

## A Scaling Laws

Every parameter goes through a multiplication and addition per input unit in the forward pass, and twice that in the backward pass resulting in 6 flops per parameter per input. For the attention mechanism, the  $QK^T$  operation dominates the computational cost, requiring 2 FLOPs (multiply and add) per dimension for each query-key pair. With  $d$  dimensions,  $L$  layers, and a sequence length of  $S$ , this creates  $S$  dot products per layer per input unit. Accounting for both forward and backward passes ( $3\times$  multiplier), we get  $6dLS$  FLOPs total. This term becomes particularly significant at smaller scales where attention costs outweigh the linear parameter costs as Bi et al. [11] already point out.

Notice how the batch size is not simply a difference in constant factor but also in the exponent. In our experiments, we find that many values of batch size and learning rates are possible and that optimal models for a given compute budget lie roughly on a line in BSZ/LR space such that both grow linearly with respect to each other. This of course is only valid for a certain range of values above which the model becomes unstable and loses performance. Our hypothesis is that simply scaling the batch size such that it equals  $k \times \text{BSZ}_{\text{BPE}}$  results in a model that is beyond that limit.

## B Regular expression

To be concrete, the regular expression used to define Stage 1 pooling is shown below:

```
(\p{L}{1,16}) | \p{N}{1,3} | ?([^\s\p{L}\p{N}]){1,3}+[\r\n]* | \s*[\r\n] | \s+(?!$) | \s+
```

Each component of the regex serves a distinct role:

- **Letters (1–16 characters)**: captures typical alphabetic words.
- **Numbers (1–3 digits)**: groups numerical tokens.
- **Punctuation (1–3 non-alphanumeric chars)**: handles symbol groups and optional line breaks.
- **Line breaks**: captures `\r\n` combinations and surrounding whitespace.
- **Trailing whitespace (non-followed by a non-space)**: captures text boundaries.
- **General whitespace**: handles space separation.

## C Ablation

### C.1 Pooling and Upsampling

We describe here the different pooling and upsampling strategies explored during our experiments. While all pooling methods yielded comparable results, they offer different trade-offs in complexity and expressiveness.

**Simple Pooling.** This is the method used in our main experiments. We directly select the positions indicated by the splitting function and retain only those tokens.

**Cross-Attention Pooling.** A cross-attention layer is applied between the original sequence and the pooled tokens. This allows the downsampled representation to aggregate information flexibly from the full input.

**Average Pooling.** Tokens within each segment defined by the splitting function are averaged to produce a single pooled representation.

**Memory Layers [28].** Motivated by the concern that pooling might limit output diversity compared to embedding-table, we experimented with appending a memory layer after pooling. This layer retrieves learned embeddings based on the pooled inputs, potentially reintroducing back the diversity.

**Simple Upsampling.** Pooled tokens are inserted back into their original positions in the sequence, and additional context is recovered via skip connections. Earlier-layer features complement the compressed representations, and attention layers help propagate information across the sequence.

**Cross-Attention Upsampling.** A cross-attention layer is applied where each upsampled token attends to the pooled representation. This mechanism allows the model to flexibly decompress higher-level abstract representations, effectively extracting contextual information to reconstruct the outputs.

**Repeat Upsampling.** Inspired by nearest-neighbor upscaling in computer vision, each token in the

Table 4: Comparison between the different upsampling tools. Notice that AU-Net 3 stages is much more sensitive to upsampling.

Model	Hswg	Arc_E	Arc_C	PIQA	SIQA	Race_MR	Race_HW	Winog	NQ	TQA
<b>Dim=2048 (1B model), 60B tokens (data-to-model ratio of 10)</b>										
AU-Net 2 Simple	62.9 ± 0.9	64.9 ± 1.9	<b>35.5</b> ± 2.7	73.4 ± 2.0	45.7 ± 2.2	54.1 ± 2.6	39.3 ± 1.6	60.5 ± 2.7	7.7 ± 0.9	16.6 ± 0.7
AU-Net 2 Average Pool	62.5 ± 1.0	61.5 ± 2.0	35.4 ± 2.7	72.9 ± 2.0	44.7 ± 2.2	52.2 ± 2.6	36.9 ± 1.6	60.4 ± 2.7	7.2 ± 0.8	15.5 ± 0.7
AU-Net 2 Memory Layer	62.8 ± 0.9	<b>66.5</b> ± 1.9	34.4 ± 2.7	72.2 ± 2.0	45.3 ± 2.2	<b>55.2</b> ± 2.6	38.7 ± 1.6	61.3 ± 2.7	8.0 ± 0.9	16.6 ± 0.7
AU-Net 2 Repeat Up	<b>64.2</b> ± 0.9	64.4 ± 1.9	35.2 ± 2.7	<b>74.4</b> ± 2.0	<b>46.1</b> ± 2.2	53.9 ± 2.6	39.0 ± 1.6	61.7 ± 2.7	<b>8.8 ±</b> 0.9	<b>20.4</b> ± 0.7
AU-Net 2 Multi-Linear	<b>63.5</b> ± 0.9	64.4 ± 1.9	35.3 ± 2.7	74.0 ± 2.0	45.3 ± 2.2	55.1 ± 2.5	<b>39.6</b> ± 1.6	<b>62.6</b> ± 2.6	8.3 ± 0.9	18.4 ± 0.7
AU-Net 3 Simple	60.6 ± 1.0	60.8 ± 2.0	32.3 ± 2.7	72.1 ± 2.1	46.3 ± 2.2	53.1 ± 2.6	38.6 ± 1.6	62.0 ± 2.6	6.0 ± 0.8	13.3 ± 0.6
AU-Net 3 Multi-Linear	<b>66.0</b> ± 0.9	<b>64.1</b> ± 1.9	<b>35.7</b> ± 2.7	<b>75.1</b> ± 2.0	<b>45.9</b> ± 2.2	<b>55.4</b> ± 2.5	<b>39.3</b> ± 1.6	<b>64.0</b> ± 2.6	<b>7.3 ±</b> 0.9	<b>18.7</b> ± 0.7

