Token-Wise Residual Latent Adapters: Steering Seq2Seq Models for Protein Fitness Extrapolation

Anonymous Authors[1]

Abstract

Protein design requires extrapolating beyond training data to achieve higher fitness. State-of-the-art methods typically fine-tune billion-parameter language models end-to-end, often combined with external scorers, data distillation, and multiple rounds of iterative refinement. We introduce a residual latent adapter, a 5M parameter MLP inserted between the encoder and decoder of a frozen ProtT5-3B model, which learns a token-wise residual transformation on encoder embeddings via a simple MSE objective. In a single forward pass with no external scorer, RLA achieves comparable or superior fitness to methods requiring 600× more trainable parameters and multi-stage pipelines, particularly on the harder extrapolation benchmarks most relevant to practical protein engineering. Our results demonstrate that a compact residual transformation in latent space provides a simple, data-efficient, and compute-efficient approach to protein fitness extrapolation.

1. Introduction

Improving protein designs with generative models fundamentally requires extrapolation: models must propose sequences with functional properties beyond those observed during training (Madani et al., 2020). In practice, protein language models are pretrained on large general sequence databases and subsequently adapted to specific protein families using relatively small labeled datasets (Suzek et al., 2007; Rives et al., 2021). State-of-the-art methods such as direct preference optimization (DPO) with data distillation achieve strong extrapolation, but require training all parameters of the base model, often combined with external scorers and multiple rounds of iterative refinement (Rafailov et al., 2023; Karimi et al., 2025; 2024). Parameter-efficient alternatives such as LoRA reduce trainable parameters but typically underperform full fine-tuning in extrapolation quality (Hu et al., 2022).

We propose a different approach: freezing the a encoder-decoder language model (Raffel et al., 2020) entirely and training a residual latent adapter that operates on encoder embeddings to learn a token-wise transformation toward higher-fitness representations. The key insight is that the preference between a lower- and higher-fitness sequence can be captured as a residual update in embedding space, without modifying the encoder-decoder mapping itself. Existing latent-space methods typically aggregate token embeddings into a pooled sequence-level representation before applying updates (Lee et al., 2023); in contrast, our method operates directly on per-token representations, preserving positional alignment.

We validate this approach on GFP and AAV benchmarks with a T5 model, demonstrating that our 5M parameter adapter, trained with a simple MSE objective, achieves comparable or superior extrapolation to methods requiring 600× more trainable parameters and multi-stage pipelines. Our contributions are:

- We introduce a residual latent adapter between the encoder and decoder of a frozen T5 model for protein sequence extrapolation.
- We show that this simple approach matches or exceeds state-of-the-art methods, including those using external scorers and data distillation, particularly on harder extrapolation benchmarks.
- We demonstrate data efficiency, with extrapolation performance saturating at 10–20% of available training pairs.

2. Method

We freeze a pretrained ProtT5-3B encoder–decoder and insert a trainable residual latent adapter between them, which applies a token-wise function (approximated by a multilayer perceptron) to encoder embeddings via a residual update before decoding. Given paired lower- and higher-fitness

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

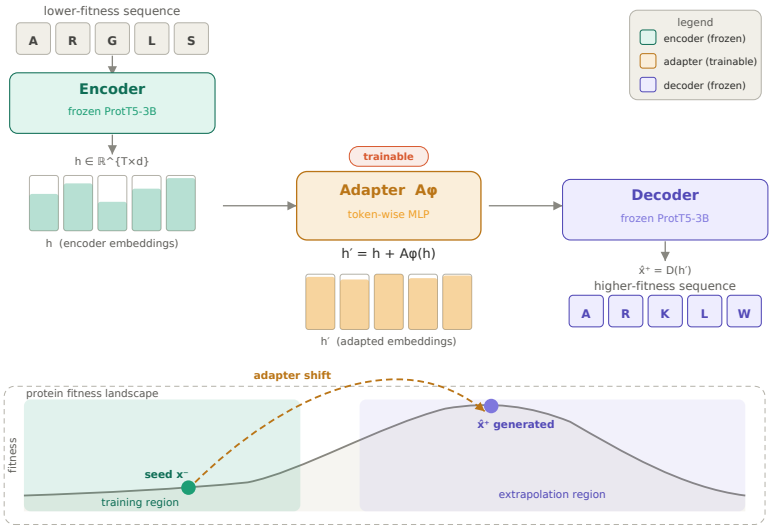


Figure 1. Overview of RLA approach for protein fitness extrapolation.

sequences, the adapter is trained with a mean squared error (MSE) loss to regress encoder embeddings of worse sequences toward those of better sequences.

