Complementary Distillation for Protein Language Models

Edward Wijaya
wijaya@stemrim.com

Abstract

Large autoregressive protein language models (700M+ parameters) can generate novel protein sequences *de novo*, yet their computational demands preclude deployment on commodity hardware and limit throughput in high-throughput screening workflows. While knowledge distillation has been applied to masked protein language models and domain-specific causal models, no systematic study has addressed general-purpose autoregressive protein language models. We present a distillation framework combining standard temperature-scaled knowledge distillation with two protein-specific enhancements: uncertainty-aware position weighting, which emphasizes biologically variable regions via teacher entropy, and calibration-aware label smoothing, which regularizes teacher distributions to improve student calibration. A central finding is the *complementary effect*: each enhancement individually degrades distillation quality, yet their combination yields a 53% perplexity improvement over baseline distillation—a result we explain through an information-theoretic analysis of noise amplification and signal filtering. We train three compressed student models at 3.8–20× compression ratios, all of which outperform their respective baselines (31–87% perplexity improvement). The resulting models achieve 2.6–6.1× inference speedup while preserving amino acid distributions consistent with natural proteins (KL divergence < 0.015), enabling practical protein design on consumer-grade GPUs.

1 Introduction

Protein language models (pLMs) trained on evolutionary sequence data now enable computational protein design [1–3]. By learning the statistical patterns of natural protein sequences, autoregressive pLMs can generate novel sequences *de novo* with properties resembling those found in nature [1]. Among these, ProtGPT2—a GPT-2 architecture model [4] with 738 million parameters trained on UniRef50 [5]—has demonstrated the ability to produce sequences with natural amino acid distributions, plausible secondary structure content, and globular characteristics.

However, the computational cost of large pLMs creates a significant barrier to practical deployment. ProtGPT2 requires high-end GPUs for inference, generates sequences at limited throughput ($\sim$3 seconds per sequence), and cannot be deployed on edge devices or in resource-constrained laboratory settings. These constraints are particularly acute in biopharma applications such as ML-guided antibody affinity maturation [6] and enzyme engineering campaigns [3], where evaluating thousands to millions of candidate sequences demands both high throughput and cost-effective deployment.

Knowledge distillation [7] offers a principled approach to model compression by training a smaller student model to mimic the probability distributions of a larger teacher model. The key insight of Hinton et al. is that temperature-softened output distributions encode rich inter-class relationships—"dark knowledge"—that one-hot labels cannot convey. For protein sequences, these soft distributions capture amino acid substitution patterns: the teacher's prediction that position t should be leucine, with isoleucine and valine as secondary preferences, conveys evolutionary constraints that a hard label alone cannot express.

Distillation has been successfully applied to masked protein language models. DistilProtBERT [8] compressed ESM-style models [2] using response-based distillation, and MTDP [9] introduced multi-teacher distillation for protein representations. For causal protein LMs, SpiderGPT [10] applied distillation to a domain-specific model trained on 592 spider silk sequences. However, *no systematic study has addressed distillation for general-purpose autoregressive protein language models*—despite these being the models required for open-ended *de novo* sequence design.

We address this gap with a distillation framework that combines standard Hinton-style knowledge distillation with two protein-specific enhancements: (1) *uncertainty-aware position weighting*, which uses teacher entropy to emphasize biologically variable regions during distillation, and (2) *calibration-aware label smoothing*, which applies confidence-dependent smoothing to teacher distributions to improve student calibration [11, 12]. Our central finding is a *complementary effect*: uncertainty weighting alone increases perplexity by 95% and calibration smoothing alone increases it by 109%, yet their combination improves perplexity by 53% over baseline distillation. We provide a mechanistic explanation grounded in information theory: smoothing acts as a noise filter on teacher distributions, while weighting amplifies the cleaned signal at biologically important positions.

Our contributions are as follows:

1. The first systematic study of knowledge distillation for general-purpose autoregressive protein language models.
2. Two protein-specific distillation enhancements: uncertainty-aware position weighting and calibration-aware label smoothing.
3. Discovery and mechanistic explanation of the complementary effect, where individually harmful modifications combine for substantial improvement.
4. A comprehensive evaluation framework spanning perplexity, calibration (ECE), amino acid distributional fidelity, and inference benchmarks.
5. Open-source compressed models at three scales (37M, 78M, 194M parameters) available on HuggingFace.

2 Results

2.1 Ablation reveals complementary effect

To assess the contribution of each enhancement, we conducted a 2×2 ablation study using the Tiny architecture (4 layers, 4 heads, 256 embedding dimensions), toggling uncertainty-aware position weighting and calibration-aware label smoothing independently (Table 1). All four configurations used identical training hyperparameters ($T = 2.0$, $\alpha = 0.5$, learning rate $= 10^{-3}$, 3 epochs).

