SafePro: Generation of Safe and Functional Proteins via Cross-Task Conflict-Aware Alignment

Chongyang Gao
Northwestern University
Evanston, Illinois, USA
chongyanggao2026@u.northwestern.edu

Marco Postiglione
Northwestern University
Evanston, Illinois, USA
marco.postiglione@northwestern.edu

V.S. Subrahmanian
Northwestern University
Evanston, Illinois, USA
vss@northwestern.edu

Abstract

Protein language models (PLMs) enable scalable protein design but lack intrinsic safety constraints. We introduce SAFEPRO, a reinforcement-learning framework jointly optimizing structural foldability, toxicity avoidance, and virulence avoidance. SAFEPRO features two novel components: (1) *Cross-Task Attentive Fusion (CTAF)*, a joint reward model that employs gated cross-attention between toxicity and virulence experts to capture shared biological signals through selective cross-task sharing, and (2) *Task-Conflict-Saliency MGDA (TCS-MGDA)*, an adaptive weight-calibration procedure for Proximal Policy Optimization (PPO). TCS-MGDA targets gradient Gram-matrix dilution via parameter-space saliency filtering. On ProGen2, SAFEPRO improves composite alignment scores by $+5.21$ over the strongest fixed-weight baseline, significantly reducing toxicity ($0.0564 \rightarrow 0.0139$) and virulence ($0.6443 \rightarrow 0.5407$) while maintaining structural quality. External classifiers and homology screening also show reductions in selected toxicity and virulence proxies relative to the base policy. Among structurally designable outputs (scRMSD < 2 Å), SAFEPRO reduces the CTAF-flagged harmful fraction from 89% under fixed-weight PPO to 34% while producing more than twice as many designable sequences as the base model. We also curate a virulence dataset of $54{,}166$ sequences from VFDB, available on request.

1 Introduction

Protein language models (PLMs) trained on millions of sequences now enable *de novo* design of enzymes, binders, and therapeutics at scale [Madani et al., 2023, Nijkamp et al., 2023, Ferruz et al., 2022]. As these systems approach deployment, they raise a fundamental dual-use concern: *the ability to generate functional proteins is agnostic to whether those proteins are beneficial or harmful* [Urbina et al., 2022, Soice et al., 2023]. Current safeguards operate *downstream*, screening sequences only after generation. In contrast, large language models are increasingly aligned *at the model level* [Qi et al., 2025, Yi et al., 2024, Zeng et al., 2025]. An upstream alignment for PLMs would complement current screening by discouraging harmful outputs at generation time at the model level. Reinforcement learning from reward models [Ouyang et al., 2022, Bai et al., 2022], where the PLM is fine-tuned via PPO [Schulman et al., 2017] using learned feedback, is a natural approach to this problem, but adapting it to proteins faces two challenges.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Challenge 1: Harmfulness is multi-dimensional. Unlike natural language, where harmfulness is often scalar, protein risk spans at least two axes: *toxicity* (direct cellular damage) and *virulence* (capacity to cause disease) [Peterson, 1996, Do Vale et al., 2016]. Modeling them jointly creates a trade-off. Independent classifiers ignore shared signals due to data scarcity, but a shared backbone induces *negative transfer* [Standley et al., 2020] due to differing label distributions and class imbalance. A reward model must therefore share information selectively while preserving task-specific decision boundaries. **Challenge 2: Trade-offs shift during training.** PPO [Schulman et al., 2017] requires a scalar reward, forcing a weighting over objectives. Fixed weights are suboptimal because the trade-offs between foldability, toxicity, and virulence evolve as the policy explores the sequence space. While MGDA [Désidéri, 2012, Sener and Koltun, 2018] provides a principled mechanism for adaptive gradient weighting, it degrades substantially when applied to large PLMs under bounded rewards in $[0, 1]$. A possible explanation is **Gram-matrix dilution** (Section 4.3): large, aligned gradient components may obscure localized task differences when inner products are aggregated across the full parameter space.

SAFEPRO addresses these issues within a PPO alignment loop. **(i)** Our novel *Cross-Task Attentive Fusion (CTAF)* is a joint toxicity–virulence reward model built on a frozen ESM2 [Lin et al., 2022] backbone, where each task's expert attends to token-level representations from the other task's expert through gated cross-attention, enabling selective sharing without collapsing tasks. **(ii)** Our *Task-Conflict-Saliency MGDA (TCS-MGDA)* adapts weights during training by selecting parameters with higher standardized cross-task variation and switching between restricted and full Gram matrices using a cosine-gap threshold. Our main contributions are as follows:

- We formulate safe protein generation as a multi-objective alignment problem over foldability, toxicity, and virulence, and introduce **SAFEPRO**, a general framework that improves safety while preserving structural quality. We propose **CTAF**, a reward model that enables selective cross-task information sharing via gated attention, improving composite outcomes over independent classifiers in 5 of 6 fixed-weight PPO settings. We propose **TCS-MGDA**, a novel adaptive weighting method designed to mitigate Gram-matrix dilution, improving the composite score[1] by $+6.41$ over standard MGDA and $+5.21$ over the best fixed-weight baseline, without predefined weights.
- We show empirically that safety gains transfer to deployment-relevant outputs: among sequences passing a structural self-consistency filter (scRMSD < 2 Å), SAFEPRO reduces the harmful fraction from 89% to 34% while producing more than twice as many designable sequences as the base model.
- We curate a virulence dataset of $54{,}166$ sequences combining VFDB [Liu et al., 2022] with a filtered non-virulence set from 21 bacterial proteomes, available on request (Appendix A).

2 Related Work

RL-based protein design. ProDCARL [Sheng et al., 2026] aligns diffusion-based peptide generators using entropy-regularized policy gradients, while ApexAmphion [Cao et al., 2025] applies PPO to PLMs with composite activity and physicochemical rewards. SAFEPRO focuses on full-length protein generation, jointly optimizing structural quality, toxicity avoidance, and virulence avoidance with adaptive objective weights.

Safety-aware and multi-objective protein optimization. Knowledge-guided Preference Optimization (KPO) [Wang et al., 2025c] uses protein safety knowledge graphs to discourage harmful generations, while Logit Diff Amplification (LDA) [Burda et al., 2026] steers generation at inference time. MOFormer [Wang et al., 2025a], PepPFDPO [Wang et al., 2025b], and PepZOO [Liu et al., 2025] balance activity and toxicity through multi-objective or preference-based optimization. SAFEPRO combines joint toxicity–virulence reward modeling through CTAF with structural-quality optimization in a single alignment loop.

Conflict-aware multi-objective learning. MGDA [Désidéri, 2012, Sener and Koltun, 2018], PCGrad [Yu et al., 2020], GradNorm [Chen et al., 2018], CAGrad [Liu et al., 2021], and Nash-MTL [Navon et al., 2022] use gradient-based strategies for multi-objective learning. We compare

[1]The composite score averages structural quality with inverted safety scores. Higher is better (Section 4).

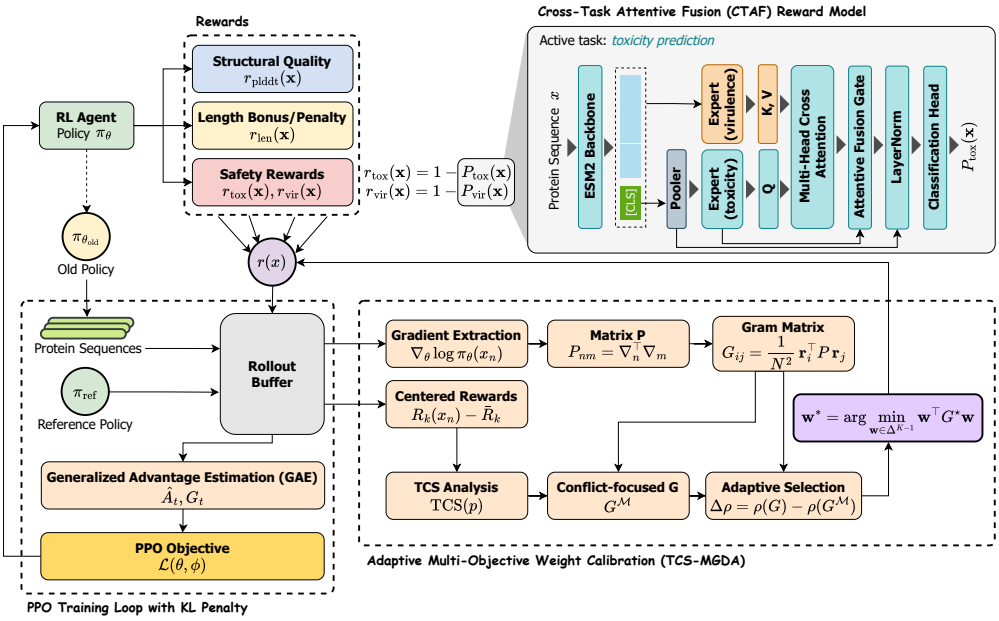

Figure 1: Overview of the SAFEPRO framework. CTAF supplies joint toxicity/virulence rewards, ESMFold supplies structural rewards, and TCS-MGDA dynamically calibrates the reward weights during PPO fine-tuning of the PLM policy.