Formally, our generative protein design model uses a frozen ProtT5-3B encoder-decoder backbone. Let $E(\cdot)$ and $D(\cdot)$ denote the encoder and decoder, respectively. Given an input (lower-fitness) sequence x^-, the encoder produces token-wise latent representations:

$$h = E(x^-) \in \mathbb{R}^{T \times d}$$

where T is the sequence length and d is the embedding dimension, with $h_t \in \mathbb{R}^d$ denoting the embedding at position t.

We introduce a trainable residual latent adapter A_ϕ, parameterized as a residual MLP, which is applied independently at each sequence position:

$$h'_t = h_t + A_\phi(h_t), \quad t = 1, \ldots, T$$

Equivalently, assume A_ϕ is applied row-wise, in matrix form we have:

$$h' = h + A_\phi(h)$$

The adapted token embeddings h' are then passed to the frozen decoder to generate an improved sequence:

$$\hat{x}^+ = D(h')$$

Only the adapter parameters ϕ (approximately 5M parameters) are optimized; both the encoder and decoder remain frozen. Training is performed on paired sequences (x^-, x^+), where x^+ has higher measured fitness than x^-. Let

$$h^+ = E(x^+)$$

denote the encoder embeddings of the higher-fitness target sequence. The adapter is trained to regress from h to h^+ using a mean squared error objective applied across all positions:

$$\mathcal{L}(\phi) = \mathbb{E}_{(x^-, x^+) \sim \mathcal{D}_{\text{pair}}} \left[\frac{1}{d \cdot T} \sum_{t=1}^{T} \left\| h_t + A_\phi(h_t) - h_t^+ \right\|_2^2 \right]$$

Standard dropout is applied within A_ϕ for regularization. This formulation learns a token-wise residual transformation in latent space, translating lower-fitness embeddings toward higher-fitness regions while preserving the pretrained encoder-decoder manifold.