The baseline (standard Hinton-style distillation) achieved a perplexity of 18.95. Applying uncertainty weighting alone degraded perplexity to 36.89 (+95%), while calibration smoothing alone degraded it further to 39.64 (+109%). Both individual enhancements also increased KL divergence from the teacher and worsened expected calibration error (ECE). Taken individually, neither enhancement appears beneficial.

However, when both enhancements are applied simultaneously, perplexity drops to 8.93—a 53% improvement over baseline and a 75–77% improvement over either individual enhancement. KL divergence decreases from 2.23 to 1.62, and ECE improves from 0.274 to 0.216. This complementary

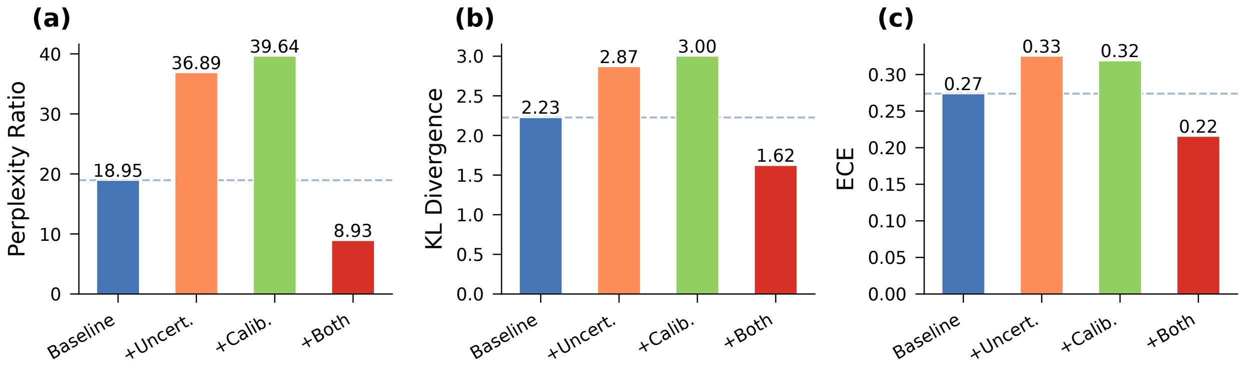

Figure 1: Ablation study showing the complementary effect. Each enhancement individually degrades distillation quality (higher perplexity, higher KL divergence, higher ECE), but their combination yields a 53% perplexity improvement over baseline.

Table 1: Ablation study results on Tiny architecture (4L/4H/256E). Each enhancement individually degrades distillation quality, but their combination yields a 53% improvement over baseline. PPL: perplexity; KL: KL divergence from teacher; ECE: expected calibration error.

Configuration	Uncertainty	Calibration	PPL	KL Div	ECE	vs. Baseline
Baseline (standard KD)	×	×	18.95	2.23	0.274	—
+Uncertainty only	✓	×	36.89	2.87	0.325	+95%
+Calibration only	×	✓	39.64	3.00	0.319	+109%
+Both (synergy)	✓	✓	**8.93**	**1.62**	**0.216**	**−53%**

effect, where two individually harmful modifications combine to produce substantial improvement, is the central finding of this work (Fig. 1).

2.2 Scaling across model sizes

To test whether the complementary effect generalizes beyond the ablation architecture, we trained paired baseline and synergy models at three scales: Tiny (20× compression), Small (9.4×), and Medium (3.8×). Based on the ablation finding that synergy training requires careful learning rate selection, we adopted a protocol using approximately half the baseline learning rate with 500 steps of linear warmup (see Methods for details).

Table 2 and Fig. 2 show that synergy models outperform baselines at all three scales. The improvement is largest at the highest compression ratio (87% at 20× compression for Tiny) and decreases with scale (54% for Small, 31% for Medium). This trend is expected: as student capacity approaches teacher capacity, the marginal benefit of enhanced distillation diminishes because standard KD already transfers knowledge effectively.

2.3 Calibration analysis

Expected calibration error (ECE) measures the alignment between predicted confidence and empirical accuracy across binned probability intervals [11, 13]. We computed ECE with 10-bin quantization on held-out protein sequences (Fig. 3).

Synergy models improve calibration at the Tiny scale (ECE 0.183 vs. 0.345, a 47% reduction)

Table 2: Scaling results across three model sizes. Synergy models use both uncertainty weighting and calibration smoothing with adjusted learning rates and warmup. All synergy models outperform their respective baselines.