compressed sequence is repeated a variable number of times, as determined by the splitting function. For this strategy to remain competitive during training, it is important to include local positional biases within each repeated segment.

**Multi-Linear Upsampling.** Each pooled token is transformed using a different linear projection for each position in the target segment. This allows upsampled tokens to vary based on their relative position while remaining conditioned on the same source. This method is used in our main experiments due to its favorable balance between simplicity and expressiveness.

## C.2 Layer Allocations

To evaluate the impact of distributing different numbers of layers across stages, we conducted ablations varying the layer allocation strategy. The first stage (byte level) is fixed to three layers for all models. As shown in table 5, we allocate a certain percentage of the total layers to the final stage (stage 3), while ensuring that each intermediate stage retains at least three layers.

We report results for several allocation schemes, and retain the 75% variant—where 75% of the layers are allocated to the final stage—as the default configuration in the main paper.

Table 5: Comparison between the different layer percentages in the last stage (the third one).

Model	Hswg	Arc_E	Arc_C	PIQA	SIQA	Race_MR	Race_HW	Winog	NQ	TQA
<b>Dim=2048 (1B model), 60B tokens (data-to-model ratio of 10)</b>										
AUNet 3 (25%)	65.3 ± 0.9	63.3 ± 1.9	36.0 ± 2.8	74.2 ± 2.0	46.8 ± 2.2	54.8 ± 2.6	38.7 ± 1.6	63.4 ± 2.6	8.9 ± 0.9	21.0 ± 0.8
AUNet 3 (50%)	66.0 ± 0.9	64.1 ± 1.9	35.7 ± 2.7	75.1 ± 2.0	45.9 ± 2.2	55.4 ± 2.6	39.3 ± 1.6	64.0 ± 2.7	7.3 ± 0.9	18.7 ± 0.7
AUNet 3 (75%)	67.4 ± 0.9	65.9 ± 1.9	36.7 ± 2.7	75.5 ± 2.0	46.9 ± 2.2	55.4 ± 2.5	40.5 ± 1.6	64.2 ± 2.6	9.6 ± 1.0	22.6 ± 0.8

## D Hyperparameters

As explained in section 2.3, we use a specific batch size and learning rate for each compute budget and architecture. Aside from this, all other hyperparameters remain fixed. A summary table of all hyperparameters can be found in table 6. We use sequence packing for dataloading during training along with FSDP. The complete details can be seen in appendix G.

Model	LR	BSZ	w.d.	lr min	grad clip	seqlen	total tokens
BPE	$19.3C^{-0.177}$	$29.9C^{0.231}$	0.1	$0.01 \times \text{lr\_max}$	0.2	2048	$(F_{\text{model}} / \text{token})^2 \gamma_{\text{token}}$
AU-Net	$6.6C^{-0.176}$	$0.66C^{0.321}$	0.1	$0.01 \times \text{lr\_max}$	0.2	8192	$(F_{\text{model}} / \text{byte})^2 (20.7936 \gamma_{\text{token}})$

Table 6: Summary of all hyperparameters. *w.d.* stands for weight decay.  $\gamma_{\text{tokens}}$  corresponds to the data-to-model ratio and is reported in bold in each result table, alongside the budget  $C$ . Flops per token/byte are detailed in the table of appendix G. Warmup spans 10% of the total training steps, and we employ a cosine learning rate scheduler. The total number of steps is computed as  $\frac{\text{total\_tokens}}{\text{BSZ}}$ .