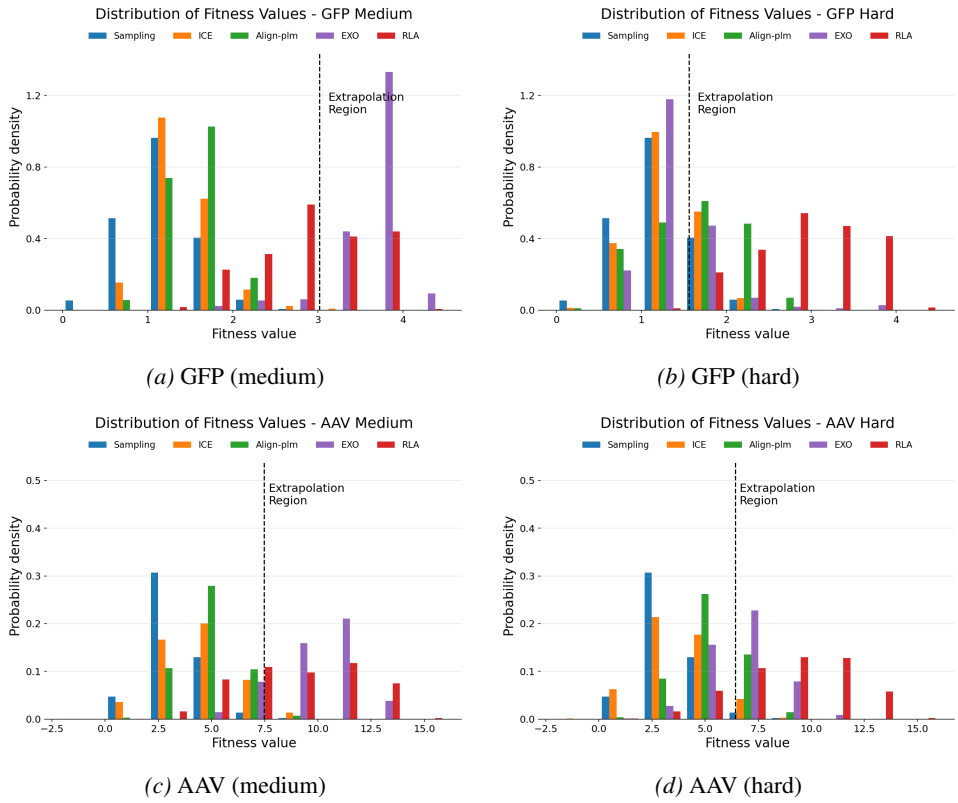
2.1. Dataset

We use the same AAV and GFP training pair splits as previous work to allow a direct benchmark of our method against existing results (Bryant et al., 2021; Sarkisyan et al., 2016; Karimi et al., 2025). Briefly, the dataset comprises AAV and GFP protein sequences with experimentally measured reward values. Each dataset is divided into two subtypes based on the difficulty of extrapolation: a *medium* dataset, in which the highest-performing proteins are 6 mutations away from training sequences, and a *hard* dataset, in which the distance is 7 mutations. The highest-performing sequences are held out during training, and the goal is to generate proteins that extrapolate into this unseen high-fitness region.

2.2. Training and Baselines

We compare RLA approach against several baselines:

- Full end-to-end fine-tuning of the pretrained T5 model



(a) GFP (medium) *(b)* GFP (hard)

(c) AAV (medium) *(d)* AAV (hard)

Figure 2. Distribution of predicted fitness values for all scorer-free methods. The dashed line indicates the extrapolation threshold. RLA produces a distribution shifted toward higher fitness values compared to all baselines, particularly on the hard splits.

- LoRA adapters applied to encoder and decoder layers
- State-of-the-art Direct Preference Optimization (DPO) with data distillation
- Negative control, where the residual adapter is randomly initialized and not trained

For all experiments, the residual adapter is trained solely on the pairwise "worse → better" data using MSE loss, and no other model parameters are updated. This setup isolates the effect of the residual adapter and allows a fair comparison with prior state-of-the-art methods in low-data extrapolative protein design.

2.3. Evaluation Metrics

To benchmark against prior work, we adopt the same four evaluation metrics used in previous studies:

1. **Extrapolation Percentage**: The percentage of generated sequences that fall within the extrapolation region of the fitness landscape, as assessed by evaluator models. This is our primary metric.
2. **Fitness100**: The average fitness of the top 100 generated sequences, measuring how far the generated sequences are from the training region in the fitness landscape.
3. **Distance100**: The average edit distance between each of the top 100 generated sequences and their closest top 100 ground truth sequences from the extrapolation region (sequences unseen by any model).
4. **Diversity100**: The median pairwise distance among the top 100 generated sequences. Higher diversity does not necessarily correlate with better performance, as random sequences can achieve maximal diversity, but it provides insight into the exploration–exploitation trade-off.

3. Results

We evaluate RLA against all published baselines from Karimi et al. (2025), including methods that employ external scorers, data distillation, and multiple rounds of iterative refinement. Our method uses none of these: it requires only a single forward pass through a 5M parameter adapter (roughly 600× fewer trainable parameters than the full 3B model), trained with a simple MSE objective on pairwise data.

Table 1. Comparison against scorer-free baselines on GFP and AAV benchmarks. Baseline results are reproduced from Karimi et al. (2025) (average of 5 runs). All methods shown use a single generation pass with no external scorer. RLA trains 5M parameters versus 3B for ICE and EXO.