MGDA, PCGrad, CAGrad, and Nash-MTL on protein generation with coupled safety objectives and ESMFold [Lin et al., 2023] structural feedback. TCS-MGDA targets Gram-matrix dilution through parameter selection and a cosine-gap switch between restricted and full Gram matrices for PPO weight calibration.

3 The SafePro Framework

SAFEPRO aligns protein generation with structural quality and toxicity and virulence avoidance (Figure 1). Its two novel modules are **CTAF** for reward estimation (Section 3.2) and **TCS-MGDA** for objective-weight calibration (Section 3.4). Both operate within a standard KL-regularized PPO loop (Section 3.3). Length regularization discourages short-sequence degeneration, while EMA smooths calibrated weights.

3.1 Alignment Objectives

Let π_θ be a policy with $\theta \in \mathbb{R}^D$ that generates a protein sequence $x = (x_1, \ldots, x_T)$ auto-regressively over an amino acid vocabulary. SAFEPRO aligns π_θ by maximizing a multi-dimensional reward vector $\mathbf{r}(x)$ over a *utility objective* (r_{util}) and two *safety objectives* ($r_{\text{tox}}, r_{\text{vir}}$), combined as

$$\mathcal{R}(x) = w_{\text{util}}\, r_{\text{util}}(x) + \mathbf{w}_{\text{safe}}^\top \mathbf{r}_{\text{safe}}(x) + r_{\text{len}}(x), \tag{1}$$

where $r_{\text{util}}(x) \in [0, 1]$ is the normalized predicted Local Distance Difference Test (pLDDT) score for structural foldability, $\mathbf{r}_{\text{safe}}(x) = [1-P_{\text{tox}},\, 1-P_{\text{vir}}]^\top$ encodes safety, and $w_{\text{util}}, \mathbf{w}_{\text{safe}}$ are non-negative weights dynamically calibrated by TCS-MGDA. The term $r_{\text{len}}(x)$ is a two-regime soft length regularizer that penalizes degenerate short sequences ($|x| < L_{\min} = 400$) and provides a capped bonus for sequences meeting the length threshold (full form in Appendix I). Safety is assessed through computational toxicity and virulence proxies, while structural quality is assessed through predicted confidence and self-consistency, not experimentally validated function.

3.2 Cross-Task Attentive Fusion (CTAF) Reward Model

Our novel CTAF model jointly scores toxicity and virulence to supply $P_{\text{tox}}(x)$ and $P_{\text{vir}}(x)$ to Eq. (1). Its design aims to mitigate negative transfer from hard parameter sharing while exploiting partial overlap between the two tasks: each task maintains a private expert and selectively incorporates the other expert's token-level information through a learned, input-dependent gate (Figure 1). This is especially important due to the large class imbalance of toxicity screening (Appendix G), where shared representations are often dominated by the majority class.

Backbone and experts. A frozen ESM2 [Lin et al., 2022] backbone produces hidden states $H \in \mathbb{R}^{T\times d}$ (d=1280). We pool the classification (CLS) token, the special token prepended to every input sequence whose final hidden state $H_{[0]}$ serves as a fixed-length summary of the whole sequence, as $h = \tanh(W_{\text{pool}}H_{[0]} + b_{\text{pool}})$. Two separately parameterized experts $e_{\text{tox}} = \text{Expert}_{\text{tox}}(h)$ and $e_{\text{vir}} = \text{Expert}_{\text{vir}}(h)$, each $\text{Linear}(d, d_{\text{exp}}) \to \text{GELU} \to \text{Dropout} \to \text{Linear}(d_{\text{exp}}, d)$ with $d_{\text{exp}} = 1024$, where GELU is the Gaussian Error Linear Unit activation, provide task-specific transformations with residual-compatible outputs; cross-attention still couples the experts.

Asymmetric cross-attention with adaptive gating. For active task τ, e_τ supplies a single query. The other task $\bar{\tau}$ supplies $H_{\bar{\tau}} = \text{Expert}_{\bar{\tau}}(H) \in \mathbb{R}^{T\times d}$, applying its expert to each token without pooling. Keys and values use projections shared across tasks:

$$Q = (W_Q^{(\tau)} e_\tau)^\top \in \mathbb{R}^{1\times d}, \qquad K = H_{\bar{\tau}} W_K^\top, \quad V = H_{\bar{\tau}} W_V^\top \in \mathbb{R}^{T\times d}. \tag{2}$$

Multi-head attention with $H_a = 4$ heads ($d_h = d/H_a = 320$) normalizes attention over the T token positions and yields $c = \text{MultiHead}(Q, K, V)^\top \in \mathbb{R}^d$. An input-dependent gate then modulates how much cross-task information is admitted:

$$\gamma = \sigma(W_{g2} \cdot \text{GELU}(W_{g1}[e_\tau; c]) + b_g), \qquad f_\tau = \gamma \cdot c + (1-\gamma)\cdot e_\tau. \tag{3}$$

The gate output bias b_g is initialized to -2.0. A residual+norm step $o_\tau = \text{LayerNorm}(h + \text{Dropout}(f_\tau))$ feeds a task-specific head $P_\tau(x) = \sigma(\text{head}_\tau(o_\tau))$.

Training. We alternate e_{tox} epochs on toxicity with e_{vir} epochs on virulence, with the backbone frozen, using separate AdamW optimizers (task-specific rates for task-private parameters, averaged rates for shared ones). We select the checkpoint maximizing $\frac{1}{2}(F1_{\text{tox}} + F1_{\text{vir}})$ on validation.

3.3 KL-Regularized PPO

SAFEPRO uses PPO [Schulman et al., 2017] with a token-level KL penalty against a frozen reference π_{ref} and a terminal biological reward:

$$r_t = -\beta\big(\log \pi_\theta(x_t \mid x_{<t}) - \log \pi_{\text{ref}}(x_t \mid x_{<t})\big) + \mathcal{R}(x)\,\mathbf{1}[t = T], \tag{4}$$

where β controls KL strength. We use GAE [Schulman et al., 2015] and clipped PPO updates. The full procedure and hyperparameters are given in Appendices I and J.

3.4 TCS-MGDA: Adaptive Multi-Objective Weight Calibration

The three objectives in Eq. (1) are not always aligned, and these conflicts evolve during training (Table 2). SAFEPRO addresses this with *TCS-MGDA*, which combines gradient-based MGDA weight adaptation [Désidéri, 2012, Sener and Koltun, 2018] with a Task-Conflict Saliency (TCS) mechanism that selects parameters with higher standardized cross-task variation for weight calibration, targeting the Gram-matrix dilution discussed in Section 4.3. Algorithm 1 summarizes this procedure.

Efficient Gram matrix computation. On calibration iterations, we extract per-sample gradients $\nabla_n \triangleq \nabla_\theta \log \pi_\theta(x_n) \in \mathbb{R}^D$ from a gradient-enabled policy scoring forward pass over the fixed generated sequences, separate from no-gradient generation. The per-objective gradient is the reward-weighted mean $\hat{g}_k = \frac{1}{N}\sum_n R_k(x_n)\,\nabla_n$. We compute the unregularized Gram matrix $\bar{G}$ without forming these D-dimensional vectors:

$$\bar{G}_{ij} = \hat{g}_i^\top \hat{g}_j = \frac{1}{N^2}\sum_{n,m} R_i(x_n)\,R_j(x_m)\,\nabla_n^\top \nabla_m = \frac{1}{N^2}\,\mathbf{r}_i^\top P\,\mathbf{r}_j, \tag{5}$$

Algorithm 1 TCS-MGDA for Multi-Objective PPO (SAFEPRO)

Require: Policy π_θ, reward models, calibration interval τ, EMA rate α, TCS threshold λ, top fraction ξ, length-reward threshold $L_{\min}$, Gram regularization ϵ_G