We also add hyperparameters sweep details as described in section 2.3 for both AU-Net and BPE:

Hyperparameters vs. Compute (highlighted = within 1.0% of best)

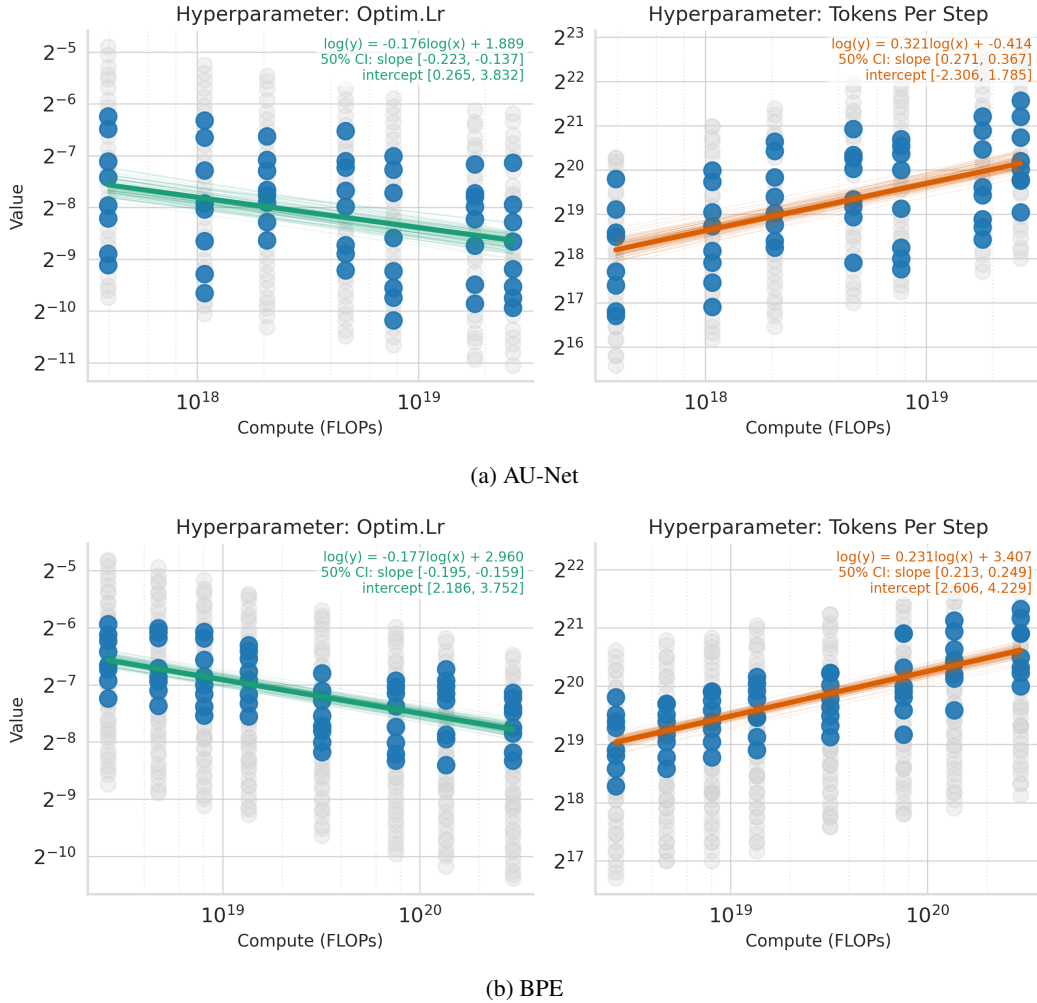


Figure 4: Hyperparameter sweep with fitting: learning rate (left) and batch size (right) for both AU-Net (top) and BPE (bottom).

Figure 4 summarizes the hyperparameter sweeps on DCLM across model scales. We find a broad region of stable configurations rather than a single narrow optimum, indicating that AU-Net trains reliably over a wide range of learning rates (LR) and batch sizes (BSZ). Smaller AU-Net variants remain stable even at high learning rates (up to  $1.5 \times 10^{-2}$ ), suggesting strong tolerance to aggressive optimization. Compared to the baseline BPE Transformer, AU-Net follows a similar LR compute scaling trend but with a lower intercept, reflecting slightly smaller effective learning rates but that decay at the same rate. Batch size increases consistently with compute, starting from smaller values and scaling smoothly, which indicates good parallelization potential at larger scales.

## E CUTE Benchmark Detailed results

We evaluate both the 7.5B BPE baseline and AU-Net 2 on the CUTE benchmark [17], which tests a model’s ability to manipulate both words and characters. As shown in Table 7, our byte-level model performs better on character-level tasks, while the BPE baseline takes the lead on word-level ones. This reflects a natural trade-off: tokenizer-based models operate on word-like units, making them less sensitive to character structure, whereas byte-level models handle characters explicitly.

This contrast highlights a key design trade-off. Byte-level models are more flexible with unseen or morphologically rich inputs, while tokenized models benefit from stronger word-level priors. Surprisingly, despite lacking explicit character access, BPE models still perform well on spelling and reverse spelling tasks, suggesting that such skills can emerge from token-level patterns with enough capacity and data.

Table 7: Accuracy of BPE and AU-Net on word-level and letter-level tasks in CUTE.

