PLOS Computational Biology

RESEARCH ARTICLE

A curriculum learning approach to training antibody language models

Sarah M. Burbach[1], **Bryan Briney**[1,2,3,4,5]*

1 Department of Immunology and Microbiology, The Scripps Research Institute, La Jolla, California, United States of America, **2** Center for Viral Systems Biology, The Scripps Research Institute, La Jolla, California, United States of America, **3** Multi-omics Vaccine Evaluation Consortium, The Scripps Research Institute, La Jolla, California, United States of America, **4** Scripps Consortium for HIV/AIDS Vaccine Development, The Scripps Research Institute, La Jolla, California, United States of America, **5** San Diego Center for AIDS Research, The Scripps Research Institute, La Jolla, California, United States of America

* briney@scripps.edu

OPEN ACCESS

Citation: Burbach SM, Briney B (2025) A curriculum learning approach to training antibody language models. PLoS Comput Biol 21(9): e1013473. https://doi.org/10.1371/journal.pcbi.1013473

Editor: Chaok Seok, Seoul National University, KOREA, REPUBLIC OF

Received: March 5, 2025

Accepted: August 28, 2025

Published: September 11, 2025

Data availability statement: The code used for model training and evaluation is available on GitHub (github.com/brineylab/curriculum-paper). The training data and model weights for CurrAb are available on Zenodo (doi.org/10.5281/zenodo.14661302). The 650M-parameter mixed models, including CurrAb, are

Abstract

There is growing interest in pre-training antibody language models (**AbLMs**) with a mixture of unpaired and natively paired sequences, seeking to combine the proven benefits of training with natively paired sequences with the massive scale of unpaired antibody sequence datasets. However, given the novelty of this strategy, the field lacks a systematic evaluation of data processing methods and training strategies that maximize the benefits of mixed training data while accommodating the significant imbalance in the size of existing paired and unpaired datasets. Here, we introduce a method of curriculum learning for AbLMs, which facilitates a gradual transition from unpaired to paired sequences during training. We optimize this method and compare it to other data sampling strategies for AbLMs, including a constant mix and a fine-tuning approach. We observe that the curriculum and constant approaches show improved performance compared to the fine-tuning approach in large-scale models, likely due to their ability to prevent catastrophic forgetting and slow overfitting. Finally, we show that a 650M-parameter curriculum model, CurrAb, outperforms existing mixed AbLMs in downstream residue prediction and classification tasks.

Author summary

Antibodies are essential components of our adaptive immune system, but traditional methods of antibody engineering are both costly and time consuming. Recent advances in artificial intelligence have increased interest in utilizing language models to accelerate this process in silico. To this end, antibody-specific language models can be trained using existing datasets of unpaired sequences (containing a single chain) and paired antibody sequences (containing both chains, as found in nature). Unpaired sequences are more abundant, but paired

uploaded on Hugging Face (huggingface.co/collections/brineylab/curriculum-paper-685b08a4b6986df7c5a5e3c4).

Funding: This work was funded by the National Institutes of Health (P01-AI177683, U19-AI135995, R01-AI171438, P30-AI036214, and UM1-AI144462) and the Pendleton Foundation. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: I have read the journal's policy and the authors of this manuscript have the following competing interests: BB is an equity shareholder in Infinimmune and a member of their Scientific Advisory Board.

sequences allow the model to learn the interactions between antibody chains, which heavily influence antibody specificity. We introduce a training approach inspired by curriculum learning that gradually transitions from unpaired to paired sequences, allowing the model to learn the distinct data types more effectively. We show that our curriculum approach, along with other strategies that mix data types throughout training, leads to improved downstream performance. These findings contribute to the improvement of antibody language models by ensuring that we utilize the limited training data most effectively.

Introduction

Antibodies are a diverse and essential component of the adaptive immune system, with total available repertoire diversity estimated as high as 10^{18} unique antibodies [1]. This exceptional diversity results initially from the somatic recombination of germline gene segments [2] and is further refined upon antigen exposure via clonal expansion, somatic hypermutation, and antigen-driven selection of productive mutations. Within each recombined antibody gene, diversity is greatest in the complementarity-determining regions (**CDRs**). There are six CDR loops in each antibody, three encoded by the heavy chain and three by the light chain, which comprise the antigen-recognition site and thus determine antibody specificity.

Given the enormous diversity of the antibody repertoire, language models (**LMs**) are of increasing interest for their potential to speed up the discovery and engineering of novel antibodies. Antibody language models (**AbLMs**) are generally preferred over protein language models (**pLMs**) for antibody-related tasks, particularly due to their ability to learn immunological mechanisms such as affinity maturation [3] and antigen specificity [4,5]. Despite this, AbLMs tend to learn germline-encoded features easily but struggle with mutated residues and in the primarily non-templated CDR3 [4,6]. Given this fact, a major focus of current studies is to improve model performance in the CDR regions. We have recently shown that pre-training an AbLM with natively paired antibody sequences (containing both the heavy and light chains) rather than unpaired sequences (containing only a single chain) improves the model's ability to learn immunologically significant features that span both heavy and light chains, including heightened cross-chain attention on the functionally critical CDRs [4].

Despite the proven training benefits of paired sequences, available paired antibody sequence datasets are much smaller than unpaired datasets, and it is well documented that the cross-entropy (**CE**) loss of language models scales with dataset size [7,8]. Given this, recent models [6,9,10] have incorporated a mix of unpaired and natively paired sequences, with the intuition that we can supplement the limited paired data with the larger quantities of unpaired data available. In theory, this should also improve a model's ability to learn non-germline residues, since it sees a more diverse collection of somatically mutated antibodies in the larger unpaired datasets. Models trained with a mixture of unpaired and paired data typically pre-train using unpaired sequences and fine-tune with paired sequences [6]. IgBert and IgT5

from Kenlay et al 2024 [9] introduced a variation of this approach, using a 2:1 ratio of unpaired to paired data during the fine-tuning phase to help mitigate catastrophic forgetting [11].

However, due to the recency of this pre-training shift, we lack a systematic evaluation of pre-training strategies to determine the optimal parameters for a model trained on a mixture of these datatypes. We theorized that a smoother transition between unpaired and paired sequences may help mitigate the risk of catastrophic forgetting further. Specifically, we introduce a method of training AbLMs inspired by curriculum learning [12], which has been successful in the natural language processing domain. Curriculum learning involves ordering the training data, traditionally from 'easiest' to 'hardest' [13]. There are many different approaches to data ordering, one of which is length-based ordering, where sequence length increases as training progresses [14,15]. To apply this to AbLMs, we ordered the data to start with the shorter and more abundant unpaired sequences and transition to the longer and less abundant paired sequences gradually during training. We hypothesized that this may enable the model to balance its knowledge of both sequence types without catastrophic forgetting while still emphasizing the paired sequences. We will compare this curriculum approach with other training methods, including single data-type models, the traditional fine-tuning approach, and a constant approach (where data is mixed throughout training). In addition, we will train a large 650M-parameter curriculum model, CurrAb, and compare its performance on downstream residue prediction and classification tasks to existing models.

Results

Implementing curriculum learning for antibodies. Given the unique challenges associated with pre-training on both unpaired and paired sequence data, we introduce a modified version of curriculum learning for AbLMs. This method modifies the data sampling to select a variable percentage of unpaired sequences based on the training step. The highest percentage of unpaired sequences is selected at the beginning of training and this percentage decreases as training progresses, to emphasize paired sequences in the final stages of training. To implement this, a sigmoid decay function is used as the unpaired probability curve that determines the ratio of unpaired:paired data in each training batch (Fig 1).