Initialize $\mathbf{w}_0 \leftarrow \mathbf{1}/K$
for iteration $t = 1, 2, \ldots$ **do**
  $\mathbf{w}_t \leftarrow \mathbf{w}_{t-1}$
  cal $\leftarrow (t \bmod \tau = 0)$ or $(t = 1)$
  Sample $\{x_n\}_{n=1}^N \sim \pi_\theta$ without gradients
  Score the fixed sequences with policy and reference; enable policy gradients if cal
  Compute rewards $R_k(x_n)$ for all k, n
  if cal **then**
    Compute $\nabla_n = \nabla_\theta \log \pi_\theta(x_n)$ and $P_{nm} = \nabla_n^\top \nabla_m$
    $\bar{G} \leftarrow \frac{1}{N^2} R^\top P R$ ▷ Eq. (5)
    $G \leftarrow \bar{G} + \epsilon_G I$
    Compute centered gradients S (Eq. (6)); L2-normalize rows, then apply per-parameter z-score
    Compute $\mathrm{TCS}(p)$ (Eq. (7)); select top-M set $\mathcal{M}$
    Compute $\bar{G}^{\mathcal{M}}$; set $G^{\mathcal{M}} \leftarrow \bar{G}^{\mathcal{M}} + \epsilon_G I$
    Compute cosine gap $\Delta\rho = \rho(G) - \rho(G^{\mathcal{M}})$
    $G^\star \leftarrow G^{\mathcal{M}}$ if $\Delta\rho > \lambda$, else $G^\star \leftarrow G$
    $\mathbf{w}^* \leftarrow \arg\min_{\mathbf{w} \in \Delta^{K-1}} \mathbf{w}^\top G^\star \mathbf{w}$ ▷ analytic for $K = 3$
    $\tilde{\mathbf{w}}_t \leftarrow \alpha\, \mathbf{w}^* + (1 - \alpha)\, \mathbf{w}_{t-1}$
    Post-process $\tilde{\mathbf{w}}_t$ to obtain $\mathbf{w}_t$ (Appendix J)
  end if
  Compute combined rewards and GAE advantages with $\mathbf{w}_t$
  Update π_θ with PPO
end for

where $\mathbf{r}_k \in \mathbb{R}^N$ stacks per-sample rewards for objective k and $P \in \mathbb{R}^{N \times N}$ is the gradient inner-product matrix. We write $G = \bar{G} + \epsilon_G I$ for the regularized matrix used in calibration.

Task-Conflict Saliency. For more stable alignment, SAFEPRO centers rewards before aggregating gradients. Let $\nabla_n^{(p)}$ denote the component of ∇_n at parameter p. The centered per-objective signal is

$$s_k^{(p)} = \frac{1}{N} \sum_{n=1}^{N} \big(R_k(x_n) - \bar{R}_k\big)\, \nabla_n^{(p)}, \tag{6}$$

where $\bar{R}_k$ is the per-objective mean reward. Centering expresses each sequence's reward relative to its objective-specific batch mean. As gradients are extremely sparse (magnitudes $\sim 10^{-7}$, concentrated in $<1\%$ of parameters), raw cross-objective variance collapses to zero almost everywhere. We therefore apply a two-stage scale-free normalization: (i) L2 normalization $\tilde{s}_k^{(p)} = s_k^{(p)} / (\|s_k\|_2 + \epsilon)$ reduces per-objective scale differences, and (ii) per-parameter z-scoring across objectives, $z_k^{(p)} = (\tilde{s}_k^{(p)} - \mu^{(p)})/(\sigma^{(p)} + \epsilon)$, standardizes the signals across objectives at each parameter. The TCS score aggregates the result:

$$\mathrm{TCS}(p) = \frac{1}{K} \sum_k |z_k^{(p)}|, \qquad \overline{\mathrm{TCS}} = \frac{1}{|\mathcal{A}|} \sum_{p \in \mathcal{A}} \mathrm{TCS}(p), \tag{7}$$

where $\mathcal{A}$ restricts to active parameters (magnitude $>10^{-4}$ of the global max), excluding inactive zeros. TCS measures standardized cross-task variation, not raw-gradient sign conflict.

Adaptive conflict-focused Gram matrix. SAFEPRO selects the top-M parameters by TCS as $\mathcal{M}$, with $M = \max(M_{\min}, \lfloor \xi D \rfloor)$ (ξ=0.05, $M_{\min}$=1000), in $\mathcal{O}(D)$ via partial sort. The normalized signals are used only to select $\mathcal{M}$; the restricted Gram uses the original reward-weighted gradients:

$$\bar{G}_{ij}^{\mathcal{M}} = \sum_{p \in \mathcal{M}} \hat{g}_i^{(p)} \hat{g}_j^{(p)}, \qquad G^{\mathcal{M}} = \bar{G}^{\mathcal{M}} + \epsilon_G I.$$

We compute the cosine statistic below from these regularized matrices. Rather than always using G or $G^{\mathcal{M}}$, we switch based on the cosine-similarity gap

$$\Delta\rho = \rho(G) - \rho(G^{\mathcal{M}}), \qquad \rho(G) = \frac{2}{K(K-1)} \sum_{i<j} \frac{G_{ij}}{\sqrt{G_{ii}G_{jj}}}, \tag{8}$$

choosing $G^\star = G^{\mathcal{M}}$ when $\Delta\rho > \lambda$ and $G^\star = G$ otherwise. The threshold λ ranges from zero-threshold switching (λ=0) to standard MGDA (λ=∞).

Weight update. Given $G^\star$, SAFEPRO solves $\mathbf{w}^* = \arg\min_{\mathbf{w}\in\Delta^{K-1}} \mathbf{w}^\top G^\star \mathbf{w}$. For $K = 3$, an analytic solver enumerates the seven simplex candidates (3 vertices, 3 edge optima, 1 interior Karush–Kuhn–Tucker (KKT) point [Karush, 1939]) and returns the global minimum without iteration. Let $\bar{G}^\star = G^\star - \epsilon_G I$ be the corresponding unregularized Gram matrix. The regularized objective and the squared norm of the weighted gradient in the selected parameter space satisfy

$$q^\star = {\mathbf{w}^*}^\top G^\star \mathbf{w}^* = \|\mathbf{d}^\star\|^2 + \epsilon_G \|\mathbf{w}^*\|_2^2, \qquad \|\mathbf{d}^\star\|^2 = {\mathbf{w}^*}^\top \bar{G}^\star \mathbf{w}^*. \tag{9}$$

Thus, subtracting $\epsilon_G\|\mathbf{w}^*\|_2^2$ from $q^\star$ recovers the squared gradient norm $\|\mathbf{d}^\star\|^2$, as opposed to the norm. We apply exponential moving average (EMA) smoothing with α=0.95 and post-processing (Appendix J), then recompute buffer rewards and GAE advantages using the resulting weights. TCS calibrates objective weights using the selected subspace, while PPO updates the full policy. The restricted quadratic objective is a calibration diagnostic, not a certificate of full-parameter Pareto stationarity.

4 Experimental Results

Our experiments answer four questions: **(Q1)** does CTAF outperform single-task and shared-backbone baselines (Section 4.2)? **(Q2)** is Gram-matrix dilution a real phenomenon on a deployed PLM (Section 4.3)? **(Q3)** does SAFEPRO's end-to-end pipeline outperform fixed-weight PPO and standard MGDA (Section 4.4)? **(Q4)** how sensitive is SAFEPRO to its key design choices (Section 4.5)? Throughout, we use a composite score $\mathcal{S} = \frac{1}{3}\big[\text{Mean pLDDT} + 100(1-P_{\text{tox}}) + 100(1-P_{\text{vir}})\big]$ (higher is better). Unless otherwise stated, alignment runs use `progen2-medium` as the base policy and our CTAF as the reward oracle, with N=1,000 sequences sampled per evaluation. We additionally evaluate ProtGPT2 in Appendix C. Full hyperparameters are in Appendix J.

4.1 Datasets

We use two datasets for reward-model training and evaluation, fully described in Appendix G. The *toxicity benchmark* of Zhu et al. Zhu et al. [2025] is derived from UniProt and split into 4,900/78/112 toxic and 9,800/853/1,735 non-toxic sequences for train/val/test, with CD-HIT-2D applied at a 40% identity threshold across splits. As no standardized protein-level virulence benchmark is publicly available, we additionally curate a new *virulence dataset* of 54,166 sequences combining 27,083 virulence-associated proteins from VFDB Liu et al. [2022] with 27,083 length-stratified negatives drawn from 21 phylogenetically diverse bacterial reference proteomes, after 40 keyword-based exclusion patterns and MD5-based VFDB leakage removal.

4.2 Q1: CTAF Reward Model Evaluation

We compare CTAF against two baseline families on the held-out test sets. All models share a frozen ESM2-t33 backbone and identical training regimes (Appendix J). *(i) Separate classifiers*: two independently trained heads, the standard single-task upper bound that pays no multi-task tax but doubles inference cost. *(ii) Shared-block multi-task*: a shared MLP trunk of depth $L \in \{1, 3, 5, 8\}$ blocks feeding two task-specific heads, the canonical hard parameter-sharing baseline. Varying L tests whether the shared representation merely needs more capacity to resolve the toxicity–virulence tension.