CUTE (EM)	Rand	BPE		AU-Net 2	
		Word	Char	Word	Char
Spell	0.0	-	91.5	-	<b>97.3</b>
Inverse spell	0.0	-	80.6	-	<b>91.7</b>
Contains	50.0	<b>69.9</b>	<b>66.7</b>	61.3	59.8
Delete	0.0	<b>29.6</b>	16.4	20.6	<b>22.3</b>
Insert	0.0	<b>15.9</b>	<b>9.6</b>	6.5	7.8
Substitute	0.0	<b>37.5</b>	7.6	21.2	<b>12.3</b>
Swap	0.0	<b>5.5</b>	1.6	3.3	<b>1.9</b>
Sem/Ortho	50.0	66.0	40.6	<b>75.1</b>	<b>48.1</b>
<b>Average</b>	12.5	<b>36.9</b>	39.3	33.1	<b>42.7</b>

## F Evaluation Benchmarks Details

Model	Hllswg	Arc_E	Arc_C	Boolq	CSQA	MMLU	OBQA	PIQA	SIQA	Race_M	Race_H	Winog.	NQ	TQA	GSM8k
<b>Dim=2048 (1B model), 60B tokens (data-to-model ratio of 10)</b>															
Mamba bytes	63.0 ±0.9	60.3 ±2.0	33.6 ±2.7	61.2 ±1.7	19.6 ±2.3	25.3 ±0.7	<b>38.4</b> ±4.2	75.1 ±2.0	45.8 ±2.2	46.7 ±2.6	35.2 ±1.6	59.1 ±2.7	8.2 ± 0.9	21.2 ±0.7	2.1 ± 0.8
Transformer bytes	63.0 ±0.9	61.2 ±2.0	34.7 ±2.7	60.7 ±1.7	20.0 ±2.3	24.7 ±0.7	<b>38.4</b> ±4.3	74.5 ±2.0	47.1 ±2.2	52.1 ±2.5	37.7 ±1.6	58.2 ±2.8	8.8 ± 0.9	21.4 ±0.8	2.5 ± 0.8
Transformer bytes[:4]	63.5 ±0.9	63.0 ±2.0	36.0 ±2.7	63.4 ±1.7	18.8 ±2.2	25.1 ±0.7	39.0 ±4.2	73.0 ±2.1	46.4 ±2.2	53.5 ±2.5	40.1 ±1.6	62.3 ±2.7	6.8 ± 0.8	17.2 ±0.7	2.6 ± 0.9
Transformer bytes[:5]	60.0 ±1.0	60.0 ±2.0	34.8 ±2.7	61.4 ±1.7	17.9 ±2.1	23.9 ±0.7	37.2 ±4.3	72.0 ±2.0	45.4 ±2.2	51.5 ±2.6	39.4 ±1.6	57.2 ±2.8	5.6 ± 0.8	14.7 ±0.7	2.0 ± 0.8
Transformer bytes[:6]	58.6 ±1.0	59.5 ±2.0	32.4 ±2.7	60.6 ±1.7	18.7 ±2.2	24.9 ±0.7	35.6 ±4.2	70.0 ±2.1	44.5 ±2.2	51.3 ±2.5	37.4 ±1.6	59.3 ±2.7	4.6 ± 0.7	12.3 ±0.6	2.3 ± 0.8
AUNet 2	64.2 ±0.9	64.4 ±1.9	35.2 ±2.7	62.0 ±1.7	<b>20.1</b> ±2.2	24.5 ±0.7	36.8 ±4.2	74.4 ±2.0	46.1 ±2.3	53.9 ±2.5	39.0 ±1.6	61.7 ±2.7	8.8 ± 0.9	20.4 ±0.7	2.7 ± 0.9
AUNet 3	<b>67.4</b> ±0.9	65.9 ±1.9	36.7 ±2.8	61.7 ±1.7	19.7 ±2.2	25.6 ±0.7	38.2 ±4.3	<b>75.5</b> ±2.0	46.9 ±2.2	55.4 ±2.5	<b>40.5</b> ±1.6	<b>64.2</b> ±2.6	<b>9.6</b> ± 1.0	<b>22.6</b> ±1.5	2.3 ± <b>3.5</b> ±