To optimize the curriculum strategy, we tested a variety of hyperparameters general to mixed-data models as well as curriculum-specific parameters. To efficiently evaluate a large number of pre-training parameters, we initially used a pilot-scale 55M parameter variant of the ESM-2 [16] architecture, with a slightly modified vocab of 33 characters that includes a unique <sep> token. These models were trained for 100k steps on a subset (20%) of the full paired and unpaired

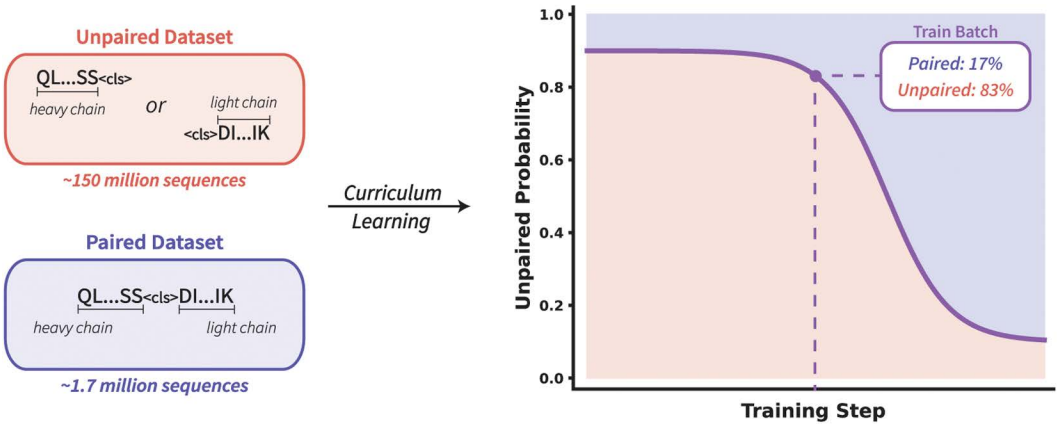


Fig 1. Curriculum learning implementation for AbLMs. This approach is designed to help handle the massive data imbalance between unpaired and paired datasets. To implement this, an unpaired probability curve determines the percentage of data sampled from the unpaired dataset for a given batch at a given step. This percentage decreases as training progresses, to allow for an emphasis on paired sequences near the end of training.

https://doi.org/10.1371/journal.pcbi.1013473.g001

datasets. Models are optimized based on their cross-entropy (**CE**) loss because it is sensitive to small hyperparameter changes.

Optimizing model architecture and data processing for mixed-data models. Prior to implementing the curriculum learning approach, we assessed the impact of a few key architecture and data formatting parameters for mixed-data models in general. Most significant are the changes observed based on the positional embedding (**PE**) type. Rotary position embeddings (**RoPE**) [17] consistently allow LMs to converge more quickly and reach a lower final loss than absolute PEs. By rotating the keys and queries based on their relative distance, RoPE can better accommodate sequences of varying lengths. Given that paired antibody sequences are approximately twice the length of unpaired ones, we hypothesized that RoPE would be especially impactful for AbLMs trained with a mixture of paired and unpaired sequences.

To test this, we trained four models: two trained using only paired sequences (paired-only models) and two trained using only unpaired sequences (unpaired-only models), with either RoPE or absolute PE. Each of these models was then evaluated using both paired and unpaired test datasets (Fig 2). As expected, RoPE models perform slightly better on CE loss than absolute PE models when tested with the same data type they were trained on (paired-only models tested with paired sequences and unpaired-only models tested with unpaired sequences). Notably, when evaluated using heterologous data (paired test sequences for unpaired-only models or unpaired test sequences for paired-only models), RoPE models outperform absolute PE models by a large margin (Fig 2A, 2C). In the paired-only models, the absolute PE model performs similarly to the RoPE model on unpaired heavy chains, but significantly worse across all regions in unpaired light chains (Fig 2B). Similarly, in the unpaired-only models, the CE loss of the absolute PE model on paired sequences progressively increases throughout the heavy chain and is significantly higher in the light chain, compared to the RoPE model (Fig 2D). To explore the cause of the deteriorated performance in the light chain, we trained two additional paired models with the chain order reversed (light chain preceding the heavy chain) (S1 Fig). In these tests, we observe that the absolute PE model performs poorly in the heavy chain (rather than the light chain), confirming that the performance decrease is caused by the position of the chains in the sequence inputs.

We next tested different separator token(s) and varying ratios of unpaired:paired data in mixed-data models. First, we note that a single separator token produced slight but consistent improvement on CE loss, regardless of whether the separator token was uniquely used for chain separation (<sep>) or a reuse of the start-of-string token (<cls>) (S1 Table). Interestingly, adding a separator to unpaired sequences (immediately following heavy chains or immediately preceding light chains) resulted in better performance than omitting the separator, suggesting that a separator is useful even in an unpaired-only model. Second, when testing ratios of unpaired:paired data, we observe that increasing training time for one data type consistently results in a lower loss of that data type, but it always comes at the cost of increased loss on the other data type. Once the total training dataset exceeds 50% paired sequences, the paired loss fails to improve any further (S2 Table). Based on this, we train all subsequent mixed-data models with a fixed percentage, 62.5%, of unpaired data. This hyperparameter ensures all mixed-data models are trained on the same amount (just over one epoch) of unpaired data.

Optimizing curriculum learning for antibodies. To optimize our curriculum learning method, we tested a variety of modifications to the unpaired probability curve and learning rate (**LR**) schedule. We first tested different ranges of probabilities for these curves, with the widest ranging from 1.0 to 0.0 (max1) and the narrowest ranging from 0.7 to 0.3 (max0.7) (Fig 3A). Performance on the unpaired test set reveals that the CE loss decreases as the probability range narrows (Fig 3B). Performance on the paired test set is much less variable, with the max0.8 range resulting in the lowest loss by a small margin. These results suggest that including a mix of data types (with a minimum of ~20% of the minority data type) throughout training is useful to the models' final performance. Second, we tested the impact of the slope of the unpaired probability curve decay, which is determined by the k-value. Models with a larger k-value decay at a steeper rate than models with a smaller k-value, resulting in a more rapid transition from unpaired to paired training data (Fig 3C). Changing the slope has almost no effect on the paired performance, with $k = 50$ performing only slightly better than the others (Fig 3D). We observe a slightly larger impact on the unpaired loss, with $k = 15$ resulting in the lowest CE loss.