CTAF strictly dominates both baseline families. Table 1 shows *CTAF outperforms separate single-task classifiers on every metric* (toxic F1 +0.0584, toxic MCC +0.0636, virulence F1 +0.0268, virulence MCC +0.0544). Joint training is therefore not merely computationally efficient: it is strictly

Table 1: Test-set performance of SAFEPRO's CTAF against single-task and shared-block multi-task baselines. **Bold** = best per column; underline = second-best.

	Toxicity			Virulence		
Model	F1	MCC	AUROC	F1	MCC	AUROC
Separate (single-task)	0.8035	0.7907	0.9880	0.9158	0.8295	0.9708
Shared-block, L=1	0.7909	0.7853	0.9864	0.8756	0.7519	0.9479
Shared-block, L=3	0.7544	0.7534	0.9857	0.8837	0.7677	0.9528
Shared-block, L=5	0.7721	0.7686	0.9865	0.8791	0.7571	0.9489
Shared-block, L=8	0.7518	0.7509	0.9828	0.8782	0.7551	0.9444
CTAF (ours)	**0.8619**	**0.8543**	**0.9884**	**0.9426**	**0.8839**	**0.9824**

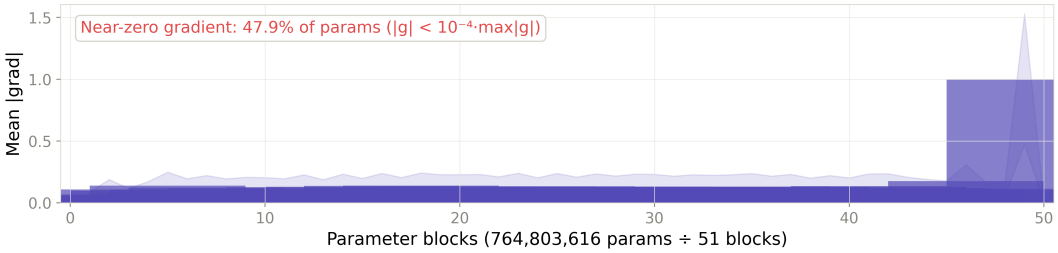

Figure 2: Mean gradient magnitude across N=32 samples. For each block, we plot $\frac{1}{N}\sum_n \|\nabla_n^{(b)}\|_2$, then divide by the peak block value so the y-axis is in units of *fraction of the peak block magnitude*; the shaded band shows $\pm\sigma$ across samples under the same normalization. The annotation reports the fraction of parameters with near-zero gradient, defined as $|g_i| < 10^{-4} \cdot \max_j |g_j|$ within each sample, averaged across the N samples.

better, indicating the two tasks carry complementary information. Hard parameter sharing, in contrast, under-performs even the single-task upper bound at every depth (toxic F1 0.75–0.79, virulence F1 near 0.88). Shared-block models trade precision for recall (0.62–0.69 vs 0.93–0.95), whereas CTAF reaches a balanced operating point (0.81/0.92); these results support the overall CTAF architecture but do not isolate the contribution of its experts.

Average gate values of 0.20 (toxicity) and 0.06 (virulence) confirm selective, predominantly task-private sharing, consistent with toxins being a strict subset of virulence factors. Wilcoxon signed-rank tests confirm significance for all toxicity comparisons ($p < 0.001$ vs. separate classifiers and all shared-block depths). For virulence, separation from the separate classifier is marginal ($p = 0.059$); among shared-block depths only Block-3 reaches significance ($p = 0.004$).

4.3 Q2: Empirical Evidence for Gram-Matrix Dilution

Figure 2 shows the mean per-block gradient magnitude on `progen2-medium` (D = 764,803,616, N=32): 47.9% of parameters carry essentially near zero gradient on any backward pass, and within the active half the output-projection block near index 49 exceeds any backbone block by an order of magnitude. Each $\bar{G}_{ij} = \sum_p \hat{g}_i^{(p)} \hat{g}_j^{(p)}$ pools two regimes into one scalar: well-sampled, large-magnitude, cross-objective-aligned products from the dominant block, and sparsely-sampled, small-magnitude, noisy products from the rest of θ. Aligned gradients with unequal task magnitudes can dominate calibration. TCS may improve PPO weighting by emphasizing coordinates with higher standardized cross-task variation.

TCS ranks parameters by standardized cross-task variation after reward centering, L2 normalization, and per-parameter z-scoring (Equations (6)–(7)). The top-M subset is intended to reduce the influence of dominant shared components on weight calibration, without guaranteeing raw-gradient conflict. We use $G^{\mathcal{M}}$ only when $\rho(G) - \rho(G^{\mathcal{M}}) > \lambda$, and retain G otherwise. Figure 7 shows positive cosine gaps in approximately 52% of calibrations. The per-sample view (Appendix K) shows sample-specific sparsity, motivating dynamic rather than fixed parameter selection.

Table 2: End-to-end alignment results on ProGen2-medium. Upper block: fixed-weight PPO with independent classifiers (Ind. Cls.) or CTAF as the training reward model. Lower block: adaptive gradient methods, inference-time steering, and preference optimization. All policies share the CTAF safety evaluator and ESMFold structural evaluation. Higher pLDDT and composite scores and lower safety scores are better.

Setting	Mean pLDDT		P(toxic)		P(virulent)		Composite	
	Ind. Cls.	CTAF	Ind. Cls.	CTAF	Ind. Cls.	CTAF	Ind. Cls.	CTAF
Base Model	$54.31_{\pm 15.91}$	—	$0.0635_{\pm 0.2013}$	—	$0.6525_{\pm 0.3861}$	—	60.90	—
$(0.3, 0.3, 0.4)$	$54.46_{\pm 16.21}$	$55.02_{\pm 15.93}$	$0.0494_{\pm 0.1731}$	$0.0564_{\pm 0.1852}$	$0.6687_{\pm 0.3791}$	$0.6443_{\pm 0.3882}$	60.88	**61.65**
$(0, 0.5, 0.5)$	$52.13_{\pm 14.96}$	$53.73_{\pm 15.92}$	$0.0577_{\pm 0.1861}$	$0.0500_{\pm 0.1731}$	$0.6926_{\pm 0.3680}$	$0.6804_{\pm 0.3775}$	59.03	**60.23**
$(0.5, 0, 0.5)$	$50.89_{\pm 14.84}$	$53.57_{\pm 15.55}$	$0.0622_{\pm 0.1898}$	$0.0564_{\pm 0.1819}$	$0.7025_{\pm 0.3647}$	$0.6710_{\pm 0.3798}$	58.14	**60.28**
$(0.5, 0.5, 0)$	$53.42_{\pm 16.11}$	$54.62_{\pm 15.69}$	$0.0471_{\pm 0.1655}$	$0.0492_{\pm 0.1726}$	$0.6863_{\pm 0.3738}$	$0.6639_{\pm 0.3848}$	60.03	**61.10**
$(1, 0, 0)$	$54.66_{\pm 15.62}$	$54.08_{\pm 16.00}$	$0.0547_{\pm 0.1882}$	$0.0495_{\pm 0.1708}$	$0.6557_{\pm 0.3828}$	$0.6753_{\pm 0.3803}$	**61.21**	60.53
$(0, 1, 0)$	$54.73_{\pm 16.02}$	$54.48_{\pm 15.77}$	$0.0531_{\pm 0.1809}$	$0.0524_{\pm 0.1753}$	$0.6676_{\pm 0.3821}$	$0.6607_{\pm 0.3826}$	60.89	**61.06**
$(0, 0, 1)$	$53.50_{\pm 15.47}$	—	$0.0519_{\pm 0.1717}$	—	$0.6664_{\pm 0.3764}$	—	60.56	—
	pLDDT		**CTAF toxicity**		**CTAF virulence**		**Composite**	
Standard MGDA	$55.83_{\pm 14.39}$		$0.0523_{\pm 0.1766}$		$0.6926_{\pm 0.3717}$		60.45	
PCGrad	$51.50_{\pm 15.13}$		$0.0287_{\pm 0.1312}$		$0.6473_{\pm 0.3944}$		61.30	
CAGrad	$53.37_{\pm 16.14}$		$0.0475_{\pm 0.1655}$		$0.6361_{\pm 0.3912}$		61.67	
Nash-MTL	$55.08_{\pm 15.74}$		$0.0402_{\pm 0.1594}$		$0.6417_{\pm 0.3898}$		62.30	
LDA	$53.85_{\pm 15.85}$		$0.0329_{\pm 0.1341}$		$0.6280_{\pm 0.3884}$		62.58	
DB-DPO	$53.52_{\pm 15.74}$		$0.0284_{\pm 0.1257}$		$0.6250_{\pm 0.3964}$		62.72	
SAFEPRO (TCS-MGDA)	**56.04**$_{\pm 14.66}$		**0.0139**$_{\pm 0.1164}$		**0.5407**$_{\pm 0.3685}$		**66.86**	

4.4 Q3: End-to-End Alignment

We evaluate SAFEPRO's full pipeline (CTAF reward modeling + KL-regularized PPO + TCS-MGDA weight calibration) against two ablations on the same backbone: *independent-classifier rewards* (replacing CTAF) under a grid of fixed weights $(w_{\text{tox}}, w_{\text{vir}}, w_{\text{pLDDT}})$, and *standard MGDA* [Désidéri, 2012, Sener and Koltun, 2018] (replacing TCS-MGDA) on top of CTAF. We also compare PCGrad [Yu et al., 2020], CAGrad [Liu et al., 2021], Nash-MTL [Navon et al., 2022], LDA [Burda et al., 2026], and database-paired DPO (DB-DPO) [Wang et al., 2025c] on ProGen2-medium using the same evaluation protocol. DB-DPO uses database-derived preference pairs and follows KPO's preference-learning rationale without reproducing its knowledge graph or graph-based pairing. Results are in Table 2, with implementation details in Appendix D.