	Hard				Medium			
Method	Ext.↑	Fit.$_{100}$ ↑	Dist.$_{100}$ ↓	Div.$_{100}$	Ext.↑	Fit.$_{100}$ ↑	Dist.$_{100}$ ↓	Div.$_{100}$
AAV								
Sampling	1.64	7.42	4.49	7.18	1.64	7.42	4.49	7.18
ICE	5.58	8.18	9.08	13.56	4.59	9.43	7.72	11.49
Align-plm	20.76	9.01	7.60	8.16	3.49	8.66	7.29	6.22
EXO	52.75	10.95	5.30	8.46	**85.07**	13.18	**1.64**	1.08
RLA	**80.70**	**13.95**	**2.31**	4.29	64.32	**14.00**	2.09	3.73
GFP								
Sampling	18.52	1.94	10.21	20.10	18.52	1.94	10.21	20.10
ICE	27.16	2.07	10.93	15.76	0.16	2.39	8.47	14.41
Align-plm	54.45	2.55	9.64	4.17	0.00	2.12	6.13	5.41
EXO	24.27	2.81	9.08	14.96	**92.92**	**4.04**	2.13	2.57
RLA	**99.05**	**3.95**	**1.01**	2.27	41.60	3.93	**0.98**	2.10

Table 2. ...

	Hard				Medium			
Method	Ext.↑	Fit.$_{100}$ ↑	Dist.$_{100}$ ↓	Div.$_{100}$	Ext.↑	Fit.$_{100}$ ↑	Dist.$_{100}$ ↓	Div.$_{100}$
AAV								
ICE+scorer	37.01	10.26	6.50	9.90	33.17	10.80	6.52	9.89
BiGGS	16.80	10.85	5.70	6.38	4.88	10.21	8.05	8.34
LatprotRL	64.82	13.29	2.45	4.67	38.63	12.53	2.83	5.21
EXO+scorer	84.04	**14.25**	1.52	2.53	**94.23**	13.90	**2.00**	3.05
EXO (Comb. distill.)	78.19	13.97	1.74	1.70	70.85	13.24	2.85	2.91
EXO (Comb. distill.)+scorer	**98.96**	14.21	**1.51**	1.08	84.71	13.19	3.14	3.34
RLA	80.70	13.95	2.31	4.29	64.32	**14.00**	2.09	3.73
GFP								
ICE+scorer	14.87	1.96	9.52	18.19	0.02	2.53	8.22	15.68
BiGGS	**99.53**	3.83	3.48	6.01	**55.50**	3.89	4.13	5.74
LatprotRL	88.28	3.88	1.48	2.86	38.22	3.92	1.56	3.04
EXO+scorer	50.91	3.79	1.73	3.08	58.09	**3.96**	2.75	4.04
EXO (Comb. distill.)	65.86	3.65	2.42	4.07	28.84	3.82	2.24	3.14
EXO (Comb. distill.)+scorer	71.15	3.75	2.10	3.46	32.41	3.86	2.16	3.19
RLA	99.05	**3.95**	**1.01**	2.27	41.60	3.93	**0.98**	2.10

3.1. Main Results

Table 1 and Figure 2 compare our method against scorer-free baselines that, like ours, use a single generation pass with no external scorer. Among ProtT5-based methods, RLA outperforms EXO on 3 of 4 datasets in top-100 fitness while training 600× fewer parameters. On AAV Hard, we achieve 13.95 fitness and 80.70% extrapolation versus EXO's 10.95 and 52.75%. On GFP Hard, we reach 3.95 fitness and 99.05% extrapolation versus 2.81 and 24.27%. We also achieve the lowest distance to ground truth on all 4 datasets. GFP Med is our weakest setting in extrapolation percentage (41.60% versus EXO's 92.92%), though our fitness remains competitive (3.93 versus 4.04).

Karimi et al. (2025) conducted a series of ablations to improve their results, including adding an external scorer for iterative candidate selection, data distillation to retrain on generated sequences, and combining both strategies. Table 2 compares RLA against these enhanced variants, as well as BiGGS and LatprotRL which use different model architectures entirely. Even without any of these enhancements, RLA remains competitive or superior on fitness and distance. On AAV Med, RLA achieves the highest fitness of any method (14.00), surpassing EXO+scorer (13.90) and the full distillation pipeline (13.19). On GFP Hard, RLA fitness (3.95) exceeds all methods including BiGGS (3.83) and LatprotRL (3.88), with the lowest distance to ground truth (1.01).

RLA is designed as a single-pass generator. Repeated application of the learned residual transformation tends to converge due to the limited capacity of the residual network, a trade-off inherent to lightweight models. While this limits

Table 3. Ablation study comparing RLA against LoRA-SFT and an embedding noise perturbation negative control ($\sigma = 0.25$). All methods use the same frozen ProtT5-3B backbone.