AUNet 4	66.4 ±0.9	<b>67.4</b> ±1.9	<b>37.0</b> ±2.8	<b>63.3</b> ±1.7	18.2 ±2.1	<b>25.9</b> ±0.7	38.2 ±4.3	74.5 ±2.0	<b>47.6</b> ±2.3	<b>55.6</b> ±2.6	39.3 ±1.6	62.0 ±2.6	5.1 ± 0.7	15.5 ±0.7	<b>3.5</b> ± 1.0
Transformer BPE	63.6 ±0.9	62.8 ±2.0	36.5 ±2.7	62.6 ±1.6	18.8 ±2.2	25.5 ±0.7	37.4 ±4.3	75.1 ±2.0	45.2 ±2.2	53.9 ±2.6	39.3 ±1.6	61.6 ±2.7	8.8 ± 0.9	26.3 ±0.8	2.3 ± 0.8
<b>Dim=2048 (1B model), 370B tokens (data-to-model ratio of 40)</b>															
AUNet 2	69.9 ±0.9	68.6 ±1.9	38.9 ±2.8	64.3 ±1.7	20.8 ±2.3	27.9 ±0.7	39.6 ±4.4	76.8 ±1.9	46.6 ±2.2	57.7 ±2.5	42.8 ±1.7	64.6 ±2.6	13.0 ±1.1	32.5 ±0.9	3.0 ± 0.9
AUNet 3	72.9 ±0.9	72.3 ±1.8	<b>43.3</b> ±2.9	61.8 ±1.7	19.7 ±2.3	27.5 ±0.7	<b>41.6</b> ±4.3	<b>78.1</b> ±1.9	47.1 ±2.3	58.8 ±2.6	43.3 ±1.7	<b>68.7</b> ±2.6	<b>15.3</b> ±1.2	<b>39.1</b> ±0.9	3.7 ± 1.0
AUNet 4	<b>73.7</b> ±0.9	<b>72.6</b> ±1.8	43.2 ±2.8	62.0 ±1.7	<b>23.1</b> ±2.4	<b>31.1</b> ±0.8	40.8 ±4.2	78.0 ±1.9	<b>47.6</b> ±2.2	<b>59.0</b> ±2.6	<b>43.9</b> ±1.7	67.2 ±2.6	14.0 ±1.1	35.5 ±0.9	<b>5.3</b> ± 1.2
Transformer BPE	70.2 ±0.9	68.6 ±1.9	38.5 ±2.8	<b>62.9</b> ±1.7	21.8 ±2.3	26.3 ±0.7	40.6 ±4.3	76.9 ±1.9	46.2 ±2.2	56.7 ±2.6	41.8 ±1.6	65.4 ±2.6	13.6 ±1.1	37.2 ±0.9	4.4 ± 1.1
<b>Dim=4096 (8B model), 200B tokens (data-to-model ratio of 5)</b>															
AUNet 2 7B	<b>79.1</b> ±0.8	<b>80.0</b> ±1.6	<b>51.2</b> ±2.9	<b>68.3</b> ±1.6	<b>63.6</b> ±2.7	<b>50.0</b> ±0.8	<b>45.0</b> ±4.4	<b>80.1</b> ±1.8	<b>50.0</b> ±2.3	<b>61.9</b> ±2.5	<b>44.4</b> ±1.7	<b>72.2</b> ±2.4	<b>22.1</b> ±1.3	50.9 ±0.9	10.0 ±1.6
Transformer 7B	77.3 ±0.8	74.3 ±1.8	49.5 ±2.9	63.8 ±1.7	63.2 ±2.7	48.4 ±0.8	43.6 ±4.3	80.0 ±1.8	48.4 ±2.2	60.3 ±2.5	43.6 ±1.6	70.5 ±2.5	21.1 ±1.3	<b>51.1</b> ±0.9	<b>10.7</b> ±1.7
LLaMa 3.1 (15T)	80.7 ±0.8	83.3 ±1.5	54.8 ±2.8	75.0 ±1.5	74.6 ±2.5	65.4 ±0.8	45.4 ±4.4	80.8 ±1.8	49.6 ±2.3	65.3 ±2.5	49.4 ±1.7	74.5 ±2.4	29.1 ±1.5	64.4 ±0.9	54.7 ±2.7