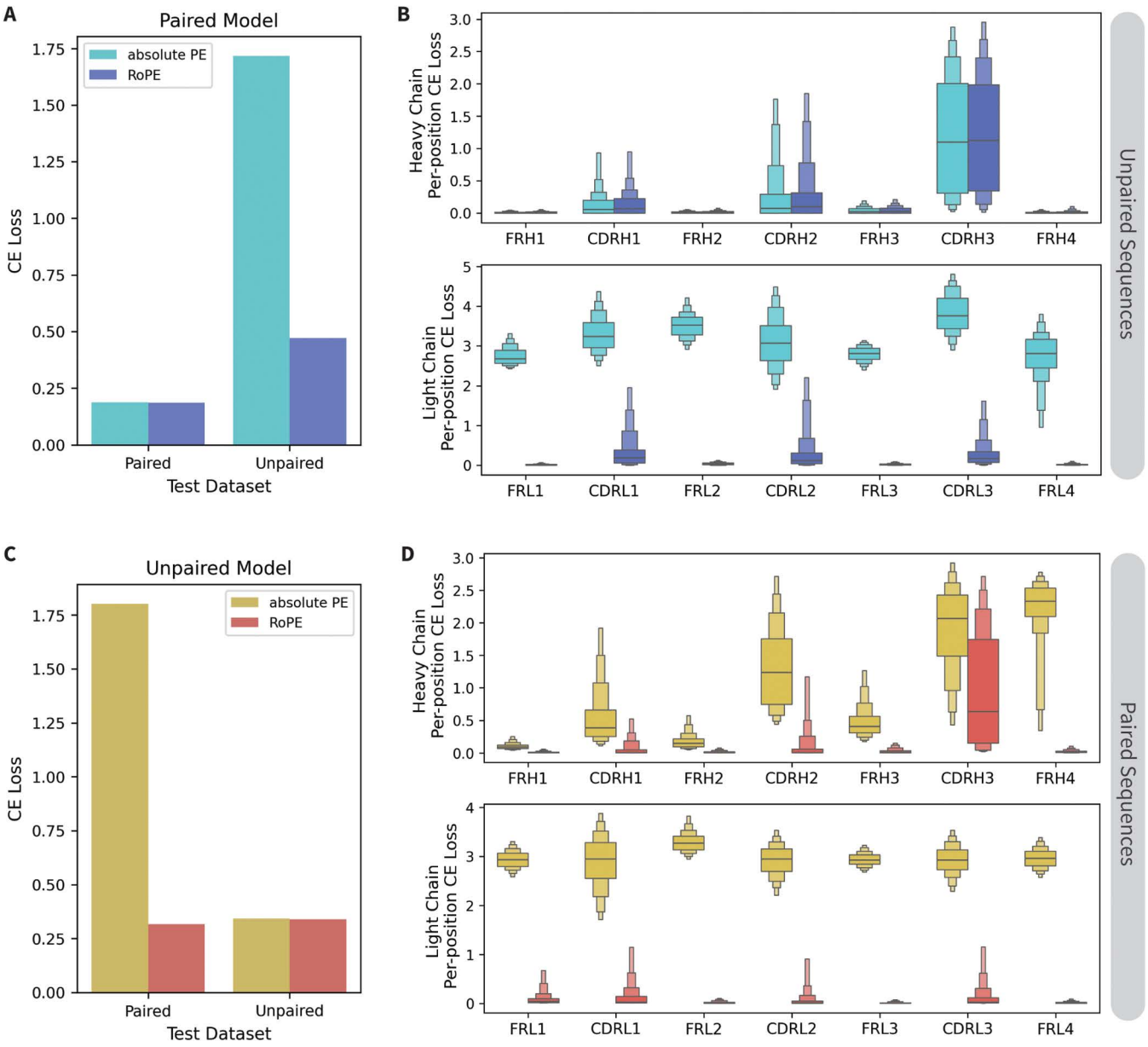
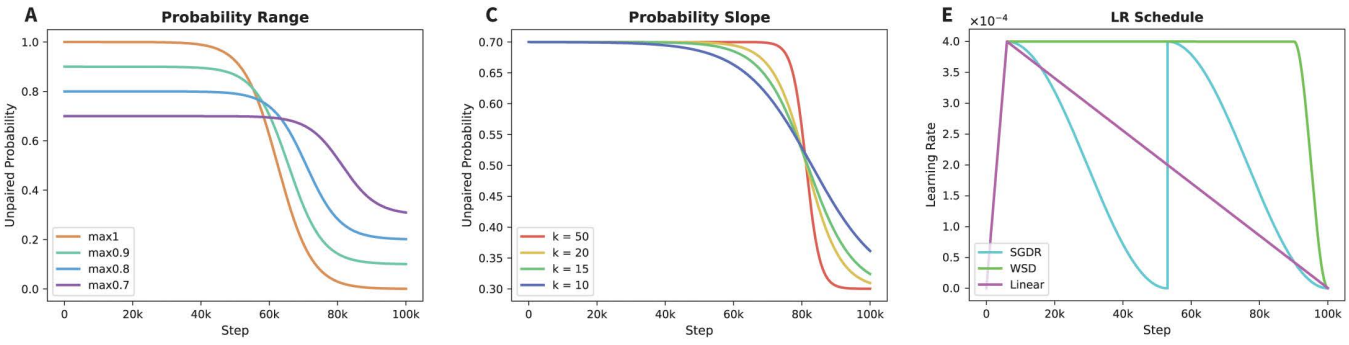


Fig 2. Comparing RoPE and absolute PE with paired and unpaired models. Two (A-B) paired-only models and (C-D) two unpaired-only models were trained with either RoPE or absolute PE. (A, C) CE loss on paired and unpaired test datasets of ~10k sequences each. (B) Per-position CE loss of the paired-only models on 1k sequences from the unpaired test dataset. (D) Per-position CE loss of the unpaired-only models on 1k sequences from the paired test dataset.

https://doi.org/10.1371/journal.pcbi.1013473.g002

Finally, we tested different learning rate schedules to determine if a higher learning rate later in training is useful for learning paired sequences. We evaluated a linear-decay schedule, a warmup-stable-decay (**WSD**) [18] schedule, and a cosine annealing schedule (commonly known as stochastic gradient descent with warm restarts (**SGDR**)) [19] (Fig 3E). We find that the linear LR results in the lowest unpaired test loss, while WSD results in the lowest test loss for paired sequences (Fig 3F). The improved performance of the WSD LR suggests that a higher learning rate during the transition



B

Max Prob	Paired CE Loss	Unpaired CE Loss
max1	0.1834	0.3569
max0.9	0.1834	0.3520
max0.8	**0.1832**	0.3506
max0.7	0.1833	**0.3502**

D

K value	Paired CE Loss	Unpaired CE Loss
50	**0.1832**	0.3505
20	0.1833	0.3502
15	0.1833	**0.3501**
10	0.1833	0.3504

F

LR	Paired CE Loss	Unpaired CE Loss
SGDR	0.1834	0.3558
WSD	**0.1831**	0.3539
Linear	0.1833	**0.3502**

Fig 3. Optimizing hyperparameters for curriculum models. (A-B) Testing different ranges of the unpaired probability curve. (C-D) Testing different slopes of the unpaired probability curve by adjusting the k-value. (E-F) Testing three different LR schedules: linear, WSD, and SGDR. All models were evaluated based on their CE loss on the paired and unpaired test datasets.

https://doi.org/10.1371/journal.pcbi.1013473.g003

can assist the model in learning paired sequences. However, this small gain in paired performance comes at a much larger cost to unpaired performance.

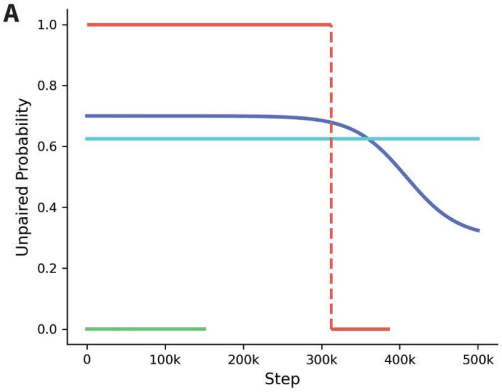
Based on the limited improvements observed in the paired CE loss, we optimized the curriculum implementation based on the unpaired CE loss. Therefore, our optimized parameters are a probability range of 0.7 to 0.3, a k-value of 15, and a linear LR schedule.

Assessing methods for training mixed models. To directly compare strategies for training with mixed datasets, we trained a series of four models: one model pre-trained on unpaired data and finetuned on paired data (*finetuned*), one model trained with a constant ratio of paired and unpaired data (*constant*), one model trained using our curriculum strategy that dynamically adjusts the ratio of paired and unpaired data throughout training (*curriculum*), and a control model trained using only paired data (*paired-only*). Initial tests performed on small 55M-parameter models, with an unpaired-only model as well, are presented in S2 Fig. These small-scale results revealed that the unpaired-only model was not competitive with other training strategies, so a large-scale version was not trained.

A distinct advantage of mixed models is the massively increased training data scale, which could allow longer training of larger models without overfitting. Therefore, we trained these models with 650M-parameters and an ESM-2 architecture for 500k steps with a reduced learning rate (1e-4) to stabilize training (Fig 4A). The mixed models were trained with a fixed unpaired percentage to ensure that they were trained with the same total amount (in epochs) of unpaired and paired data. However, due to overfitting, the 150k checkpoint was used for downstream testing of the paired-only model and the 75k checkpoint during paired finetuning was used for the finetuned model.