	Hard				Medium			
Method	Ext.↑	Fit.$_{100}$ ↑	Dist.$_{100}$ ↓	Div.$_{100}$	Ext.↑	Fit.$_{100}$ ↑	Dist.$_{100}$ ↓	Div.$_{100}$
AAV								
RLA	**80.70**	**13.95**	**2.31**	4.29	**64.32**	**14.00**	**2.09**	3.73
LoRA-SFT	11.01	8.18	9.13	12.67	15.58	9.83	7.31	7.72
Noise (σ=0.25)	2.53	7.95	4.42	7.15	0.70	7.88	4.79	6.52
GFP								
RLA	**99.05**	**3.95**	**1.01**	2.27	**41.60**	**3.93**	**0.98**	2.10
LoRA-SFT	28.20	2.13	10.87	7.16	0.25	2.37	7.83	11.90
Noise (σ=0.25)	72.20	2.77	4.98	9.64	0.45	2.84	4.75	8.94

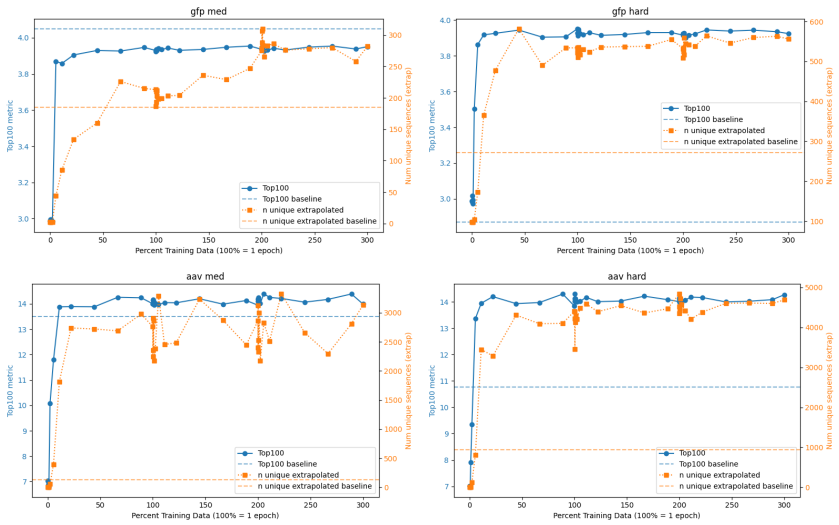


Figure 3. Data ablation results for GFP and AAV benchmarks (medium and hard splits). Each panel shows Top-100 fitness (left y-axis) and number of unique designs (right y-axis) as a function of the fraction of training data used. Dashed orange and blue lines indicate the published state-of-the-art baseline for each respective metric.

the use of iterative design strategies, it does not diminish the quality of the initial generation, which already matches or exceeds methods requiring substantially more complex pipelines.

3.2. Low-Parameter Ablations

To evaluate alternative parameter-efficient approaches, we compare against LoRA adapters applied to the encoder and decoder layers (Table 3). LoRA-SFT substantially underperforms RLA across all settings: on AAV Hard, LoRA achieves a top-100 fitness of 8.18 compared to our 13.95. This suggests that low-rank weight updates distributed across the model are less effective than a targeted residual transformation in latent space for this task.

To further verify that the adapter's improvements are not simply an artifact of perturbing the latent space, we evaluate a noise perturbation baseline. We add Gaussian noise to encoder embeddings ($h \mathrel{+}= \sigma \cdot \epsilon$, $\epsilon \sim \mathcal{N}(0, I)$). For $\sigma \in [0, 0.1]$, no change in output is observed; for $\sigma > 1$, the decoder produces no viable sequences. At $\sigma = 0.25$, the model generates sequences with poor fitness (e.g., AAV Hard top-100 fitness of 7.95 versus our 13.95), confirming that the adapter's gains arise from a learned directional transformation rather than arbitrary perturbation.