Table 8: Performance of AU-Net on many benchmarks.

## G List of Models

This appendix provides detailed configuration parameters for all models used in the experiments, organized into three categories for clarity.

Table 9: Model architecture parameters including dimensions, layers, and FFN sizes. Semicolons separated values for different stages in hierarchical models. The values shown in the “Layers” column correspond to the number of layers on both sides of the U-Net, except for the final value on the right, which represents the central stage. For example, AU-Net 3 1B row has  $3 \times 2 + 3 \times 2 + 18 = 30$  layers in total.

Name	Dim	Layers	Head Dim	FFN Dim
Transformer bytes 1B	2048	25	128	5632
Mamba bytes 1B	2048	25	64	5632
Transformer 1B BPE	2048	25	128	5632
AUNet 2 1B	512; 2048	3; 25	64; 128	1536; 5632
AUNet 3 1B	512; 2048; 3072	3; 3; 18	64; 128; 128	1536; 5632; 8192
AUNet 4 1B	512; 2048; 3072; 4608	3; 3; 4; 10	64; 128; 128; 128	1536; 5632; 8192; 12288
Transformer 1B dm8 BPE	2048	25	128	5632
AUNet 2 1B dm8	512; 2048	3; 25	64; 128	1536; 5632
AUNet 3 1B dm8	512; 2048; 3072	3; 3; 18	64; 128; 128	1536; 5632; 8192
AUNet 4 1B dm8	512; 2048; 3072; 4608	3; 3; 3; 12	64; 128; 128; 128	1536; 5632; 8192; 12288
Transformer 7B dm1	4096	32	128	11008
AUNet 2 7B dm1	1024; 4096	3; 32	64; 128	4096; 14336
Scaling baseline 1e19	1024	12	128	2816
Scaling baseline 2e19	1152	13	128	3072
Scaling baseline 4e19	1280	14	128	3584
Scaling baseline 8e19	1536	15	128	4096
Scaling baseline 1e20	1664	17	128	4608
Scaling baseline 3e20	1792	20	128	4864
Scaling baseline 5e20	2048	21	128	5632
Scaling baseline 1e21	2304	24	128	6144
Scaling baseline 2e21	2560	26	128	6912
Scaling baseline 3e21	2816	29	128	7680
Scaling baseline 6e21	3072	34	128	8192
Scaling baseline 1e22	3456	37	128	9216
Scaling AUNet 2 1e19	256; 1024	3; 11	64; 128	768; 2816
Scaling AUNet 2 2e19	256; 1152	3; 13	64; 128	768; 3072
Scaling AUNet 2 4e19	256; 1280	3; 14	64; 128	768; 3584
Scaling AUNet 2 8e19	384; 1536	3; 14	64; 128	1024; 4096
Scaling AUNet 2 1e20	384; 1536	3; 19	64; 128	1024; 4096
Scaling AUNet 2 3e20	512; 1920	3; 17	64; 128	1536; 5120
Scaling AUNet 2 5e20	512; 2048	3; 21	64; 128	1536; 5632
Scaling AUNet 2 9e20	512; 2304	3; 24	64; 128	1536; 6144
Scaling AUNet 2 2e21	640; 2560	3; 26	64; 128	1792; 6912
Scaling AUNet 2 3e21	640; 2688	3; 33	64; 128	1792; 7168
Scaling AUNet 2 6e21	768; 3200	3; 32	64; 128	2048; 8704
Scaling AUNet 2 1e22	896; 3584	3; 35	64; 128	2560; 9728
Scaling AUNet 3 1e19	256; 1024; 1536	3; 3; 4	64; 128; 128	768; 2816; 4096
Scaling AUNet 3 2e19	256; 1152; 1792	3; 3; 5	64; 128; 128	768; 3072; 4864
Scaling AUNet 3 5e19	256; 1280; 1920	3; 3; 7	64; 128; 128	768; 3584; 5120
Scaling AUNet 3 7e19	256; 1280; 1920	3; 3; 10	64; 128; 128	768; 3584; 5120
Scaling AUNet 3 2e20	384; 1536; 2304	3; 3; 10	64; 128; 128	1024; 4096; 6144
Scaling AUNet 3 3e20	384; 1536; 2304	3; 3; 15	64; 128; 128	1024; 4096; 6144
Scaling AUNet 3 5e20	512; 1920; 2816	3; 3; 13	64; 128; 128	1536; 5120; 7680
Scaling AUNet 3 9e20	512; 2048; 3072	3; 3; 16	64; 128; 128	1536; 5632; 8192
Scaling AUNet 3 2e21	512; 2304; 3456	3; 3; 18	64; 128; 128	1536; 6144; 9216
Scaling AUNet 3 3e21	640; 2560; 3840	3; 3; 21	64; 128; 128	1792; 6912; 10240
Scaling AUNet 3 6e21	640; 2688; 4096	3; 3; 26	64; 128; 128	1792; 7168; 11008
Scaling AUNet 3 1e22	768; 3200; 4864	3; 3; 26	64; 128; 128	2048; 8704; 13056

Table 10: Training configuration including computational costs, steps, batch sizes, and tokenization