We evaluated each model with a masked language modeling objective using the held-out test data (Fig 4B) and observed that the curriculum and constant models performed best across CE loss and accuracy on the paired test set. We additionally observed that the curriculum model performed best across both CE loss and accuracy on the unpaired test set, followed by the constant model. The boost in unpaired performance in the curriculum model compared to the other mixed models suggests that the curriculum training schedule enables the model to retain its knowledge of unpaired sequences the best. To further assess the paired accuracy of the mixed models, we also performed per-position inference to calculate the CDRH3 accuracy on mutated, paired sequences (Fig 4C). We observed small differences in CDRH3

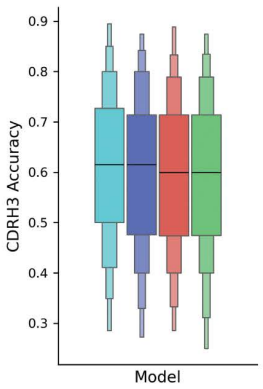
A



B

	Paired		Unpaired	
Model	**CE Loss**	**Accuracy**	**CE Loss**	**Accuracy**
constant	0.1816	**0.9511**	0.2781	0.9340
curriculum	**0.1813**	**0.9511**	**0.2777**	**0.9342**
finetuned	0.1827	0.9509	0.4422	0.9040
paired	0.1835	0.9507	0.4776	0.8997

C



D

Classification	Model	Accuracy	F1	MCC	AUC	AUPR
Native vs Random Pairs	constant	0.6426 (± 0.0021)	0.5511 (± 0.0029)	0.3124 (± 0.0048)	0.6978 (± 0.0025)	**0.7300** (± 0.0028)
	curriculum	**0.6428** (± 0.0008)	0.5462 (± 0.0029)	**0.3156** (± 0.0011)	**0.6985** (± 0.0017)	0.7289 (± 0.0016)
	finetuned	0.6328 (± 0.0018)	0.5335 (± 0.0036)	0.2934 (± 0.0039)	0.6808 (± 0.0029)	0.7085 (± 0.0036)
	paired	0.6373 (± 0.0017)	**0.5618** (± 0.0024)	0.2926 (± 0.0036)	0.6920 (± 0.0011)	0.7189 (± 0.0014)

E

Classification	Model	Accuracy	F1	MCC
Paired HD vs Flu vs CoV	constant	**0.6830** (± 0.0021)	**0.6836** (± 0.0023)	**0.5252** (± 0.0032)
	curriculum	0.6796 (± 0.0051)	0.6801 (± 0.0051)	0.5200 (± 0.0076)
	finetuned	0.6710 (± 0.0040)	0.6718 (± 0.0040)	0.5072 (± 0.0057)
	paired	0.6567 (± 0.0015)	0.6571 (± 0.0014)	0.4853 (± 0.0022)

F

Classification	Model	Accuracy	F1	MCC
Unpaired HD vs Flu vs CoV	constant	0.6369 (± 0.0046)	0.6377 (± 0.0047)	0.4560 (± 0.0072)
	curriculum	**0.6401** (± 0.0052)	**0.6408** (± 0.0053)	**0.4612** (± 0.0077)
	finetuned	0.6373 (± 0.0073)	0.6383 (± 0.0072)	0.4572 (± 0.0107)
	paired	0.6260 (± 0.0057)	0.6266 (± 0.0058)	0.4397 (± 0.0088)

G

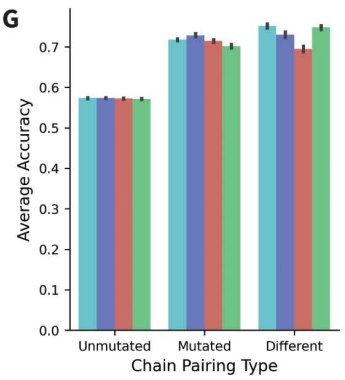


Fig 4. Comparing the performance of mixed model training methods. (A) Unpaired probability curves for the four 650M-parameter models. (B) CE loss and accuracy on paired and unpaired test datasets of ~50k sequences. (C) Mixed models accuracy at predicting CDRH3 of ~500 mutated paired sequences from the test set. Results for the three classification tasks, which are one (D) *Native vs Random* pair classification and two *Healthy Donor vs Flu vs CoV* specificity classifications using (E) paired and (F) unpaired sequences. (G) The results of the pair classification were additionally split by mutated pairs, unmutated pairs, and mismatched pairs. Metrics on classification tasks are mean and standard error, with the highest values bolded and the second highest values underlined.

https://doi.org/10.1371/journal.pcbi.1013473.g004

accuracy, with the curriculum and constant model showing a slightly higher median accuracy compared to the finetuned and paired models. However, this difference is small, indicating that the method of training a mixed model does not have a significant impact on the model's understanding of the CDRH3 region.

To further assess model performance, we performed a pair classification task, introduced in Ng and Briney 2025 [20]. The goal of this binary classification task is to identify whether a given paired sequence is a native or random pairing. The dataset was generated from the paired test dataset and is composed of naive (unmutated) and memory (mutated) sequences. The curriculum model performed best across all metrics except F1, followed by the constant model (Fig 4D). Next, we separated the test data for pair classification into three subsets: pairs in which both chains are mutated

(*mutated*), pairs in which both chains are unmutated (*unmutated*), or pairs in which one chain is mutated while the other is not (*different*) (Fig 4G). The models generally perform the best on the 'different' subset, with the constant and paired-only models showing the highest classification accuracy, indicating that pair classification is driven at least in part by identifying similar levels of mutation across both chains. Models perform similarly well on the 'mutated' subset, with the curriculum model showing the highest classification accuracy, suggesting that these models have learned the types of mutations in each chain that pair well together. Performance on the 'unmutated' subset is lowest, as expected, given that chain pairing of naive chains is random by design, to increase the potential diversity of the antibody repertoire.

We next performed three-way specificity classification tasks, in which the model was finetuned to classify between coronavirus (**CoV**) specific antibodies, influenza (**Flu**) specific antibodies, and nonspecific healthy donor (**HD**) antibodies. Models were fine-tuned with either paired sequences (Fig 4E) or unpaired heavy chain sequences (Fig 4F). The constant model outperformed all models on the paired task, while the curriculum model outperformed all models on the unpaired task. Results for a two-way specificity classification task (CoV vs HD) can be found in S3 Table, where we observe similar results on the paired task, but observe that the finetuned model performs the best on the unpaired task in this case.

These results indicate that either a constant or a curriculum model will outperform a finetuned model in most instances for training large-scale models, likely because these strategies help prevent catastrophic forgetting and allow the model to be trained for longer without overfitting. Additionally, while the paired-only model performed poorly in these tests, this is likely because the number of parameters is too large for the small paired dataset. This conclusion is supported by the results of the 55M-parameter models in S2 Fig, where the paired-only model outperforms the mixed models most of the time. This suggests that a paired-only strategy is preferred for training small models, but as model size increases, the mixed-data strategies are superior.

Large-scale curriculum model outperforms existing mixed models. To further assess the 650M-parameter curriculum model, which we will call CurrAb, we additionally compare its performance to several existing mixed AbLMs: IgBERT [7], IgT5 [7], AbLang2 [3], and AntiBERTa2 [8], and the 650M-parameter ESM-2 [16] pLM (Fig 5).

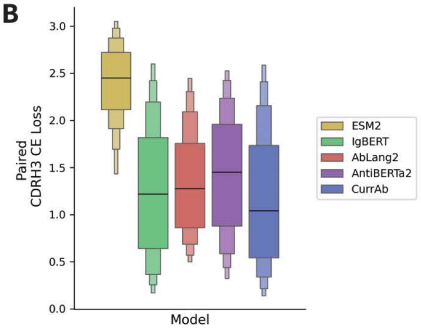
First, we assess the model's performance on a masked language modeling objective with a dataset of ~25k memory B-cell sequences from 3 healthy donors [16]. Since this dataset was published recently and is not in the OAS, we chose to use it instead of our test dataset to reduce the likelihood of data leakage with the training datasets of the other models. IgT5 was excluded from this task since it is an encoder-decoder architecture. CurrAb outperforms other models on inference of paired sequences and unpaired heavy chain sequences, followed closely by IgBERT, then AbLang2 (Fig 5A). IgBERT outperforms other models on unpaired light chains, followed by CurrAb. Strangely, AntiBERTa2 performs well on unpaired heavy chains but very poorly on paired and unpaired light chain sequences. Per-position inference of the CDRH3 on a random sample of 1k sequences reveals a similar trend, with CurrAb showing the lowest median CE loss (Fig 5B). The full per-position inference results for paired and unpaired sequences are presented in S3 Fig.