(Q3a) CTAF rewards beat independent classifiers in 5 of 6 fixed-weight settings. CTAF improves the composite score in five of six paired settings; the exception is toxicity-only optimization. Gains are largest where a single safety task couples with structural quality ($+2.14$ at $(0.5, 0, 0.5)$; $+1.20$ at $(0, 0.5, 0.5)$) and more moderate under uniform or balanced-safety allocations ($+0.77$, $+1.07$). The composite score varies by more than three points across the weight grid, motivating adaptive calibration.

(Q3b) SAFEPRO outperforms fixed-weight and additional alignment baselines. SAFEPRO achieves the best mean values on all three objectives in Table 2: pLDDT 56.04, predicted toxicity 0.0139, and predicted virulence 0.5407. Its composite score of 66.86 exceeds standard MGDA by 6.41, the best fixed-weight baseline by 5.21, and the strongest additional baseline, DB-DPO (62.72), by 4.14 points. Mann-Whitney U tests confirm statistical significance for all improvements over standard MGDA ($p < 0.001$ for toxicity and virulence; $p = 0.003$ for pLDDT) and over the best fixed-weight baseline ($p < 0.001$ across all metrics).

(Q3c) Calibration dynamics differ between SAFEPRO and standard MGDA. Alongside the 66.86 vs. 60.45 composite-score gap in Table 2, the methods show different calibration dynamics. Across 1,000 PPO iterations, standard MGDA's regularized diagnostic $q^\star = (\mathbf{w}^*)^\top G^\star \mathbf{w}^*$ shows persistent fluctuations (median 6.3×10^5), while SAFEPRO reaches lower values late in training (median 9.3×10^2 over the final quarter). These diagnostics use different calibration spaces and do not establish a full-parameter convergence advantage or a common descent direction. Appendix L reports the trajectories and the per-calibration cosine gap $\Delta\rho_t = \rho(G_t) - \rho(G_t^{\mathcal{M}})$ underlying the switching rule.

Table 3: Sensitivity of SAFEPRO to the conflict-detection threshold λ. All runs use identical hyperparameters except λ. Standard MGDA included as the no-TCS reference.

λ	Mean pLDDT	P(toxic)	P(virulent)	Composite
MGDA	55.83	0.0523	0.6926	60.45
0.8	55.83	0.0523	0.6926	60.45
0.5	53.67	0.0477	0.6766	60.41
0.2	53.67	0.0477	0.6766	60.41
0.003	56.29	0.0189	0.6237	64.01
0.002	56.04	0.0139	0.5407	**66.86**
0.001	56.95	0.0421	0.5968	64.36
0	51.17	0.0072	0.5835	64.03

(Q3d) SAFEPRO reduces the harmful fraction among designable outputs across thresholds. For each policy we sample N=1,000 sequences and retain those whose self-consistency RMSD (scRMSD) between the original backbone and an ESMFold [Lin et al., 2023] inverse-folding round-trip falls below 2Å, following the same protocol described by Wagner et al. [2024]. Within this designable subset we report the fraction flagged as *harmful*, i.e. predicted toxic ($P_{\text{tox}}(x) > \tau$) or virulent ($P_{\text{vir}}(x) > \tau$) by CTAF as τ sweeps over $[0.5, 0.99]$.

Table 4: External evaluation on 1,000 sequences per policy. Classifier entries are mean predicted scores; VFDB entries are offensive-category hit counts per 1,000 sequences. Lower classifier scores and fewer hits indicate lower measured risk. Amino-acid composition entropy is an output diagnostic, not a safety score. Fixed denotes PPO with weights $(0.3, 0.3, 0.4)$.

Measure	Base	Fixed	MGDA	SAFEPRO
ToxDL2, $P(\text{toxic})$	0.053	0.027	0.031	0.025
ToxinPred2, $P(\text{toxic})$	0.370	0.391	0.372	0.369
VirulentHunter, $P(\text{virulent})$	0.523	0.567	0.539	0.467
VFDB offensive hits/1k	25	3	11	6
AA composition entropy (bits)	3.30	2.77	3.01	3.06

Figure 3 shows that SAFEPRO reduces the harmful fraction at every threshold: at τ=0.5, only 33.9% of its designable outputs are flagged as harmful, vs. 88.9% for fixed-weight PPO and 100% for the base model and standard MGDA. This safety gain is not purchased at the cost of structural quality: SAFEPRO produces 62 designable sequences, more than twice the 27 from the base model and fixed-weight baseline, and 27% more than standard MGDA (49). External evaluators not used for policy optimization show lower ToxDL2 and VirulentHunter scores and fewer VFDB matches than the base policy, while ToxinPred2 shows no significant change (Table 4). Composition entropy is an output diagnostic, not a safety measure. These computational results provide evidence beyond the training reward model without establishing biological safety. Screening details and statistical comparisons are in Appendix B.

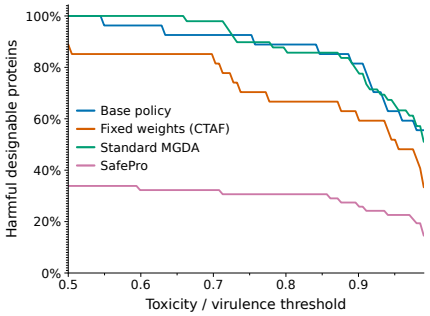

Figure 3: Fraction of designable proteins (scRMSD < 2 Å) flagged as harmful by CTAF (i.e., P_{tox} or P_{vir} above the threshold τ) as τ sweeps over $[0.5, 0.99]$. Lower is safer.

VFDB matches decrease across effector delivery, adherence, and exotoxin categories (Table 5). The largest absolute reduction occurs in effector delivery (15 to 3 hits per 1,000 sequences); these homology results do not establish changes in biological activity.

The alignment gains also transfer to ProtGPT2, where SAFEPRO achieves the highest mean pLDDT and lowest mean predicted toxicity and virulence among the compared policies (Table 6). Its composite score exceeds the best fixed-weight baseline and standard MGDA by 4.30 and 4.20 points, respectively. The full fixed-weight grid and evaluation details are in Appendix C.

Table 5: VFDB category-level homology hits per $1{,}000$ generated sequences. Each sequence is counted once according to its best qualifying hit. Change is relative to the base policy; it is undefined when the base count is zero. The category counts sum to the aggregate offensive-category counts in Table 4.

VFDB category	Base	SAFEPRO	Change
Effector delivery (T3SS/T4SS/T6SS)	15	3	-80%
Adherence (host interaction)	5	1	-80%
Exotoxin	5	2	-60%
Invasion	0	0	—

Table 6: ProtGPT2 results on $1{,}000$ sequences per policy. Entries report means; the full grid and standard deviations are in Table 8. Best fixed uses weights $(0.3, 0.3, 0.4)$ in toxicity, virulence, and pLDDT order.

Policy	pLDDT ↑	$P(\text{toxic})\downarrow$	$P(\text{virulent})\downarrow$	Composite ↑
Base	57.20	0.0472	0.6633	62.05
Best fixed	57.35	0.0487	0.6120	63.76
Standard MGDA	55.03	0.0609	0.5736	63.86
SAFEPRO	**58.59**	**0.0424**	**0.5017**	**68.06**

4.5 Sensitivity Analyses

(Q4a) Conflict detection threshold λ. Table 3 sweeps λ, which governs the switch from the full Gram matrix G to the conflict-focused $G^{\mathcal{M}}$. Large thresholds ($\lambda \in \{0.8, 0.5, 0.2\}$) rarely trigger conflict detection, reducing SAFEPRO to standard MGDA (composites near 60.4). Small but non-zero thresholds ($\lambda \in \{0.001, 0.002, 0.003\}$) engage the conflict-focused projection frequently enough to reshape updates; the peak at λ=0.002 (66.86) achieves the lowest $P(\text{toxic})$ and $P(\text{virulent})$ among the tested non-zero thresholds while preserving pLDDT. Setting λ=0 reduces pLDDT to 51.17, supporting a positive switching threshold. The tested small non-zero thresholds span a factor of three ($\lambda \in [10^{-3}, 3{\times}10^{-3}]$), and all outperform standard MGDA in composite score.