Name	Total FLOPs	Tokens/Step	FLOPs/Token	Steps	G/Acc	Batch Size	Seq Len	NGpus	Tokenizer
Transformer bytes 1B	nan	nan	nan	60000	1	4	4096	NaN	bytes
Mamba bytes 1B	nan	nan	nan	60000	1	4	4096	NaN	bytes
Transformer 1B BPE	6.6e20	1.0e06	1.1e10	60000	1	4	4096	64	tiktoken
AUNet 2 1B	5.1e20	1.6e06	1.8e09	180000	1	12	8192	16	bytes
AUNet 3 1B	6.7e20	2.8e06	2.3e09	105000	1	14	8192	24	bytes
AUNet 4 1B	8.0e20	2.8e06	2.8e09	105000	1	14	8192	24	bytes
Transformer 1B dm8 BPE	3.6e21	1.2e06	9.9e09	310000	1	9	2048	64	tiktoken
AUNet 2 1B dm8	3.2e21	1.8e06	1.8e09	950000	1	7	8192	32	bytes
AUNet 3 1B dm8	4.0e21	5.8e06	2.3e09	300000	1	11	8192	64	bytes
AUNet 4 1B dm8	5.0e21	5.8e06	2.9e09	300000	1	11	8192	64	bytes
Transformer 7B dm1	9.5e21	2.1e06	4.5e10	100000	1	2	4096	256	tiktoken
AUNet 2 7B dm1	1.2e22	4.2e06	1.0e10	277834	1	1	8192	128	bytes
Scaling baseline 1e19	2.0e19	7.7e05	1.9e09	14008	15	25	2048	1	tiktoken
Scaling baseline 2e19	3.3e19	8.8e05	2.3e09	16120	18	24	2048	1	tiktoken
Scaling baseline 4e19	5.6e19	9.9e05	2.9e09	19438	22	22	2048	1	tiktoken
Scaling baseline 8e19	1.1e20	1.2e06	4.0e09	24252	15	19	2048	2	tiktoken
Scaling baseline 1e20	2.0e20	1.4e06	5.1e09	28160	13	17	2048	3	tiktoken
Scaling baseline 3e20	3.3e20	1.6e06	6.5e09	32466	11	14	2048	5	tiktoken
Scaling baseline 5e20	6.0e20	1.8e06	8.6e09	39573	12	12	2048	6	tiktoken
Scaling baseline 1e21	1.1e21	2.1e06	1.2e10	46980	4	8	2048	32	tiktoken
Scaling baseline 2e21	2.0e21	2.4e06	1.5e10	55900	4	9	2048	32	tiktoken
Scaling baseline 3e21	3.6e21	2.8e06	2.0e10	64711	3	7	2048	64	tiktoken
Scaling baseline 6e21	6.5e21	3.1e06	2.7e10	77520	4	6	2048	64	tiktoken
Scaling baseline 1e22	1.2e22	3.9e06	3.6e10	84915	3	5	2048	128	tiktoken
Scaling AUNet 2 1e19	1.1e19	8.7e05	2.4e08	56240	2	53	8192	1	bytes
Scaling AUNet 2 2e19	2.2e19	1.1e06	3.2e08	63457	3	43	8192	1	bytes
Scaling AUNet 2 4e19	3.7e19	1.3e06	4.2e08	68794	4	39	8192	1	bytes
Scaling AUNet 2 8e19	7.6e19	1.6e06	6.0e08	79862	6	32	8192	1	bytes
Scaling AUNet 2 1e20	1.3e20	1.8e06	7.9e08	89768	4	28	8192	2	bytes
Scaling AUNet 2 3e20	2.6e20	2.4e06	1.1e09	99036	4	24	8192	3	bytes
Scaling AUNet 2 5e20	5.0e20	2.9e06	1.5e09	108807	4	18	8192	5	bytes
Scaling AUNet 2 9e20	9.3e20	3.7e06	2.1e09	119909	1	14	8192	32	bytes
Scaling AUNet 2 2e21	1.7e21	4.2e06	2.9e09	141463	2	8	8192	32	bytes
Scaling AUNet 2 3e21	3.1e21	5.2e06	3.9e09	153594	2	10	8192	32	bytes
Scaling AUNet 2 6e21	5.9e21	6.3e06	5.3e09	176692	1	8	8192	96	bytes
Scaling AUNet 2 1e22	1.1e22	7.9e06	7.3e09	193096	1	6	8192	160	bytes
Scaling AUNet 3 1e19	1.3e19	9.0e05	2.5e08	56992	2	55	8192	1	bytes
Scaling AUNet 3 2e19	2.4e19	1.1e06	3.4e08	64347	3	45	8192	1	bytes
Scaling AUNet 3 5e19	4.7e19	1.4e06	4.8e08	71962	4	42	8192	1	bytes
Scaling AUNet 3 7e19	7.3e19	1.6e06	5.9e08	79222	5	38	8192	1	bytes
Scaling AUNet 3 2e20	1.5e20	2.0e06	8.6e08	90845	4	30	8192	2	bytes
Scaling AUNet 3 3e20	2.7e20	2.4e06	1.1e09	100198	4	24	8192	3	bytes
Scaling AUNet 3 5e20	5.2e20	2.9e06	1.6e09	113583	2	22	8192	8	bytes
Scaling AUNet 3 9e20	9.2e20	3.7e06	2.1e09	119374	1	14	8192	32	bytes
Scaling AUNet 3 2e21	1.7e21	4.2e06	2.9e09	141594	1	16	8192	32	bytes
Scaling AUNet 3 3e21	3.3e21	5.2e06	4.0e09	158941	1	10	8192	64	bytes
Scaling AUNet 3 6e21	6.0e21	6.3e06	5.4e09	177921	1	8	8192	96	bytes
Scaling AUNet 3 1e22	1.2e22	8.4e06	7.6e09	187518	1	4	8192	256	bytes

Table 11: Optimization hyperparameters including learning rates, weight decay, and scheduler settings.