Second, we perform the pair classification task described previously, and observe that CurrAb outperforms the other models across all metrics except F1, followed most closely by AbLang2 (Fig 5C). ESM-2 performs the worst, consistent with previous studies showing that AbLMs outperform pLMs on antibody-specific tasks [21,22]. The unexpectedly low performance of IgBERT and AntiBERTa2, which is only slightly better than that of ESM-2, may be due to the classification head of BERT models, which is only a single projection layer that may not be sufficient for this complex task. We additionally performed the three-way specificity classification with CoV-specific, Flu-specific, and HD sequences. CurrAb again outperformed other models on both the paired (Fig 5D) and unpaired (Fig 5E) specificity classification tasks. The second-best model performances are AbLang2 in paired specificity classification and AntiBERTa2 in unpaired specificity classification. A closer examination of the paired specificity classification results showed that CurrAb was > 10% more accurate than the next best-performing model, AbLang2 (Fig 5F). Similar patterns are observed in a two-way (CoV vs HD) specificity classification task, as shown in S3 Fig.

A

	Paired		Heavy Chain		Light Chain	
Model	**CE Loss**	**Accuracy**	**CE Loss**	**Accuracy**	**CE Loss**	**Accuracy**
ESM2	1.2620	0.6363	1.2200	0.6541	1.2952	0.6256
IgBERT	0.4735	0.8756	0.6047	0.8400	**0.3652**	**0.9119**
IgT5	-	-	-	-	-	-
AbLang2	0.6211	0.8646	0.7099	0.8313	0.4710	0.9034
AntiBERTa2	1.4729	0.6039	0.8985	0.8045	2.6415	0.3614
CurrAb	**0.4713**	**0.8762**	**0.5982**	**0.8414**	0.3686	0.9110

B



C

Classification	Model	Accuracy	F1	MCC	AUC	AUPR
Native vs Random Pairs	ESM2	0.5108 (± 0.0019)	0.5280 (± 0.0042)	0.5146 (± 0.0030)	0.5093 (± 0.0016)	0.0216 (± 0.0038)
	IgBERT	0.5213 (± 0.0032)	0.5364 (± 0.0050)	0.0427 (± 0.0064)	0.5369 (± 0.0031)	0.5267 (± 0.0030)
	IgT5	0.6089 (± 0.0006)	0.5417 (± 0.0021)	0.2278 (± 0.0012)	0.6238 (± 0.0024)	0.6585 (± 0.0019)
	AbLang2	0.6163 (± 0.0023)	**0.5493** (± 0.0029)	0.2436 (± 0.0048)	0.6579 (± 0.0011)	0.7011 (± 0.0019)
	AntiBERTa2	0.5308 (± 0.0016)	0.5059 (± 0.0067)	0.0620 (± 0.0030)	0.5430 (± 0.0027)	0.5342 (± 0.0024)
	CurrAb	**0.6428** (± 0.0008)	0.5462 (± 0.0029)	**0.3156** (± 0.0011)	**0.6985** (± 0.0017)	**0.7289** (± 0.0016)

D

Classification	Model	Accuracy	F1	MCC
Paired HD vs Flu vs CoV	ESM2	0.5232 (± 0.0045)	0.5200 (± 0.0038)	0.2954 (± 0.0084)
	IgBERT	0.5266 (± 0.0099)	0.5226 (± 0.0103)	0.2926 (± 0.0148)
	IgT5	0.4789 (± 0.0102)	0.4534 (± 0.0130)	0.2390 (± 0.0149)
	AbLang2	0.5780 (± 0.0092)	0.5777 (± 0.0093)	0.3675 (± 0.0138)
	AntiBERTa2	0.5759 (± 0.0040)	0.5768 (± 0.0037)	0.3658 (± 0.0063)
	CurrAb	**0.6796** (± 0.0051)	**0.6801** (± 0.0051)	**0.5200** (± 0.0076)

E

Unpaired HD vs Flu vs CoV	ESM2	0.4982 (± 0.0043)	0.4932 (± 0.005)	0.2568 (± 0.0063)
	IgBERT	0.5023 (± 0.0073)	0.5014 (± 0.0071)	0.2539 (± 0.0110)
	IgT5	0.4991 (± 0.0106)	0.4870 (± 0.0128)	0.2599 (± 0.0134)
	AbLang2	0.5659 (± 0.0064)	0.5656 (± 0.0067)	0.3492 (± 0.0096)
	AntiBERTa2	0.6110 (± 0.0064)	0.6116 (± 0.0063)	0.4174 (± 0.0096)
	CurrAb	**0.6401** (± 0.0052)	**0.6408** (± 0.0053)	**0.4612** (± 0.0077)

F

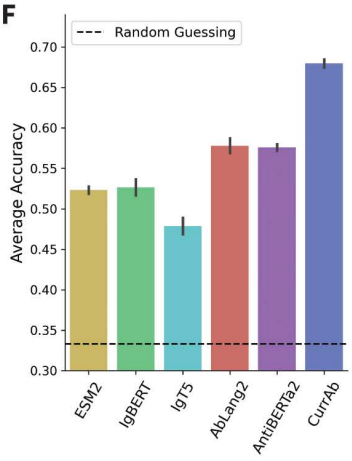


Fig 5. Comparing CurrAb to existing mixed data AbLMs and pLMs. (A) CE loss and accuracy of MLM inference on ~25k memory B-cell sequences. IgT5 is excluded because it is an encoder-decoder architecture. (B) Per-position inference in the CDRH3 on a 1k subset of the memory B-cell sequences. Results of three classification tasks, which are one (C) *Native vs Random* pair classification and two *Healthy Donor vs Flu vs CoV* specificity classification tasks using (D) paired and (E) unpaired sequences. (F) The bar plot shows the mean accuracy and standard error of the pair classification task. Metrics on classification tasks are mean and standard error, with the highest values bolded and the second highest values underlined.

https://doi.org/10.1371/journal.pcbi.1013473.g005

IgBERT and IgT5 were trained with an approach closest to the curriculum strategy – these models were pre-trained using unpaired sequences and finetuned with a mix of unpaired and paired sequences. IgBERT performs well on the inference tasks, suggesting that this training strategy is similarly beneficial to a curriculum strategy for residue prediction. However, despite this similar training approach and the large size of IgT5 (3B parameters), these models tend to perform poorly on classification tasks. This suggests that model architecture (particularly rotary embeddings and the size of the classifier head) and a larger emphasis on paired sequences are critical to high performance on these tasks.

Discussion

Recent studies on AbLMs have shown that the benefits of training on natively paired sequences can be sufficient to overcome a significant disadvantage in training data scale compared to unpaired sequences. The high cost and effort required to recover natively paired antibody sequences has sparked interest in methods that maximize the training value of these limited datasets, including supplementing paired sequences with unpaired sequences. Theoretically, this mixed training approach should allow the model to learn critical cross-chain features while leveraging the greater diversity in much larger unpaired sequence datasets. However, it is not clear how best to combine these two data types to maximize training value.