(Q4b) Length regularization r_{len}. We ablate r_{len} by replacing it with 0, holding all else fixed. With r_{len} enabled, the mean generated length $\bar{L}_t$ stays in $[490, 511]$ throughout training (Appendix Figure 4). Without it, $\bar{L}_t$ drifts from $\sim$480 at iteration 600 to $\sim$250 by iteration 700, decaying to a single-digit floor by iteration 900. This ablation identifies a short-sequence degeneration mode under the remaining rewards, which length regularization prevents in the tested setting.

5 Conclusion

We presented SAFEPRO, a multi-objective alignment framework that jointly optimizes structural foldability, toxicity avoidance, and virulence avoidance in protein language models. Its two novel components target concrete limitations of direct transfer from LLM alignment: (i) CTAF selectively exchanges cross-task information via gated cross-attention, replacing the choice between independent classifiers and hard parameter sharing. (ii) TCS-MGDA extends Pareto-descent weight calibration to the high-dimensional bounded-reward regime via task-conflict saliency filtering. On ProGen2-medium, SAFEPRO improves the composite score by $+5.21$ over the strongest fixed-weight baseline without requiring *a priori* weights. Among structurally designable proteins (scRMSD < 2 Å), SAFEPRO reduces the CTAF-flagged harmful fraction from 89% under fixed-weight PPO to 34% while producing more than twice as many designable sequences. Because the same machinery is inherently dual-use, we release models, data, and code on request rather than openly. Limitations, dual-use considerations, and this release policy are discussed in Appendix A.

References

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback, 2022. *URL https://arxiv. org/abs/2212.08073*, 2212, 2022.

Benjamin Buchfink, Klaus Reuter, and Hajk-Georg Drost. Sensitive protein alignments at tree-of-life scale using diamond. *Nature methods*, 18(4):366–368, 2021.

Manuel Fernández Burda, Santiago Aranguri, Iván Arcuschin Moreno, and Enzo Ferrante. Inference-time toxicity mitigation in protein language models. *CoRR*, abs/2603.04045, 2026. doi: 10.48550/ARXIV.2603.04045. URL https://doi.org/10.48550/arXiv.2603.04045.

Hanqun Cao, Marcelo Der Torossian Torres, Jingjie Zhang, Zijun Gao, Fang Wu, Chunbin Gu, Jure Leskovec, Yejin Choi, Cesar de la Fuente-Nunez, Guangyong Chen, and Pheng-Ann Heng. A deep reinforcement learning platform for antibiotic discovery. *CoRR*, abs/2509.18153, 2025. doi: 10.48550/ARXIV.2509.18153. URL https://doi.org/10.48550/arXiv.2509.18153.

Chen Chen, Yong Xu, Jian Ouyang, Xiangyi Xiong, Paweł P Łabaj, Agnieszka Chmielarczyk, Anna Różańska, Hao Zhang, Keyang Liu, Tieliu Shi, et al. Virulenthunter: deep learning-based virulence factor predictor illuminates pathogenicity in diverse microbial contexts. *Briefings in Bioinformatics*, 26(3):bbaf271, 2025.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, Proceedings of Machine Learning Research, pages 793–802. PMLR, 2018. URL http://proceedings.mlr.press/v80/chen18a.html.

Jean-Antoine Désidéri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus. Mathématique*, 350(5-6):313–318, 2012.

Ana Do Vale, Didier Cabanes, and Sandra Sousa. Bacterial toxins as pathogen weapons against phagocytes. *Frontiers in microbiology*, 7:42, 2016.

Noelia Ferruz, Steffen Schmidt, and Birte Höcker. Protgpt2 is a deep unsupervised language model for protein design. *Nature communications*, 13(1):4348, 2022.

William Karush. Minima of functions of several variables with inequalities as side constraints. *M. Sc. Dissertation. Dept. of Mathematics, Univ. of Chicago*, 1939.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv*, 2022:500902, 2022.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 18878–18890, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/9d27fdf2477ffbff837d73ef7ae23db9-Abstract.html.

Bo Liu, Dandan Zheng, Siyu Zhou, Lihong Chen, and Jian Yang. VFDB 2022: a general classification scheme for bacterial virulence factors. *Nucleic Acids Research*, 50(D1):D912–D917, 2022. doi: 10.1093/nar/gkab1107.

Xianliang Liu, Jiawei Luo, Xinyan Wang, Yang Zhang, and Junjie Chen. Directed evolution of antimicrobial peptides using multi-objective zeroth-order optimization. *Briefings Bioinform.*, 26 (1), 2025. doi: 10.1093/BIB/BBAE715. URL https://doi.org/10.1093/bib/bbae715.

Ali Madani, Ben Krause, Eric R Greene, Subu Subramanian, Benjamin P Mohr, James M Holton, Jose Luis Olmos Jr, Caiming Xiong, Zachary Z Sun, Richard Socher, et al. Large language models generate functional protein sequences across diverse families. *Nature biotechnology*, 41(8): 1099–1106, 2023.

Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. *arXiv preprint arXiv:2202.01017*, 2022.

Erik Nijkamp, Jeffrey A Ruffolo, Eli N Weinstein, Nikhil Naik, and Ali Madani. Progen2: exploring the boundaries of protein language models. *Cell systems*, 14(11):968–978, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Yingzhe Peng, Gongrui Zhang, Miaosen Zhang, Zhiyuan You, Jie Liu, Qipeng Zhu, Kai Yang, Xingzhong Xu, Xin Geng, and Xu Yang. Lmm-r1: Empowering 3b lmms with strong reasoning abilities through two-stage rule-based rl. *arXiv preprint arXiv:2503.07536*, 2025.

Johnny W Peterson. Bacterial pathogenesis. *Medical Microbiology. 4th edition*, 1996.

Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. Safety alignment should be made more than just a few tokens deep. In *The Thirteenth International Conference on Learning Representations*, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Ram Samudrala, Fred Heffron, and Jason E McDermott. Accurate prediction of secreted substrates and identification of a conserved putative secretion signal for type III secretion systems. *PLoS Pathogens*, 5(4):e1000375, 2009. doi: 10.1371/journal.ppat.1000375.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.

Neelam Sharma, Leimarembi Devi Naorem, Shipra Jain, and Gajendra PS Raghava. Toxinpred2: an improved method for predicting toxicity of proteins. *Briefings in bioinformatics*, 23(5):bbac174, 2022.

Fang Sheng, Mohammad Noaeen, and Zahra Shakeri. Prodcarl: Reinforcement learning-aligned diffusion models for de novo antimicrobial peptide design. *CoRR*, abs/2602.00157, 2026. doi: 10.48550/ARXIV.2602.00157. URL `https://doi.org/10.48550/arXiv.2602.00157`.

Emily H Soice, Rafael Rocha, Kimberlee Cordova, Michael Specter, and Kevin M Esvelt. Can large language models democratize access to dual-use biotechnology? *arXiv preprint arXiv:2306.03809*, 2023.

Trevor Standley, Amir Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *International conference on machine learning*, pages 9120–9132. PMLR, 2020.

The UniProt Consortium. UniProt: the universal protein knowledgebase in 2023. *Nucleic Acids Research*, 51(D1):D523–D531, 2023. doi: 10.1093/nar/gkac1052.

Fabio Urbina, Filippa Lentzos, Cédric Invernizzi, and Sean Ekins. Dual use of artificial-intelligence-powered drug discovery. *Nature machine intelligence*, 4(3):189–191, 2022.

Simon Wagner, Leif Seute, Vsevolod Viliuga, Nicolas Wolf, Frauke Gräter, and Jan Stühmer. Generating highly designable proteins with geometric algebra flow matching. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/8e906a69bcdfc3b9be15995b525ddda7-Abstract-Conference.html.

Li Wang, Xiangzheng Fu, Jiahao Yang, Xinyi Zhang, Xiucai Ye, Tetsuya Sakurai, Xiangxiang Zeng, and Yiping Liu. Moformer: navigating the antimicrobial peptide design space with pareto-based multi-objective transformer. *Briefings Bioinform.*, 26(6), 2025a. doi: 10.1093/BIB/BBAF376. URL https://doi.org/10.1093/bib/bbaf376.