Scale	Method	Compression	PPL	ECE	Improvement
Tiny (512E)	Baseline	20×	39.91	0.345	—
	Synergy	20×	**5.06**	**0.183**	87%
Small (768E)	Baseline	9.4×	15.19	0.235	—
	Synergy	9.4×	**7.05**	0.259	54%
Medium (1024E)	Baseline	3.8×	3.72	0.169	—
	Synergy	3.8×	**2.58**	**0.135**	31%

and at the Medium scale (ECE 0.135 vs. 0.169, a 20% reduction). At the Small scale, however, the synergy model shows a minor ECE regression (0.259 vs. 0.235). This anomaly likely reflects the fact that the Small model was the only scale where no learning rate reduction was applied; the warmup schedule alone may not fully optimize calibration. Overall, synergy distillation improves student calibration at 2 of 3 scales, with the largest gains at higher compression ratios where miscalibration risk is greatest.

2.4 Biological validity

For protein language models intended for *de novo* sequence design, preserving biologically realistic amino acid usage is essential. We evaluated the amino acid frequency distributions of generated sequences against the natural distribution observed in UniProt [5] (Fig. 4).

All student models—both baseline and synergy—produce amino acid distributions closely matching the natural UniProt distribution, with KL divergence below 0.015 in all cases. Quantifying per-residue deviations via mean absolute deviation (MAD) from UniProt frequencies reveals a clear ordering: the teacher deviates least (MAD = 0.0034), followed by the synergy student (MAD = 0.0073), then the baseline student (MAD = 0.0089). The synergy model is closer to natural on 13 of 20 amino acids, indicating that the combined enhancements improve distributional fidelity relative to standard distillation.

2.5 Compression–quality tradeoff

Plotting perplexity against compression ratio for all models reveals a Pareto frontier (Fig. 5). Synergy models dominate baseline models at every compression level tested, offering strictly better perplexity at the same model size. The improvement is most pronounced at high compression ratios, suggesting that the complementary distillation framework is especially valuable when aggressive compression is required.

2.6 Practical deployment

Inference benchmarks on an NVIDIA L40S GPU show that student models achieve substantial speedups over the teacher (Fig. 6): 6.1× for Tiny, 4.5× for Small, and 2.6× for Medium models. These speedups, combined with reduced memory requirements, enable protein sequence generation on consumer-grade GPUs. At the Tiny model's throughput, screening a library of 10^6 candidate

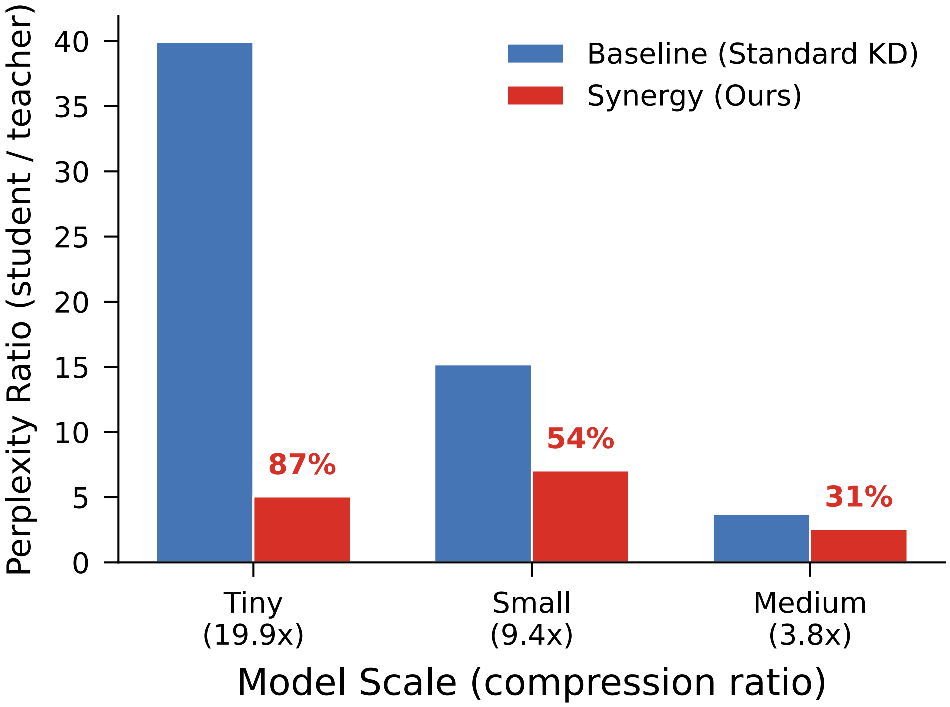

Figure 2: Perplexity ratio (student/teacher) across model scales. Synergy models outperform baselines at all three compression ratios, with the largest improvement at the highest compression (20×).

sequences—typical of combinatorial antibody or enzyme engineering campaigns—requires approximately 8 GPU-hours on a single consumer-grade device, compared to ∼50 GPU-hours for the full teacher. The Tiny model's 37M parameters fit within 2 GB of GPU memory, permitting deployment on shared laboratory workstations without dedicated accelerator infrastructure.