3.3. Latent Space Visualization

To visualize the effect of the residual adapter, we project encoder embeddings of training data, held-out extrapolation sequences, and generated designs into a shared UMAP space (Figure 4), with kernel density estimates for the training and held-out distributions. We track seed sequences before and after applying the adapter and observe a consistent shift toward regions associated with higher-fitness, held-

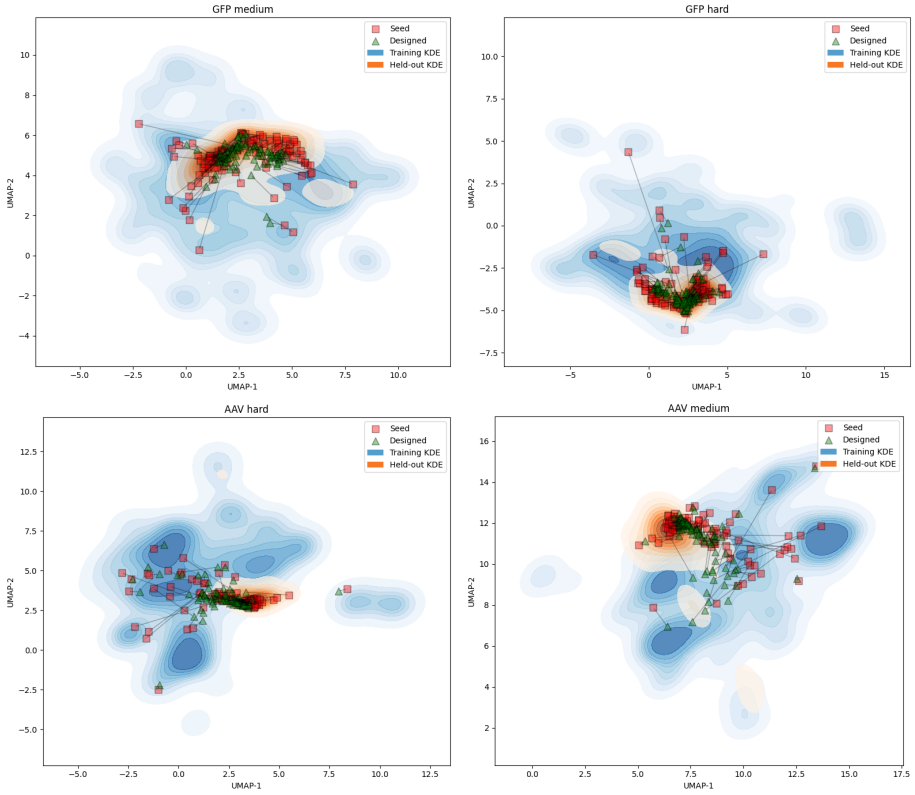


Figure 4. UMAP projections of ProtT5 encoder embeddings for GFP and AAV benchmarks (medium and hard splits). Background KDE contours show the density of 5,000 randomly sampled training sequences (teal) and held-out high-fitness extrapolation sequences (purple). Arrows connect the embeddings of 100 seed sequences before and after applying the residual adapter, illustrating the learned shift toward high-fitness regions of latent space.

out sequences. Several seeds exhibit low measured fitness despite lying close in latent space to the extrapolation region prior to adaptation. After applying the learned residual transformation, these seeds are moved directly into high-density, high-fitness regions, suggesting that the adapter learns a structured mapping toward favorable areas of the latent landscape.

3.4. Data Efficiency

We perform data ablation experiments by training on progressively smaller fractions of the pairwise training data (Figure 3). Since training pairs are constructed combinatorially from unique sequences, reducing the number of pairs does not necessarily remove unique sequences from the training set, but rather reduces the number of pairwise orderings the model observes. Despite this, extrapolation performance saturates at approximately 10–20% of the original pairs, already exceeding prior state-of-the-art, suggesting that the adapter learns the residual transformation efficiently from a relatively small number of pairwise comparisons.

4. Conclusion

Our results demonstrate that a 5M parameter residual latent adapter between a frozen ProtT5-3B encoder and decoder can match or exceed the extrapolation performance of methods that train the full 3B-parameter model with complex multi-stage pipelines. In a single forward pass with no external scorer, RLA produces comparable fitness and up to $4\times$ more unique extrapolating designs than the prior state-of-the-art, with the largest gains on the harder benchmarks most relevant to practical protein engineering. The simplicity of the approach, a single MLP trained with MSE loss, combined with its data efficiency and low compute requirements, makes it well suited for rapid iteration in experimental protein design workflows. We note that the position-wise residual formulation is most effective when the per-residue mutation rate is sufficiently high, as in the AAV benchmarks, and future work could explore position-aware loss weighting or masking strategies to improve performance on proteins with lower mutation rates such as GFP.