Lijuan Wang, Chen Qiu, Liwei Xiao, and Junjie Chen. Peppfdpo: Multi-objective antimicrobial peptide generation using pareto-frontier enhanced direct preference optimization. In *IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2025, Wuhan, China, December 15-18, 2025*, pages 1–9. IEEE, 2025b. doi: 10.1109/BIBM66473.2025.11356223. URL https://doi.org/10.1109/BIBM66473.2025.11356223.

Yuhao Wang, Keyan Ding, Kehua Feng, Zeyuan Wang, Ming Qin, Xiaotong Li, Qiang Zhang, and Huajun Chen. Enhancing safe and controllable protein generation via knowledge preference optimization. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 12553–12569. Association for Computational Linguistics, 2025c. URL https://aclanthology.org/2025.acl-long.616/.

Yongliang Wu, Yizhou Zhou, Ziheng Zhou, Yingzhe Peng, Xinyu Ye, Xinting Hu, Wenbo Zhu, Lu Qi, Ming-Hsuan Yang, et al. On the generalization of sft: A reinforcement learning perspective with reward rectification. In *International Conference on Learning Representations*, volume 2026, pages 27550–27571, 2026.

Jingwei Yi, Rui Ye, Qisi Chen, Bin Zhu, Siheng Chen, Defu Lian, Guangzhong Sun, Xing Xie, and Fangzhao Wu. On the vulnerability of safety alignment in open-access llms. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9236–9260, 2024.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3fe78a8acf5fda99de95303940a2420c-Abstract.html.

Wenjun Zeng, Dana Kurniawan, Ryan Mullins, Yuchi Liu, Tamoghna Saha, Dirichi Ike-Njoku, Jindong Gu, Yiwen Song, Cai Xu, Jingjing Zhou, et al. Shieldgemma 2: Robust and tractable image content moderation. *arXiv preprint arXiv:2504.01081*, 2025.

Lin Zhu, Yi Fang, Shuting Liu, Hong-Bin Shen, Wesley De Neve, and Xiaoyong Pan. Toxdl 2.0: Protein toxicity prediction using a pretrained language model and graph neural networks. *Computational and Structural Biotechnology Journal*, 27:1538–1549, 2025.

A Limitations and Future Work

(i) Virulence remains the most resistant axis, reflecting a known difficulty: virulence is a multifactorial phenotype, and our reward derives from a sequence-level classifier rather than functional assays. The external VirulentHunter and VFDB homology evaluations provide additional evidence for the observed reduction, but experimental validation remains the definitive test. (ii) Because CTAF acts as both reward model and primary evaluator, the original evaluation may partly reflect reward-model consistency. We mitigate this concern with external toxicity and virulence predictors, a model-free homology screen, and an amino-acid entropy audit, none of which is used for policy optimization (Appendix B). These results reduce the risk of evaluation circularity, although they remain computational rather than biological validation.

Table 7: Reported FDR-adjusted p-values for SAFEPRO versus the base policy in Table 4; n.s. denotes a nonsignificant comparison.

Measure	Adj. p
ToxDL2, $P(\text{toxic})$	10^{-25}
ToxinPred2, $P(\text{toxic})$	0.61 (n.s.)
VirulentHunter, $P(\text{virulent})$	2×10^{-5}
VFDB offensive hits/1k	3×10^{-5}
AA composition entropy (bits)	0.74 (n.s.)

Dual-use considerations and responsible release. Our framework is designed to *reduce* the harmful output rate of protein language models. However, the same machinery is inherently dual-use: the reward signals and the multi-objective optimizer that steer generation away from predicted toxicity and virulence could in principle be inverted to steer generation toward them. The CTAF reward model is also, by construction, a sequence-level predictor of harmfulness, and an unrestricted release would lower the cost of filtering or screening candidates for undesirable properties. For this reason, we do not publish the trained reward-model checkpoints, the aligned policy weights, the curated toxicity and virulence training corpora, or the generated sequences in an openly downloadable form. Instead, we release them on request: researchers may contact the corresponding author by email. We view this gated-release arrangement as the appropriate balance between reproducibility and risk for work of this kind, and we welcome discussion of alternative access mechanisms with the community.

B External Safety Evaluation

Evaluation setup. We re-score the same $1{,}000$ generated sequences per policy used in Section 4.4, including sequences that do not pass the designability filter. The four policies are the base model, fixed-weight PPO with weights $(0.3, 0.3, 0.4)$, standard MGDA, and SAFEPRO. The external classifiers are ToxDL2 [Zhu et al., 2025], ToxinPred2 [Sharma et al., 2022], and VirulentHunter [Chen et al., 2025]. We additionally use homology screening and amino-acid composition entropy. These evaluation measures are not used to optimize the policies; this does not imply disjoint training data or pretrained representations relative to CTAF.

Homology screening and statistical comparisons. Homology searches use DIAMOND v2.2.4 [Buchfink et al., 2021] against Swiss-Prot toxin entries (KW-0800; $n = 7{,}916$) and VFDB set B ($n = 30{,}214$). Strong hits require at least 40% sequence identity and 50% query coverage. Table 4 reports the VFDB offensive-category hit counts alongside the external classifier scores and composition entropy. The reported comparisons with the base policy (Table 7) use two-sided Mann–Whitney U tests on per-sequence measurements with Benjamini–Hochberg false-discovery-rate correction. These tests characterize differences between generated-sequence distributions and do not quantify variability across independent training runs.

Results and interpretation. Relative to the base policy, SAFEPRO lowers the mean ToxDL2 score from 0.053 to 0.025, the mean VirulentHunter score from 0.523 to 0.467, and VFDB offensive-category hits from 25 to 6 per $1{,}000$ sequences. ToxinPred2 shows no significant change. Fixed-weight PPO has fewer VFDB hits than SAFEPRO (3 versus 6), but also lower composition entropy than the base policy (2.77 versus 3.30 bits; reported $p < 10^{-14}$). The latter observation complicates interpretation of its lower homology hit count without establishing that reduced entropy causes the difference. For SAFEPRO, the entropy comparison with the base policy is nonsignificant (3.06 versus 3.30 bits; adjusted $p = 0.74$); this does not establish equivalence. Composition entropy also does not measure sequence-order repetition or diversity across sequences.

Category-level virulence analysis. To complement the scalar virulence predictions, we break down the VFDB homology results by functional category. Each sequence is assigned to the category of its best qualifying hit under the screening criteria above and counted once. Table 5 shows fewer matches in effector delivery, adherence, and exotoxin categories, accounting for the aggregate reduction from 25 to 6 hits per $1{,}000$ sequences in Table 4. The largest absolute reduction occurs in effector delivery, from 15 to 3 hits per $1{,}000$ sequences. These results indicate reduced similarity to annotated virulence

Table 8: Alignment results on ProtGPT2. Weights are ordered as toxicity, virulence, and pLDDT. The first three metrics are means ± standard deviations across generated sequences. Bold values mark the best means and composite score among the compared policies. Δ is the composite-score change relative to the base policy.

Policy	pLDDT ↑	$P(\text{toxic})\downarrow$	$P(\text{virulent})\downarrow$	Composite ↑	Δ
Base (ProtGPT2)	57.20 ± 13.17	0.0472 ± 0.1644	0.6633 ± 0.3979	62.05	—
Fixed $(1, 0, 0)$	57.03 ± 13.18	0.0429 ± 0.1542	0.6586 ± 0.3988	62.29	$+0.24$
Fixed $(0, 1, 0)$	54.95 ± 14.66	0.0565 ± 0.1831	0.5950 ± 0.4014	63.27	$+1.22$
Fixed $(0, 0, 1)$	57.85 ± 13.09	0.0526 ± 0.1748	0.6516 ± 0.3985	62.48	$+0.43$
Fixed $(0.5, 0, 0.5)$	57.19 ± 13.44	0.0491 ± 0.1612	0.6478 ± 0.3998	62.50	$+0.45$
Fixed $(0, 0.5, 0.5)$	56.60 ± 12.96	0.0465 ± 0.1636	0.6337 ± 0.4022	62.86	$+0.81$
Fixed $(0.5, 0.5, 0)$	55.07 ± 14.76	0.0577 ± 0.1798	0.5809 ± 0.3961	63.74	$+1.69$
Fixed $(0.3, 0.3, 0.4)$	57.35 ± 13.64	0.0487 ± 0.1636	0.6120 ± 0.4057	63.76	$+1.71$
Standard MGDA	55.03 ± 14.96	0.0609 ± 0.1871	0.5736 ± 0.3998	63.86	$+1.81$
SAFEPRO	$\mathbf{58.59} \pm 15.31$	$\mathbf{0.0424} \pm 0.1365$	$\mathbf{0.5017} \pm 0.2696$	**68.06**	$\mathbf{+6.01}$

factors; context-dependent effects require functional assays. No qualifying invasion hits are detected under either policy, which does not establish the absence of invasion-related risk.