3 Discussion

Mechanistic explanation of the complementary effect. The central finding of this work—that individually harmful modifications combine to produce substantial improvement—has a mechanistic explanation grounded in information theory. Consider the teacher's probability distribution at each sequence position as containing both *signal* (genuine amino acid preferences reflecting protein biology) and *noise* (miscalibration artifacts from the teacher's own training).

Uncertainty-aware position weighting increases the loss contribution at high-entropy positions, effectively amplifying both signal and noise. Because noise dominates at high-entropy positions (where the teacher is uncertain and potentially miscalibrated), the net effect of weighting alone is to amplify noise more than signal, degrading distillation quality.

Calibration-aware label smoothing acts as a low-pass filter on teacher distributions, blending predictions toward the uniform distribution in proportion to teacher uncertainty. Applied alone, this attenuates both signal and noise, but signal is disproportionately affected because the fine-grained probability structure at uncertain positions—which encodes biologically meaningful substitution preferences—is smoothed away.

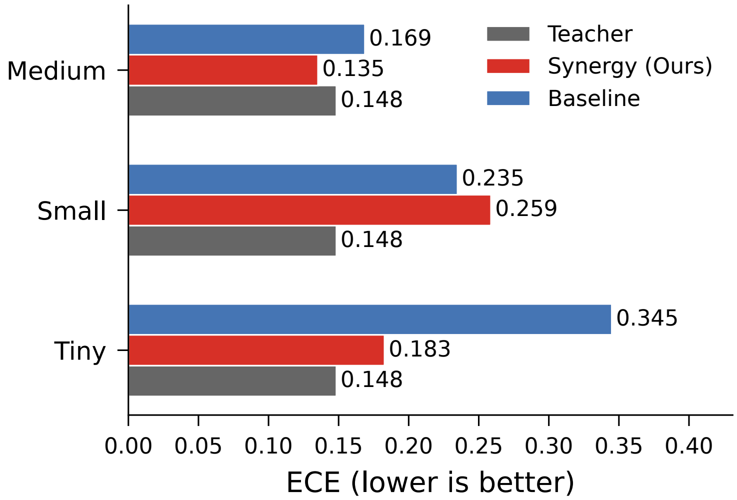

Figure 3: Expected calibration error across model scales. Synergy (red) achieves lower ECE than Baseline (blue) at Medium and Tiny scales, with the largest improvement at Tiny (0.183 vs. 0.345, 47% reduction). At Small scale, Synergy shows a minor regression (0.259 vs. 0.235). The teacher ECE (0.148) is shown for reference.

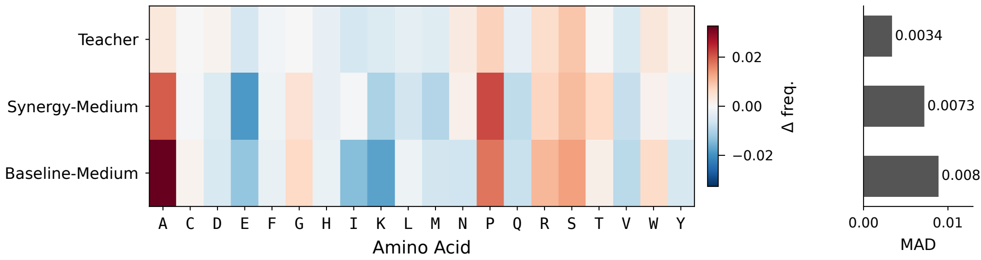

Figure 4: Amino acid frequency deviation from the natural UniProt distribution (left) with mean absolute deviation summary (right). The synergy student (MAD = 0.0073) deviates 18% less than the baseline (MAD = 0.0089), confirming that the combined enhancements improve distributional fidelity.

When both enhancements operate simultaneously, they address each other's failure mode. Calibration smoothing removes the noise that uncertainty weighting would otherwise amplify, while uncertainty weighting compensates for the signal attenuation introduced by smoothing by directing additional learning capacity toward the affected positions. The combined effect is *amplified but regularized* attention to variable positions: the student is instructed to "pay extra attention here" (weighting) while matching a denoised target (smoothing). This is analogous to a standard signal processing pipeline where amplification followed by filtering improves reception quality, whereas either operation alone degrades the signal-to-noise ratio.

Formally, the two enhancements modify different components of the per-position loss: weighting acts as an outer multiplier on the loss magnitude, while smoothing modifies the inner KL divergence target distribution. Because they operate on orthogonal aspects of the loss, their effects compose multiplicatively rather than additively, enabling synergistic interaction when the modifications address complementary failure modes.