Name	LR	WD	$\beta_1$	$\beta_2$	Scheduler	Warmup
Transformer bytes 1B	0.003	0.033	0.9	0.95	cosine	5000
Mamba bytes 1B	0.003	0.033	0.9	0.95	cosine	5000
Transformer 1B BPE	0.003	0.033	0.9	0.95	cosine	5000
AUNet 2 1B	0.00165	0.1	0.9	0.95	cosine	10000
AUNet 3 1B	0.0015	0.1	0.9	0.95	cosine	10000
AUNet 4 1B	0.0015	0.1	0.9	0.95	cosine	20000
Transformer 1B dm8 BPE	0.001	0.1	0.9	0.95	cosine	2000
AUNet 2 1B dm8	0.00094	0.1	0.9	0.95	cosine	10000
AUNet 3 1B dm8	0.0011	0.1	0.9	0.95	cosine	10000
AUNet 4 1B dm8	0.0011	0.1	0.9	0.95	cosine	10000
Transformer 7B dm1	0.001	0.05	0.9	0.95	cosine	10000
AUNet 2 7B dm1	0.000818	0.1	0.9	0.95	cosine	5000
Scaling baseline 1e19	0.008152	0.1	0.9	0.95	cosine	2000
Scaling baseline 2e19	0.007378	0.1	0.9	0.95	cosine	2000
Scaling baseline 4e19	0.006633	0.1	0.9	0.95	cosine	2000
Scaling baseline 8e19	0.005788	0.1	0.9	0.95	cosine	2000
Scaling baseline 1e20	0.005204	0.1	0.9	0.95	cosine	2000
Scaling baseline 3e20	0.004693	0.1	0.9	0.95	cosine	2000
Scaling baseline 5e20	0.0042	0.1	0.9	0.95	cosine	2000
Scaling baseline 1e21	0.003722	0.1	0.9	0.95	cosine	2000
Scaling baseline 2e21	0.003357	0.1	0.9	0.95	cosine	2000
Scaling baseline 3e21	0.003018	0.1	0.9	0.95	cosine	2000
Scaling baseline 6e21	0.002701	0.1	0.9	0.95	cosine	2000
Scaling baseline 1e22	0.002416	0.1	0.9	0.95	cosine	2000
Scaling AUNet 2 1e19	0.002923	0.1	0.9	0.95	cosine	10000
Scaling AUNet 2 2e19	0.002615	0.1	0.9	0.95	cosine	10000
Scaling AUNet 2 4e19	0.002377	0.1	0.9	0.95	cosine	10000
Scaling AUNet 2 8e19	0.002096	0.1	0.9	0.95	cosine	10000
Scaling AUNet 2 1e20	0.001906	0.1	0.9	0.95	cosine	10000
Scaling AUNet 2 3e20	0.001685	0.1	0.9	0.95	cosine	10000
Scaling AUNet 2 5e20	0.001507	0.1	0.9	0.95	cosine	10000
Scaling AUNet 2 9e20	0.001348	0.1	0.9	0.95	cosine	10000
Scaling AUNet 2 2e21	0.001214	0.1	0.9	0.95	cosine	10000
Scaling AUNet 2 3e21	0.00109	0.1	0.9	0.95	cosine	10000
Scaling AUNet 2 6e21	0.0009731	0.1	0.9	0.95	cosine	10000
Scaling AUNet 2 1e22	0.0008719	0.1	0.9	0.95	cosine	10000
Scaling AUNet 3 1e19	0.002872	0.1	0.9	0.95	cosine	10000
Scaling AUNet 3 2e19	0.002561	0.1	0.9	0.95	cosine	10000
Scaling AUNet 3 5e19	0.002279	0.1	0.9	0.95	cosine	10000
Scaling AUNet 3 7e19	0.00211	0.1	0.9	0.95	cosine	10000
Scaling AUNet 3 2e20	0.001852	0.1	0.9	0.95	cosine	10000
Scaling AUNet 3 3e20	0.001678	0.1	0.9	0.95	cosine	10000
Scaling AUNet 3 5e20	0.001496	0.1	0.9	0.95	cosine	10000
Scaling AUNet 3 9e20	0.001351	0.1	0.9	0.95	cosine	10000
Scaling AUNet 3 2e21	0.001213	0.1	0.9	0.95	cosine	10000
Scaling AUNet 3 3e21	0.001077	0.1	0.9	0.95	cosine	10000
Scaling AUNet 3 6e21	0.0009707	0.1	0.9	0.95	cosine	10000
Scaling AUNet 3 1e22	0.0008612	0.1	0.9	0.95	cosine	10000



## H Licences

Table 12: Licensing Summary for Language Model Training and Evaluation Datasets

<b>Dataset</b>	<b>License</b>
DCLM	MIT
HellaSwag	MIT
ARC-Easy	CC-BY-SA-4.0
ARC-Challenge	CC-BY-SA-4.0
MMLU	MIT
Natural Questions	Apache-2.0
TriviaQA	Apache-2.0
GSM8K	MIT
BoolQ	CC-SA-3.0
CommonsenseQA	MIT
OpenBookQA	Apache-2.0
PIQA	Not available
Social IQA	MIT
RACE-M	Custom research only
RACE-H	Custom research only
WinoGrande	Apache-2.0/CC-BY
CUTE	MIT
FLORES-200	CC-BY-SA-4.0
MMLU Multilingual	MIT
LLaMA 3 Tokenizer	Meta Community