We first performed a systematic analysis of model architectures and data formatting methods to understand their effect on mixed model training. The most impactful change was the use of rotary embeddings, which delivered a surprisingly large improvement in mixed-data model performance. Compared to absolute PE, RoPE greatly reduced the loss of homogeneous models (paired-only or unpaired-only) when tested with the heterologous data type. This is strong evidence that mixed-data models should use rotary embeddings whenever possible. In addition, we noted that mixed-data models using a single chain separator token (whether that be <cls> or <sep>) performed best, with the <cls> token showing a slight performance benefit in homogeneous models as well.

Next, we introduced a curriculum learning approach to training AbLMs, which allows for a gradual transition from unpaired to paired sequences. Including both data types throughout the entire training course, with an increased emphasis on paired sequences toward the end of training, resulted in the best final model performance. We also showed that adjusting the parameters of the unpaired probability curve can differentially affect model performance on unpaired or paired sequences, meaning the curriculum approach can be tuned for specific performance objectives. Generally, optimizing for unpaired loss resulted in larger overall performance gains on downstream tasks, perhaps because unpaired sequences comprise the bulk of the training data.

Upon comparing our curriculum strategy to existing training strategies, we observe that the large-scale curriculum and constant models consistently outperform the finetuned model on downstream residue prediction and classification tasks. This is likely because these strategies allow models to be trained longer by slowing overfitting with mixed training batches and by preventing the catastrophic forgetting of unpaired sequences. However, at a small scale, we observe that the paired-only model consistently outperforms all the mixed model types. This suggests that the ideal training strategy depends on the model size, but that you generally observe the benefits of training on unpaired and paired sequences as the model size increases. Further research is necessary to determine the tipping point in model size at which mixed training data becomes useful.

We expected that one of the primary performance-enhancing benefits of supplementing paired training data with unpaired sequences would be the ability to train larger models for longer without overfitting. CurrAb, a 650M parameter curriculum model, validated this assumption by outperforming existing models on various tasks, including residue prediction, native pairing classification, and antigen specificity classification. The magnitude of improvement over IgBERT and IgT5 in the classification tasks was surprising, particularly given that these models were finetuned with a mixture of paired and unpaired sequences. While this strategy is similar to our curriculum training approach in that the final training stages do not focus exclusively on a single data type, we note two differences that may underlie this performance disparity. First, it is possible that exposure to paired sequences only during finetuning is insufficient for the model to learn cross-chain features fully; instead, paired sequences may need to be introduced at the earliest stages of training to capture their full training value. Second, architectural optimizations such as rotary position embeddings, which are lacking in IgBERT and IgT5, may yield substantial performance improvements on downstream classification tasks, particularly those that involve paired antibody sequences.

More broadly, these findings emphasize the importance of generating larger datasets of paired antibody sequences, as paired-only models approach the performance of mixed models trained with substantially more data. In other words, the

marginal training value of new data appears to strongly favor paired antibody sequences, despite the high cost of generating natively paired antibody sequences. It may also be useful to apply additional training methods that focus learning on the CDR regions, such as increased masking rates [20,23] and focal loss [6,24], to use both the paired and unpaired data more efficiently. Additionally, generative AbLMs trained with unpaired and paired sequences are being actively developed [25,26] due to the architecture's success in the NLP domain. Future research is needed to determine the success of a curriculum training strategy in generative models, but the higher diversity observed by mixing data types should be beneficial for improving sequence generation. Beyond AbLMs, the ability of curriculum-based training approaches to prevent catastrophic forgetting is likely to be beneficial for a variety of biological models for which cross-domain expertise is important. For example, a curriculum model trained with a combination of proteins and antibodies could perform as well as a specialized antibody sequence or structure model while retaining an expert-level understanding of general proteins. Such a model could be very well-suited for tasks like antibody-antigen docking.

In summary, we report four important findings. First, RoPE significantly impacts the ability of AbLMs to generalize across unpaired and paired data, highlighting positional embeddings as a key but underappreciated component of mixed-data biological language models. Second, curriculum AbLMs typically perform best when optimized for unpaired performance, but the most important factor for final model performance is ensuring a sufficient emphasis on paired sequences throughout the entire training course. Third, a mixture of unpaired and paired sequences throughout training, using either curriculum or constant sampling methods, is useful as the model size scales. Finally, curriculum learning effectively mitigates catastrophic forgetting, making it a useful strategy for training biological models that exhibit uniformly high performance across multiple domains.

Methods

Training Data. Sequence data was downloaded from the OAS [27] on September 12th, 2024. In addition to sequences from the OAS, the paired dataset was supplemented with an internally generated dataset of ~400k paired sequences. Filtering and clustering were performed as described in AntiRef [28], with additional filtering for sequences containing 'nan' characters. The dataset clustered at 90% was chosen for both datasets, resulting in 151,764,423 unpaired sequences and 1,717,423 paired sequences in the full datasets. The final 650M-parameter model, CurrAb, was trained on the full dataset and used 96% of the data for training, with the remaining 4% left out for the evaluation and test sets. Test sets used for inference in Fig 4 were the paired test set and a sample of ~50k sequences from the unpaired test set (sampled to match the number of sequences in the paired test set).

For compute efficiency, initial tests in Figs 2, 3, S1 and S2 and S1 and S2 Tables were trained with a downsampled dataset (20% of the full dataset), resulting in 30,352,885 unpaired sequences and 343,485 paired sequences. These datasets were similarly split to use 96% of the sequences for training and the remaining 4% for the evaluation and test sets. Test sets used for inference in Figs 2 and 3 were the paired test set and a sample of ~10k sequences from the unpaired test set (to match the number of sequences in the paired test set).

For the MLM tasks in Fig 5, the dataset was obtained from Ng and Briney 2025 [20]. This dataset contains memory B-cell sequences from three donors. These sequences were clustered at 90%, resulting in 26,312 paired sequences.

For the specificity classification tasks, two datasets were generated. CoV-specific sequences were downloaded from CoV-AbDab [29] on November 11th, 2024 and an internal donor L1236. Flu-specific sequences were obtained from Wang et al. [5] and filtered for paired sequences only. Healthy donor sequences were obtained from memory B-cell sequences from the Jaffe et al. [30] and Phad et al. [31] datasets and filtered to exclude any sequences in the train and evaluation datasets to prevent data leakage for both model sizes. For the HD-Flu-CoV classifications, the datasets were clustered by class at 99%, then combined with an equal number of sequences in each class, resulting in a final dataset with 4,398 sequences. For the HD-CoV classifications, the datasets were clustered by class at 95%, then combined with an equal number of sequences in each class, resulting in a final dataset with 27,442 sequences. Both classification datasets were

split for 5-fold cross-validation with stratification. The same datasets were used for the unpaired versions of these tasks, but only the heavy chain sequence was provided.

For the pair classification task, the paired sequences from the full-data test set were filtered to exclude any sequences in the downscaled train and evaluation datasets, to prevent data leakage for both model sizes. This resulted in a dataset of 41,564 paired sequences containing both naive and memory B-cell sequences. The dataset was processed as described in Ng and Briney 2025 [20], including a split for 5-fold cross-validation with stratification.

Curriculum Implementation. To enable training of different types of mixed models (constant and curriculum), modifications were made to the HuggingFace [32] trainer using a trainer callback and by subclassing the PyTorch [33] dataset class. This facilitated the separation of unpaired and paired datasets (for logging and evaluation purposes) and allowed for tracking and updating the unpaired probability function throughout training. The unpaired probability function determined what percentage of sequences should be sampled from the unpaired dataset at each step during training. The unpaired probability $P(t)$ for the curriculum models is represented with the following equation:

$$P(t) = B - \frac{A}{1 + e^{-k \cdot (t - shift)}}$$

where t equals the current step divided by the total train steps. The values of A (height of the curve) and B (the vertical shift, aka upper bound of the curve) were modified to adjust the range of the probabilities, the k value was modified to adjust the slope of the sigmoid function, and the *shift* values were calculated to ensure that the total unpaired percentage (i.e., 62.5%) was achieved. Refer to S4 Table for the specific values used for each curriculum model.