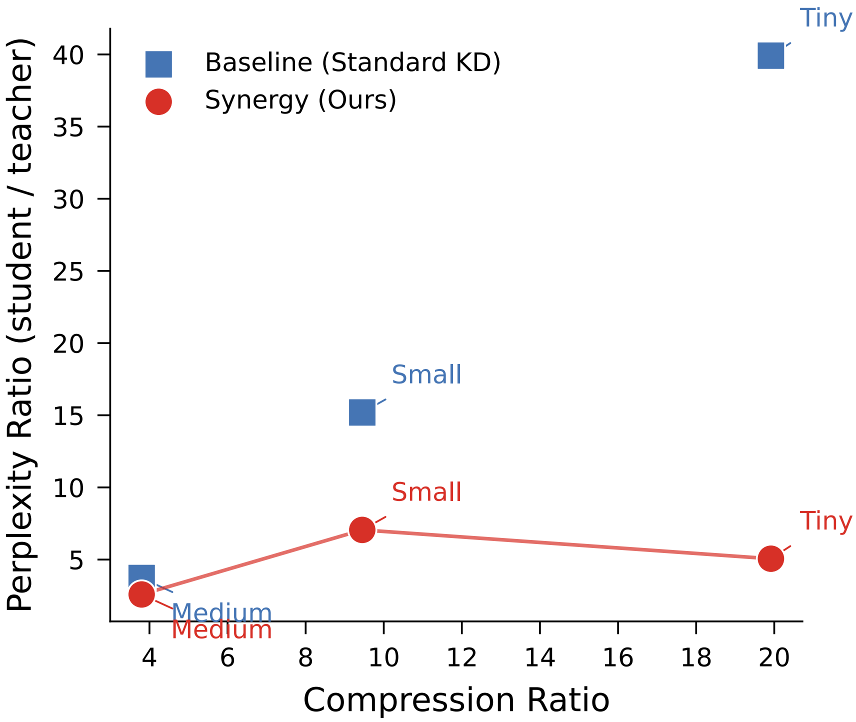

Figure 5: Compression–quality Pareto frontier. Synergy models (filled) dominate baseline models (open) at every compression ratio, achieving strictly better perplexity at the same model size.

Training dynamics and the role of warmup. Analysis of training logs reveals that the first ∼500 steps constitute a critical window for synergy training (Fig. 7). Without warmup, the modified objective—which is inherently easier to minimize from random initialization due to smoothed targets—allows the student to rapidly converge toward a degenerate minimum that achieves low training loss but poor generalization. This is evidenced by anomalously low initial loss values (6.62 vs. 7.94 for baseline at the Tiny scale) followed by severe train–evaluation misalignment.

With linear warmup over 500 steps, the near-zero initial learning rate forces the student to make incremental updates, learning basic token frequency patterns before encountering the full modified objective. By the time the learning rate reaches its target value, the student has already formed preliminary representations that constrain it to a generalizable region of the loss landscape. The enhanced objective then *refines* this foundation rather than corrupting it.

Scale-dependent effects. The synergy improvement decreases with model scale (87% → 54% → 31%), which we attribute to three factors. First, larger students have less to gain from regularization because they already approach teacher capacity (3.8× compression for Medium vs. 20× for Tiny). Second, baseline distillation improves approximately exponentially with scale, narrowing the gap. Third, larger students can better model the true distribution at variable positions natively, reducing the marginal benefit of the noise-filtering mechanism.

Learning rate scaling. A practical finding is that synergy training requires approximately half the baseline learning rate at matching scales, plus warmup. The smoothed targets create a loss landscape where the same nominal learning rate produces effectively larger functional steps; halving the learning rate compensates for this effect. This 0.5× scaling rule held at the Tiny and Medium

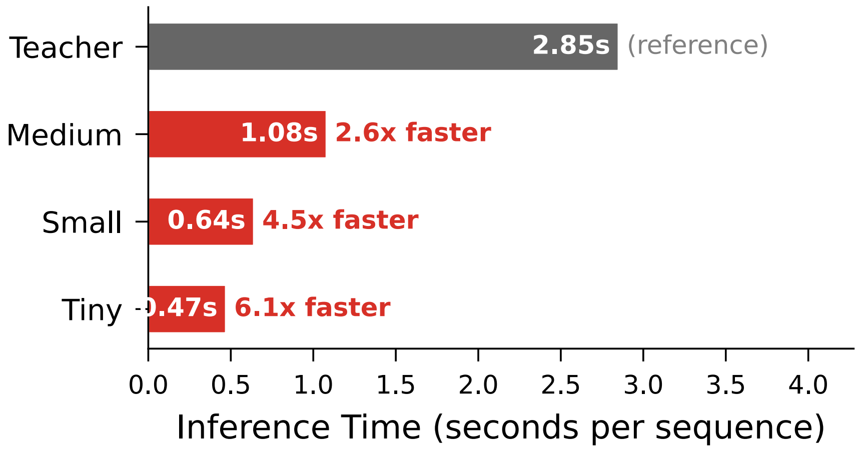

Figure 6: Inference speedup on an NVIDIA L40S GPU. Student models achieve 2.6–6.1× speedup over the ProtGPT2 teacher, enabling deployment on consumer-grade hardware.

scales. At the Small scale, where the baseline learning rate (5×10^{-4}) happened to already be appropriate, no reduction was needed.

