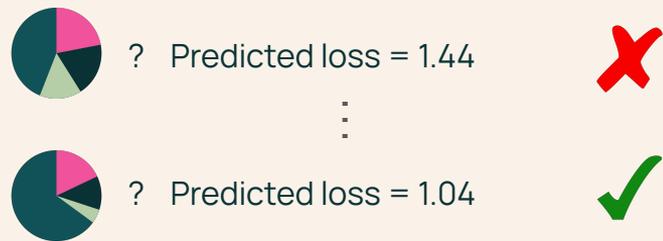


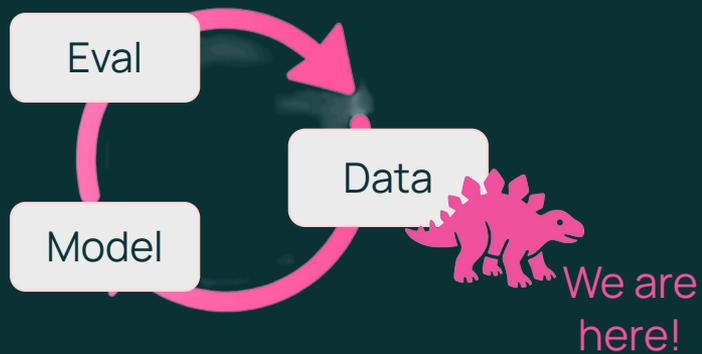
2. **Regression:** Fit function to predict LM performance given mixture  $p$



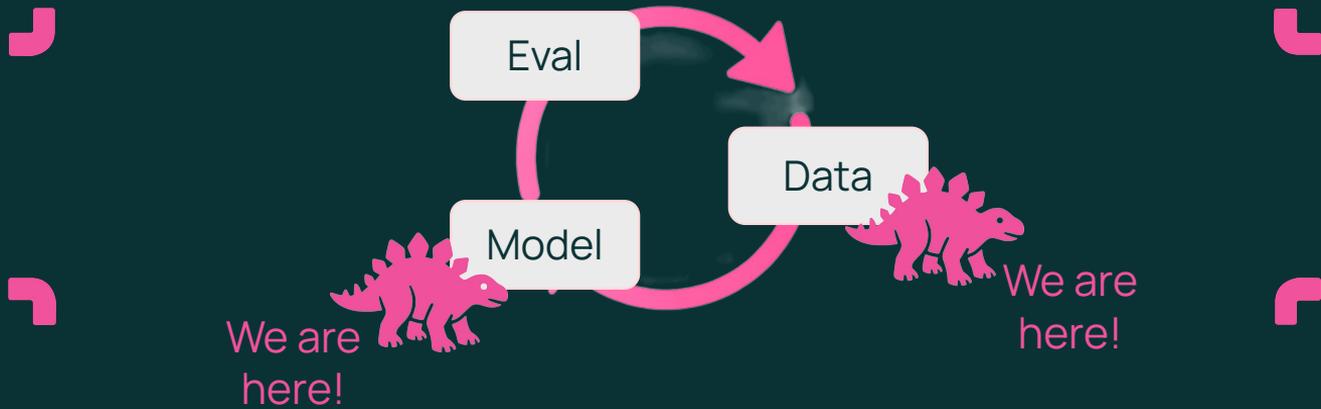
3. **Optimize:** Use fit function to solve for optimal mix  $p^*$

$$\underset{p \in \Delta^{m-1}}{\text{minimize}} \frac{1}{n} \sum_{i=1}^n \hat{f}_i(p)$$



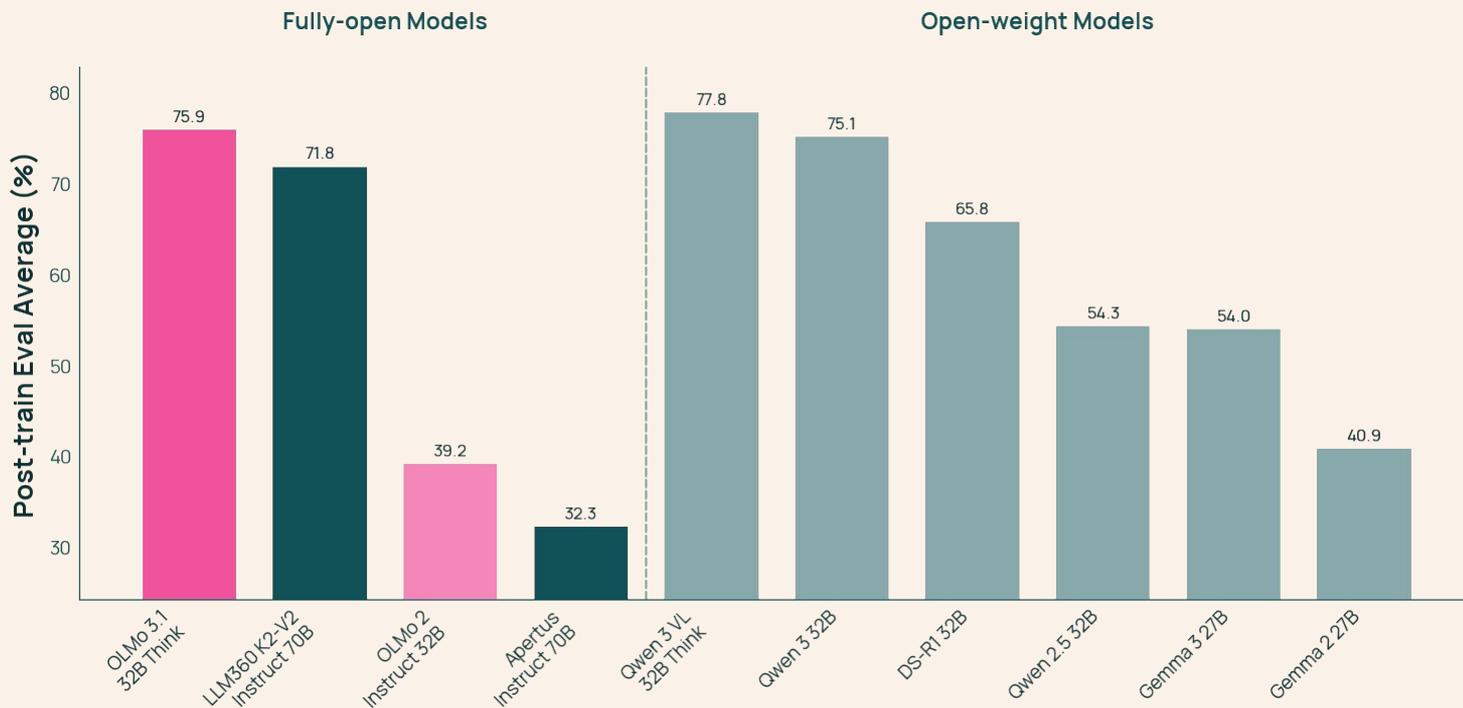


- > Data Mixing with Olmix
- > Data for Olmo 3



## > Putting it together: Olmo 3

# Olmo 3 Model Flow



# Olmo 3 Model Flow