Model Pre-Training. All models used a slightly modified ESM-2 architecture model. This is an encoder-only architecture that produces embeddings useful for downstream tasks such as specificity classification, pair classification, and structure prediction. Models were trained with an MLM objective. This means that 15% of the input sequence was selected for prediction and of these, 80% were replaced with a <mask> token, 10% were replaced with a random token from the vocabulary, and 10% were left unchanged.

All models used a slightly modified ESM-2 vocabulary of 33 tokens, with an added <sep> token. Based on the separator tests presented in S1 Table, the large-scale models were trained with a <cls> separator. Sequences were preprocessed to place the separator between the heavy and light chains of the paired sequences ("QL…SS<cls>DI…IK"), at the end of unpaired heavy chains ("QL…SS<cls>"), and at the beginning of unpaired light chains ("<cls>DI…IK"). Inputs were padded to 320 tokens to accommodate the longest paired sequence.

For initial tests, we used a modified 55M-parameter ESM-2 architecture model, with 5 layers, 20 attention heads per layer, a hidden size of 960, and an intermediate size of 3840. After initial embedding tests in Fig 2, RoPE was used for all subsequent models. Models were trained for 100k steps with a total batch size of 512. On 4 L40S graphics processing units (GPUs) using DeepSpeed ZeRO Stage 1 [34] via the Accelerate [35] library, this equates to ~11 hours per model. The peak learning rate was 4e-4, with a linear warmup for the first 6,000 steps followed by a linear decay. The only exception to this was the LR tests in Fig 3E–F, which tested WSD and SDGR LR schedules. The ideal parameters for WSD and SGDR were chosen based on Hägele et al 2024 [36].

The large-scale models, including CurrAb, used a 650M parameter ESM-2 architecture with RoPE and the <cls> token as the separator token in the paired and unpaired sequences. The models were trained for 500k steps with a total batch size of 512, equating to ~6 days on 8 A100 GPUs using DeepSpeed ZeRO Stage 1. The peak learning rate was reduced to 1e-4 to assist with model convergence, with a linear warmup for the first 30,000 steps followed by a linear decay. Since the finetuned model was trained in two stages, the unpaired-only stage was trained for 312.5k steps with 18,750 warm-up steps, and the paired-only stage was trained for 187.5k steps with 11,250 warm-up steps. Early checkpoints were taken for the paired-only model (150k steps) and the finetuned model (75k steps into the paired-only stage, aka 262.5k steps total).

Models were logged using Weights & Biases (wandb). Data from wandb was used to produce the unpaired probability, LR schedule, and evaluation loss curves.

Model Evaluation & Testing. CE loss and accuracy calculations on the eval and test sets were performed using the HuggingFace trainers evaluate function, with a custom compute_metrics function provided to ensure the metrics exclude the separator tokens from calculations. Accuracy was calculated using scikit-learn [37] and CE loss was calculated using PyTorch. CDRH3 accuracy was calculated by masking and predicting each position in the CDRH3 region individually, then calculating the accuracy of those predictions by averaging.

Plots were generated using seaborn [38] and matplotlib [39].

Classification Tasks. For all classification tasks, models were finetuned with a standard sequence classification head, with the base model weights frozen. For all the models we trained and the models hosted on Hugging Face (ESM2, IgBERT, IgT5, and AntiBERTa2), the base models were loaded with the default sequence classification head for that model architecture. For AbLang2, which is only available as a pip package, the sequence classification head was added manually and modeled after the ESM-2 classification head.

For specificity classification tasks, all models had a warmup ratio of 0.1 and a peak learning rate of 5e-5 followed by a linear decay. For binary classification (HD vs CoV), models were trained for 10 epochs and a total batch size of 32 per step. For 3-class classification (HD vs Flu vs CoV), models were trained for 5 epochs and a total batch size of 8 per step.

For pair classification tasks, models were finetuned following the training schedule presented in Ng and Briney 2025 [20]. Models were trained for 50 epochs, with a warmup ratio of 0.1, a peak learning rate of 1e-5, and a total batch size of 256 per step.

Metrics used to evaluate the classifiers were: accuracy, F1, area under the receiver operating characteristic curve (**AUC**), area under the precision-recall curve (**AUPR**), and Matthews correlation coefficient (**MCC**). Metrics on the test set were averaged across cross-validation runs and standard error was calculated. Scikit-learn was used to calculate these metrics.

Code and Data Availability. The code used for model training, evaluation, and figure generation is available on GitHub (github.com/brineylab/curriculum-paper). The training data and model weights for CurrAb are available on Zenodo (doi.org/10.5281/zenodo.14661302) and the 650M-parameter mixed models, including CurrAb, are also uploaded on Hugging Face (huggingface.co/collections/brineylab/curriculum-paper-685b08a4b6986df7c5a5e3c4).

Supporting information

S1 Fig. Comparing RoPE and absolute PE of paired-only models, with chain order reversed during training. (A) CE loss on paired and unpaired test datasets of ~10k sequences each. (B) Per-position CE loss of models on 1k sequences from the unpaired dataset.
(PDF)

S1 Table. CE Loss and accuracy for separator token tests. Mixed, paired-only, and unpaired-only models were trained with 5 different separators. Separators were placed between chains in paired sequences and unpaired sequences based on the chain (end of the heavy chains and the beginning of the light chains). Models were assessed on paired and unpaired test datasets, each containing ~ 10k sequences.
(PDF)

S2 Table. CE loss and accuracy for different unpaired percentages. Mixed models were trained with increasing percentages of unpaired data. Models were assessed on paired and unpaired test datasets, each containing ~ 10k sequences.
(PDF)

S2 Fig. Pilot 55M-parameter mixed model comparison. (A) Unpaired probability curves for the five pilot models. (B) CE loss on paired and unpaired test datasets of ~10k sequences. (C) Mixed models accuracy at predicting CDRH3 of 1k paired sequences from the test set. Results for classification tasks, which are one (D) *Native vs Random* pair classification, two *Healthy Donor vs Flu vs CoV* specificity classifications using (E) paired and (F) unpaired sequences, and two *Healthy Donor vs CoV* specificity classifications using (H) paired and (I) unpaired sequences. (G) Pair classification results were split into mutated, unmutated, and mismatched pairs. Metrics on classification tasks are mean and standard error, with the highest values bolded and the second highest values underlined.
(PDF)

S3 Table. Additional 650M-parameter mixed model comparisons. Results for *Healthy Donor vs CoV* specificity classification tasks with paired and unpaired sequences. Metrics on classification tasks are mean and standard error, with the highest values bolded and the second highest values underlined.
(PDF)

S3 Fig. Additional comparisons of existing models to CurrAb. Per-position CE loss on 1k memory B-cell (A) paired and (B) unpaired sequences. Results for *Healthy Donor vs CoV* specificity classification tasks with (C) paired and (D) unpaired sequences. Metrics on classification tasks are mean and standard error, with the highest values bolded and the second highest values underlined.
(PDF)

S4 Table. Unpaired probability equation values for curriculum models. Corresponding figure, model name, and equation values (A, B, shift, and k) are listed for each curriculum model.
(PDF)

Author contributions

Conceptualization: Sarah M Burbach, Bryan Briney.

Funding acquisition: Bryan Briney.

Methodology: Sarah M Burbach.

Software: Sarah M Burbach.

Supervision: Bryan Briney.

Validation: Sarah M Burbach.

Visualization: Sarah M Burbach.

Writing – original draft: Sarah M Burbach, Bryan Briney.

Writing – review & editing: Sarah M Burbach, Bryan Briney.