Small model ECE regression. The Small synergy model shows a minor ECE regression (0.259 vs. 0.235 for baseline), the only scale where calibration worsened. This model was the only one trained without an explicit learning rate reduction, suggesting that the warmup schedule alone is not universally sufficient to optimize calibration. A targeted learning rate sweep at this scale would likely resolve the regression.

Practical implications for biopharma. The compressed models address concrete deployment bottlenecks in protein engineering pipelines. In antibody discovery, where language-model-guided affinity maturation has shown promise [6], the 6.1× speedup of the Tiny model reduces the cost of scoring large variant libraries by an order of magnitude, making iterative design–build–test cycles feasible on commodity hardware. For enzyme engineering campaigns that screen combinatorial sequence libraries [3], on-premise deployment of compact models avoids transmitting proprietary sequences to cloud APIs, satisfying data-governance requirements common in pharmaceutical settings. More broadly, the sub-2 GB memory footprint of the Tiny model enables integration into automated laboratory workflows where GPU resources are shared across instruments.

Limitations. This work has several limitations. First, we evaluate a single teacher model (ProtGPT2); generalization to other protein LMs such as ProGen [3] or non-protein causal LMs remains to be established. Second, ECE is computed at the token level and may not fully capture sequence-level calibration relevant to downstream applications. Third, structural plausibility is assessed via predicted metrics (pLDDT from ESMFold [14]) rather than experimental validation. Fourth, all experiments use a fixed smoothing factor ($\lambda = 0.1$) and temperature ($T = 2.0$); a joint hyperparameter search over the enhanced distillation objective could yield further improvements.

Future directions. We identify three extensions. Multi-teacher distillation, combining signals from diverse protein LMs, could provide more robust soft targets. The complementary effect should

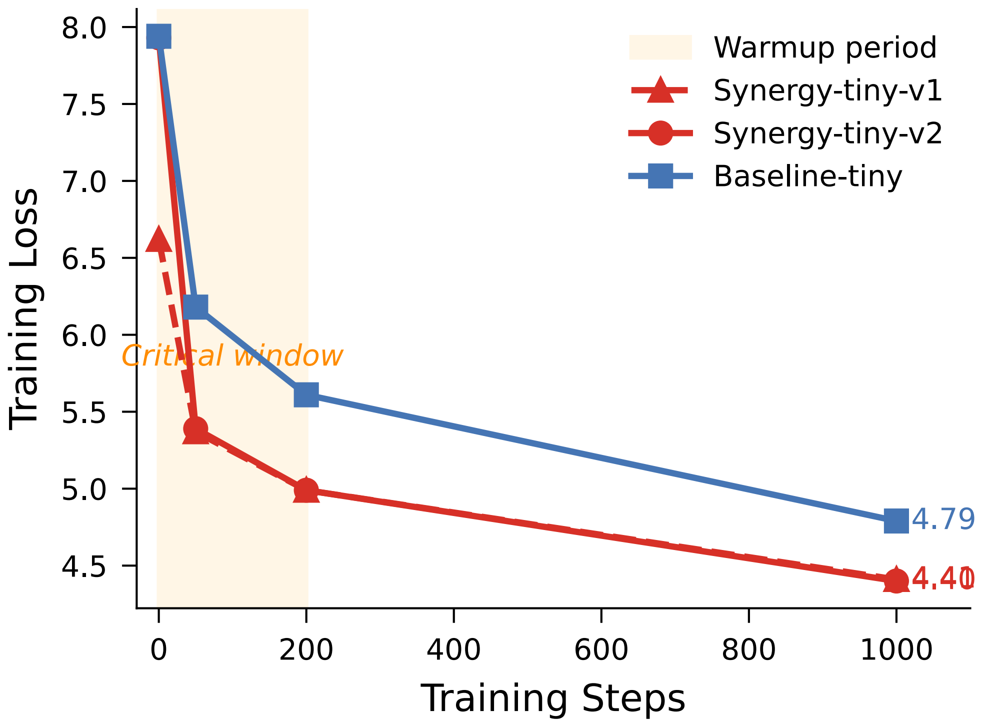

Figure 7: Training loss dynamics during the first 500 steps. Without warmup, the synergy objective allows rapid convergence to a degenerate minimum. Linear warmup over 500 steps constrains early optimization, enabling the student to reach a generalizable region of the loss landscape.

be tested on other autoregressive protein LMs and on non-protein biological sequence models. Finally, experimental validation of generated sequences—via wet-lab synthesis and characterization—would provide the strongest evidence of preserved biological function.