References

Bryant, D. H. et al. Deep mutational scanning of adeno-associated virus capsid. *Nature Biotechnology*, 2021. URL https://www.nature.com/articles/s41587-021-00854-4.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. 2022. URL https://arxiv.org/abs/2106.09685.

Karimi, M., Banerjee, S., Jaakkola, T., Dubrov, B., Shang, S., and Benson, R. Extrapolative protein design through triplet-based preference learning. 2024.

Karimi, M., Banerjee, S., Jaakkola, T., Dubrov, B., Shang, S., and Benson, R. Data distillation for extrapolative protein design through exact preference optimization. 2025.

Lee, M., Vecchietti, L. F., Jung, H., Ro, H., Cha, M., and Kim, H. M. Protein sequence design in a latent space via model-based reinforcement learning. 2023. URL https://openreview.net/forum?id=OhjGzRE5N6o. under review.

Madani, A. et al. Progen: Language modeling for protein generation. *arXiv*, 2020. URL https://arxiv.org/abs/2004.03497.

Rafailov, R. et al. Direct preference optimization: Your language model is secretly a reward model. *arXiv*, 2023. URL https://arxiv.org/abs/2305.18290.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21 (140):1–67, 2020.

Rives, A. et al. Biological structure and function emerge from scaling unsupervised learning. *Science*, 2021. URL https://www.science.org/doi/10.1126/science.abe5650.

Sarkisyan, K. S. et al. Local fitness landscape of the green fluorescent protein. *Nature*, 2016. URL https://www.nature.com/articles/nature17995.

Suzek, B. E. et al. Uniref clusters: A comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 2007. URL https://academic.oup.com/bioinformatics/article/23/10/1282/197795.

Supplementary Figures

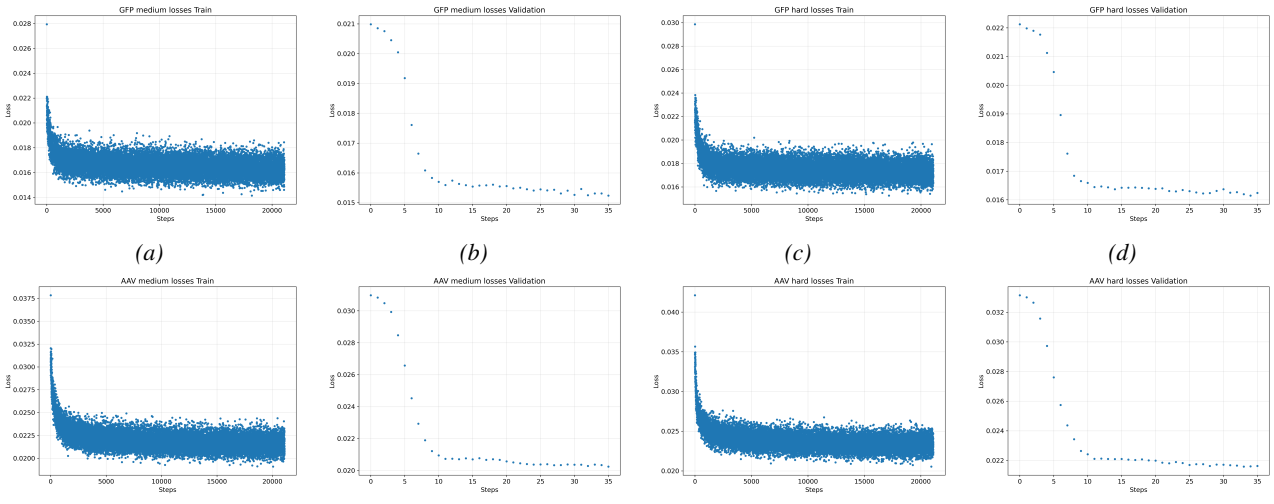


Figure 1. Adapter training and validation losses for each dataset.