References

1. Briney B, Inderbitzin A, Joyce C, Burton DR. Commonality despite exceptional diversity in the baseline human antibody repertoire. Nature. 2019;566(7744):393–7. https://doi.org/10.1038/s41586-019-0879-y PMID: 30664748
2. Tonegawa S. Somatic generation of antibody diversity. Nature. 1983;302(5909):575–81. https://doi.org/10.1038/302575a0 PMID: 6300689
3. Ruffolo JA, Gray JJ, Sulam J. Deciphering antibody affinity maturation with language models and weakly supervised learning. 2021. https://arxiv.org/abs/2101.00001
4. Burbach SM, Briney B. Improving antibody language models with native pairing. Patterns (N Y). 2024;5(5):100967. https://doi.org/10.1016/j.patter.2024.100967 PMID: 38800360
5. Wang Y, Lv H, Teo QW, Lei R, Gopal AB, Ouyang WO, et al. An explainable language model for antibody specificity prediction using curated influenza hemagglutinin antibodies. Immunity. 2024;57(10):2453-2465.e7. https://doi.org/10.1016/j.immuni.2024.07.022 PMID: 39163866

6. Olsen TH, Moal IH, Deane CM. Addressing the antibody germline bias and its effect on language models for improved antibody design. bioRxiv. 2024. 2024.02.02.578678. https://doi.org/10.1101/2024.02.02.578678
7. Kaplan J, McCandlish S, Henighan T, Brown TB, Chess B, Child R, et al. Scaling laws for neural language models. arXiv. 2020. https://arxiv.org/abs/2001.08361
8. Hoffmann J, Borgeaud S, Mensch A, Buchatskaya E, Cai T, Rutherford E, et al. Training Compute-Optimal Large Language Models. arXiv. 2022. https://arxiv.org/abs/2203.15556
9. Kenlay H, Dreyer FA, Kovaltsuk A, Miketa D, Pires D, Deane CM. Large scale paired antibody language models. arXiv. 2024. https://arxiv.org/abs/2401.00001
10. Barton J, Galson JD, Leem J. Enhancing antibody language models with structural information. bioRxiv. 2024. https://doi.org/10.1101/2023.12.12.569610
11. Kirkpatrick J, Pascanu R, Rabinowitz N, Veness J, Desjardins G, Rusu AA, et al. Overcoming catastrophic forgetting in neural networks. Proc Natl Acad Sci U S A. 2017;114(13):3521–6. https://doi.org/10.1073/pnas.1611835114 PMID: 28292907
12. Bengio Y, Louradour J, Collobert R, Weston J. Curriculum learning. In Proceedings of the 26th Annual International Conference on Machine Learning (ACM). 2009. p. 6. https://doi.org/10.1145/1553374.1553380
13. Soviany P, Ionescu RT, Rota P, Sebe N. Curriculum Learning: A Survey. arXiv. 2021. https://arxiv.org/abs/2101.00001
14. Spitkovsky VI, Alshawi H, Jurafsky D. From baby steps to leapfrog: How "less is more" in unsupervised dependency parsing. North Am Chapter Assoc Comput Linguistics. 2010:751–9.
15. Nagatsuka K, Broni-Bediako C, Atsumi M. Length-Based Curriculum Learning for Efficient Pre-training of Language Models. New Gener Comput. 2022;41(1):109–34. https://doi.org/10.1007/s00354-022-00198-8
16. Lin Z, Akin H, Rao R, Hie B, Zhu Z, Lu W, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. Science. 2023;379(6637):1123–30. https://doi.org/10.1126/science.ade2574 PMID: 36927031
17. Su J, Lu Y, Pan S, Murtadha A, Wen B, Liu Y. RoFormer: Enhanced transformer with Rotary Position Embedding. arXiv. 2021. https://arxiv.org/abs/2104.09864
18. Hu S, Tu Y, Han X, He C, Cui G, Long X, et al. MiniCPM: Unveiling the potential of Small Language Models with scalable training strategies. arXiv. 2024.
19. Loshchilov I, Hutter F. SGDR: Stochastic gradient descent with warm restarts. arXiv. 2016.
20. Ng K, Briney B. Focused learning by antibody language models using preferential masking of non-templated regions. Patterns (N Y). 2025;6(6):101239. https://doi.org/10.1016/j.patter.2025.101239 PMID: 40575131
21. Olsen TH, Moal IH, Deane CM. AbLang: an antibody language model for completing antibody sequences. Bioinform Adv. 2022;2(1):vbac046. https://doi.org/10.1093/bioadv/vbac046 PMID: 36699403
22. Leem J, Mitchell LS, Farmery JHR, Barton J, Galson JD. Deciphering the language of antibodies using self-supervised learning. Patterns (N Y). 2022;3(7):100513. https://doi.org/10.1016/j.patter.2022.100513 PMID: 35845836
23. Gao K, Wu L, Zhu J, Peng T, Xia Y, He L, et al. Pre-training Antibody Language Models for Antigen-Specific Computational Antibody Design. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining KDD'23. (Association for Computing Machinery). 2023. p. 506–17. https://doi.org/10.1145/3580305.3599468
24. Lin TY, Goyal P, Girshick R, He K, Dollár P. Focal Loss for Dense Object Detection. arXiv. 2017.
25. Turnbull OM, Oglic D, Croasdale-Wood R, Deane CM. P-IgGen: A paired antibody generative language model. bioRxiv. 2024. 2024.08.06.606780. https://doi.org/10.1101/2024.08.06.606780
26. Barton J, Gaspariunas A, Yadin DA, Dias J, Nice FL, Minns DH, et al. A generative foundation model for antibody sequence understanding. bioRxiv. 2024. https://doi.org/10.1101/2024.05.22.594943
27. Kovaltsuk A, Leem J, Kelm S, Snowden J, Deane CM, Krawczyk K. Observed Antibody Space: A Resource for Data Mining Next-Generation Sequencing of Antibody Repertoires. J Immunol. 2018;201(8):2502–9. https://doi.org/10.4049/jimmunol.1800708 PMID: 30217829
28. Briney B. AntiRef: reference clusters of human antibody sequences. bioRxiv. 2022. 2022.12.30.522338. https://doi.org/10.1101/2022.12.30.522338
29. Raybould MIJ, Kovaltsuk A, Marks C, Deane CM. CoV-AbDab: the coronavirus antibody database. Bioinformatics. 2021;37(5):734–5. https://doi.org/10.1093/bioinformatics/btaa739 PMID: 32805021
30. Jaffe DB, Shahi P, Adams BA, Chrisman AM, Finnegan PM, Raman N, et al. Dataset: Functional antibodies exhibit light chain coherence. 2022. https://doi.org/10.5281/zenodo.6348137
31. Phad GE, Pinto D, Foglierini M, Akhmedov M, Rossi RL, Malvicini E, et al. Clonal structure, stability and dynamics of human memory B cells and circulating plasmablasts. Nat Immunol. 2022;23(7):1076–85. https://doi.org/10.1038/s41590-022-01230-1 PMID: 35761085
32. Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, et al. HuggingFace's Transformers: State-of-the-art Natural Language Processing. arXiv. 2019.
33. Ansel J, Yang E, He H, Gimelshein N, Jain A, Voznesensky M, et al. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In: Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2, 2024. 929–47. https://doi.org/10.1145/3620665.3640366

34. Rajbhandari S, Rasley J, Ruwase O, He Y. ZeRO: Memory Optimizations Toward Training Trillion Parameter Models. arXiv. 2019. https://arxiv.org/abs/1910.02054
35. Gugger S, Debut L, Wolf T, Schmid P, Mueller Z, Mangrulkar S, et al. Accelerate: Training and inference at scale made simple, efficient and adaptable. 2022.
36. Hägele A, Bakouch E, Kosson A, Allal LB, Von Werra L, Jaggi M. Scaling laws and compute-optimal training beyond fixed training durations. arXiv. 2024.
37. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. J Mach Learn Res. 2011;12:2825–30. https://doi.org/10.5555/1953048.2078195
38. Waskom M. seaborn: statistical data visualization. JOSS. 2021;6(60):3021. https://doi.org/10.21105/joss.03021
39. Hunter JD. Matplotlib: A 2D Graphics Environment. Comput Sci Eng. 2007;9(3):90–5. https://doi.org/10.1109/mcse.2007.55