4 Methods

4.1 Standard distillation framework

We adopt the response-based knowledge distillation framework of Hinton et al. [7], adapted for autoregressive protein language modeling. Given a protein sequence $x = (x_1, \ldots, x_n)$ over vocabulary $\mathcal{V}$, the teacher and student models produce logit vectors $z_t^T, z_t^S \in \mathbb{R}^{|\mathcal{V}|}$ at each position t.

Temperature-scaled softmax. To reveal inter-class relationships in the teacher's predictions, logits are softened with temperature $\tau > 1$:

$$p_i^{(\tau)} = \frac{\exp(z_i/\tau)}{\sum_j \exp(z_j/\tau)} \tag{1}$$

Higher temperatures produce smoother distributions that expose the relative preferences among amino acids [7].

Soft loss. The soft loss measures the Kullback–Leibler divergence between temperature-scaled teacher and student distributions, averaged over sequence positions:

$$\mathcal{L}_{\text{soft}} = \frac{1}{n-1} \sum_{t=1}^{n-1} D_{\text{KL}}\left(p_T^{(\tau)}(\cdot|x_{\leq t}) \parallel p_S^{(\tau)}(\cdot|x_{\leq t}) \right) \tag{2}$$

Hard loss. The hard loss is the standard cross-entropy on ground-truth next-token labels:

$$\mathcal{L}_{\text{hard}} = -\sum_{t=1}^{n-1} \log p_S(x_{t+1}|x_{\leq t}) \tag{3}$$

Combined loss. The total distillation loss balances hard and soft objectives with coefficient $\alpha \in [0, 1]$, applying a τ^2 correction to maintain gradient magnitude under temperature scaling:

$$\mathcal{L} = \alpha \cdot \mathcal{L}_{\text{hard}} + (1-\alpha) \cdot \tau^2 \cdot \mathcal{L}_{\text{soft}} \tag{4}$$

The τ^2 factor compensates for the $1/\tau^2$ gradient attenuation introduced by temperature scaling in the softmax [7].

4.2 Uncertainty-aware position weighting

Protein sequences exhibit heterogeneous predictability: conserved structural positions (e.g., hydrophobic core residues) are highly predictable, while variable positions (loops, linkers, surface residues) admit multiple plausible amino acids. We exploit this structure by weighting each position's contribution to the soft loss in proportion to the teacher's prediction entropy.

Shannon entropy. At each position t, the teacher's uncertainty is quantified as:

$$u_t = H(p_T(\cdot|x_{<t})) = -\sum_{v \in \mathcal{V}} p_T(v) \log p_T(v) \tag{5}$$

where p_T uses temperature $\tau = 1$ (unscaled) to reflect the teacher's true predictive uncertainty.

Position weights. Entropies are min-max normalized per sequence and mapped to the range $[0.5, 1.0]$:

$$w_t = 0.5 + 0.5 \cdot \frac{u_t - \min(\mathbf{u})}{\max(\mathbf{u}) - \min(\mathbf{u})} \tag{6}$$

The floor of 0.5 ensures that even highly predictable positions contribute to the distillation loss, preventing the student from ignoring conserved regions entirely.

Weighted soft loss. The uncertainty-weighted soft loss replaces the uniform average in Eq. 2:

$$\mathcal{L}_{\text{soft}}^{\text{weighted}} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} w_t \cdot D_{\text{KL}}\left(p_T^{(\tau)}(\cdot|x_{\leq t}) \parallel p_S^{(\tau)}(\cdot|x_{\leq t}) \right) \tag{7}$$

where $\mathcal{T}$ denotes the set of non-padded positions.

Table 3: Model architectures and compression ratios. All models use the GPT-2 architecture with the ProtGPT2 tokenizer ($|\mathcal{V}| = 50,257$).

Model	Layers	Heads	Embedding dim	Parameters	Compression
Teacher (ProtGPT2)	36	20	1280	738M	1×
Medium	12	16	1024	∼194M	3.8×
Small	6	8	768	∼78M	9.4×
Tiny	4	4	512	∼37M	20×

4.3 Calibration-aware distillation

Well-calibrated confidence estimates are critical for protein engineering applications where model predictions guide experimental prioritization [11]. Neural networks, including large language models, tend to be poorly calibrated, and this miscalibration can be transferred during distillation. We introduce dynamic label smoothing [12] applied to teacher distributions, with smoothing intensity inversely proportional to teacher confidence.

Dynamic smoothing. At each position t, the smoothing intensity is:

$$\epsilon_t = \lambda \cdot \left(1 - \max_{v \in \mathcal{V}} p_T(v|x_{<t})\right) \tag{8}$$

where λ is a base smoothing factor. When the teacher is confident ($\max_v p_T(v) \approx 1$), smoothing is minimal ($\epsilon_t \approx 0$). When the teacher is uncertain, smoothing increases, regularizing the distribution.

Smoothed targets. The smoothed teacher distribution blends the original prediction with a uniform distribution:

$$\bar{p}_T(v) = (1 - \epsilon_t) \cdot p_T(v) + \frac{\epsilon_t}{|\mathcal{V}|} \tag{9}$$

Expected calibration error. We evaluate calibration using ECE with $B = 10$ equal-width bins [13]:

$$\text{ECE} = \sum_{b=1}^{B} \frac{|B_b|}{N} \left|\text{acc}(B_b) - \text{conf}(B_b)\right| \tag{10}$$

where B_b is the set of predictions falling in bin b, and $\text{acc}(B_b)$ and $\text{conf}(B_b)$ are the average accuracy and confidence within the bin, respectively.

4.4 Model architectures

All models use the GPT-2 architecture [4, 15] with varying depth and width. The teacher is ProtGPT2 (738M parameters) [1]. Student architectures span a 20× compression range (Table 3).

4.5 Training details

Data. We use a 10% subset of UniProt [5] protein sequences stored in Parquet format. Sequences are tokenized using the ProtGPT2 tokenizer with a maximum length of 1024 tokens.

Optimization. All models are trained with the AdamW optimizer for 3 epochs. We use Hinton et al.'s recommended defaults [7]: temperature $\tau = 2.0$ and balancing coefficient $\alpha = 0.5$. For calibration smoothing, the base smoothing factor is $\lambda = 0.1$.

Baseline models use learning rates of 10^{-3} (Tiny and Small) and 10^{-4} (Medium) without warmup. Synergy models use approximately half the baseline learning rate with 500 steps of linear warmup: 5×10^{-4} (Tiny and Small) and 5×10^{-5} (Medium). This learning rate reduction compensates for the smoother loss landscape created by label smoothing, which causes the same nominal learning rate to produce effectively larger optimization steps.

Hardware. The Medium model was trained on an NVIDIA L40S GPU (48 GB). Smaller models were trained on various NVIDIA GPUs with at least 24 GB of memory. Gradient accumulation (4 steps) was used to achieve an effective batch size of 32.

4.6 Data and code availability

Training code, evaluation scripts, and trained model weights are available at `https://github.com/ewijaya/protein-lm-distill`. Compressed models are hosted on HuggingFace at `littleworth/protgpt2-disti` Training data were derived from UniProt [5], which is freely available at `https://www.uniprot.org`.

References

[1] Noelia Ferruz, Steffen Schmidt, and Birte Höcker. ProtGPT2 is a deep unsupervised language model for protein design. *Nature Communications*, 13:4348, 2022.

[2] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021.

[3] Ali Madani, Ben Krause, Eric R Greene, Subu Subramanian, Benjamin P Mohr, James M Holton, Jose Luis Olmos, Caiming Xiong, Zhz Z Sun, Richard Socher, James S Fraser, and Nikhil Naik. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, 41:1099–1106, 2023.

[4] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019.

[5] The UniProt Consortium. UniProt: the universal protein knowledgebase in 2023. *Nucleic Acids Research*, 51(D1):D483–D489, 2023.

[6] Brian L Hie, Varun R Shanker, Duo Xu, Torben U J Bruun, Payton A Weidenbacher, Shaogeng Tang, Wesley Wu, John E Pak, and Peter S Kim. Efficient evolution of human antibodies from general protein language models. *Nature Biotechnology*, 42(2):275–283, 2024.

[7] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[8] Nadav Brandes, Dan Ofer, Yam Peleg, Nadav Rappoport, and Michal Linial. ProteinBERT: a universal deep-learning model of protein sequence and function. *Bioinformatics*, 38(8):2102–2110, 2022.

[9] Yijia Wang et al. MTDP: Multi-teacher knowledge distillation for protein representations. *Bioinformatics*, 2024.

[10] SpiderGPT Consortium. SpiderGPT: Knowledge distillation of a spider silk protein language model. *bioRxiv*, 2025.

[11] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 1321–1330, 2017.

[12] Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help? In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

[13] Mahdi Pakdaman Naeini, Gregory F Cooper, and Milos Hauskrecht. Obtaining well calibrated predictions using Bayesian binning into quantiles. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 2901–2907, 2015.

[14] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smeber, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candber, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.