Agreement with CTAF. Across the $4{,}000$ sequences, CTAF virulence scores correlate with the VFDB homology signal (Spearman $\rho = 0.28$, $p = 1.7 \times 10^{-33}$). Sequences with a strong VFDB homolog have a median CTAF virulence score of 0.99 under each policy, including SAFEPRO. These observations support agreement within the subset detectable by the homology screen, but do not establish that every homolog is recognized or exclude reward-model blind spots elsewhere. Together, the external evaluations mitigate concerns that the gains are confined to CTAF scores; they do not establish biological safety or rule out all forms of reward hacking.

C Evaluation on a Second Backbone

We repeat the alignment protocol on ProtGPT2 [Ferruz et al., 2022], using its native BPE tokenizer while retaining the three objectives (toxicity, virulence, and pLDDT), the same CTAF reward model, and the PPO configuration described in Appendix J. We re-run all seven fixed-weight configurations and standard MGDA on this backbone, so every policy in Table 8 is evaluated within the same ProtGPT2 setting. Evaluation uses $1{,}000$ generated sequences per policy and the composite score defined in Section 4.

On ProtGPT2, SAFEPRO improves the composite score by 6.01 over the base policy, 4.30 over the best fixed-weight baseline, and 4.20 over standard MGDA. It achieves the highest mean pLDDT (58.59), lowest mean predicted toxicity (0.0424), and lowest mean predicted virulence (0.5017) among the compared policies. All three means also improve relative to the base model. These results support transfer of the alignment gains across the two evaluated backbones.

D Additional Baseline Implementations

Table 2 reports five additional baselines on ProGen2-medium. Each policy is evaluated on $1{,}000$ generated sequences using the same CTAF checkpoint, ESMFold evaluation, and composite score as the main comparison. Training procedures differ by method; a common evaluation protocol does not imply identical training objectives or computational costs.

Gradient-based methods. PCGrad [Yu et al., 2020], CAGrad [Liu et al., 2021], and Nash-MTL [Navon et al., 2022] combine the toxicity, virulence, and structural-quality objectives within the PPO setup. PCGrad projects away conflicting gradient components; CAGrad balances average-loss descent with worst-task local improvement; Nash-MTL obtains a joint update through a Nash bargaining formulation.

Inference-time steering. Logit Diff Amplification (LDA) [Burda et al., 2026] steers generation using the logit difference between the frozen base model and a toxicity-finetuned copy. Steering requires no further policy training once these checkpoints are available; preparing the toxicity-finetuned copy is a separate training step.

Database-paired preference optimization. DB-DPO follows the benign-over-harmful preference-learning rationale of KPO [Wang et al., 2025c], but does not reproduce its safety knowledge graph or graph-based pairing procedure. Instead, it uses preference pairs from public protein databases: chosen sequences are drawn from the non-toxic, non-virulent subset of the ToxDL split, and rejected sequences from toxic or virulent entries in ToxDL and VFDB. The policy is optimized with the DPO loss of Rafailov et al. [2023]. We report this database-paired baseline under its own name to distinguish it from a full KPO reproduction.

E Functional Category Assignment

Proteins were assigned to functional categories based on keyword matching against their UniProt annotations (Table 9).

Table 9: Functional category assignment keywords

Category	Keywords
Metabolism	metabolic process, biosynthetic, catabolic, glycolysis, TCA cycle, oxidative phosphorylation, amino acid biosynthesis, nucleotide biosynthesis, lipid biosynthesis, carbohydrate metabolic
Genetic information	DNA replication, DNA repair, transcription, translation, ribosom, tRNA, mRNA processing, chromosome, DNA binding
Cellular processes	cell division, cell cycle, cell wall, peptidoglycan, membrane, transport, protein folding, chaperone, proteasome
Signaling	signal transduction, kinase, phosphatase, two-component, sensor, regulator, response regulator
Stress response	heat shock, cold shock, oxidative stress, osmotic stress, acid stress, DNA damage response
Unknown	hypothetical, uncharacterized, unknown function, DUF, putative

F Length Regularizer Ablation Figure

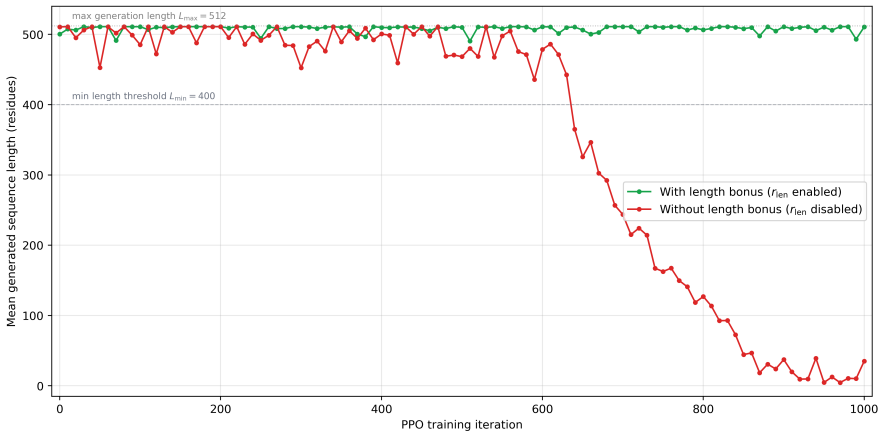

Figure 4: Effect of SAFEPRO's length regularizer r_{len} on mean generated sequence length during PPO training. Without r_{len}, the policy collapses to trivially short outputs that hack the reward model.

G Datasets

G.1 Toxicity Dataset

We adopt the protein toxicity benchmark released by Zhu et al. Zhu et al. [2025]. The dataset is derived from UniProt release 2024_03, with toxic proteins retrieved via the `KW-0800` keyword (reviewed entries only) and non-toxic controls retrieved with the complementary query that additionally excludes allergens (`NOT KW-0800 AND NOT KW-0020`). Sequences containing non-standard amino acids (`B, J, O, U, X, Z`) and exact duplicates are removed. Class imbalance in the negative pool is mitigated by random undersampling to a 1:2 toxic-to-non-toxic ratio, and near-duplicate sequences between splits are removed with CD-HIT-2D at a 40% sequence-identity threshold, which is the minimum supported by the tool and yields splits that do not share close homologues. We use three disjoint splits. The training set contains $4{,}900$ toxic and $9{,}800$ non-toxic sequences. The validation set comprises 78 toxic and 853 non-toxic sequences, with no sequence sharing more than 40% identity with any training sequence. The test set contains 112 toxic and $1{,}735$ non-toxic sequences, constructed under the same redundancy constraint. The deliberate imbalance in the validation and test splits reflects the realistic screening scenario: toxic proteins are rare in any proteome being screened, and a reward model must maintain high recall on the minority class under this skew. Because CD-HIT-2D is applied against the training set rather than within the evaluation splits, any performance our CTAF model reports on the validation and test sets is a measurement of generalization to sequences with at most 40% identity to anything seen during training, not merely of memorization of near-duplicates.

G.2 Virulence Prediction

Positive Dataset: VFDB Virulence Factors Positive training examples were obtained from the Virulence Factor Database (VFDB, `http://www.mgc.ac.cn/VFs/`), a repository of bacterial virulence-associated proteins Liu et al. [2022]. We downloaded the complete protein sequence dataset (VFDB_setB_pro.fas.gz), which contains the full collection of virulence-associated protein sequences, including both core and predicted virulence factors. Raw FASTA sequences were processed using a custom parsing pipeline to extract structured metadata from VFDB headers, including: VFG identifiers, gene names, protein functional annotations, virulence factor classifications, and source organism taxonomy. Sequences were deduplicated based on exact sequence identity, retaining entries with the most complete metadata annotations. The final positive dataset comprised 29,042 unique virulence-associated protein sequences spanning 878 bacterial species, with the highest representation from clinically significant pathogens including *Legionella pneumophila* ($n = 477$), *Pseudomonas aeruginosa* ($n = 400$), *Salmonella enterica serovar Typhimurium* ($n = 268$), and *Mycobacterium tuberculosis* ($n = 247$).

Negative Dataset: Non-Virulence Proteins Construction of a rigorous negative dataset is important for training robust classifiers that generalize beyond superficial sequence features Samudrala et al. [2009]. We sampled proteins across diverse functional categories. These included: metabolism, genetic information processing (DNA replication, transcription, translation), cellular processes (cell division, transport, protein folding), signal transduction, stress response, and proteins of unknown function. This diversity ensures the classifier learns features specifically associated with virulence rather than distinguishing functional categories. Complete reference proteomes were downloaded from UniProtThe UniProt Consortium [2023] for 21 phylogenetically diverse bacterial species with well-annotated reference strains. Gram-negative organisms included: *Escherichia coli* K-12 ($n = 4,076$), *Pseudomonas aeruginosa* PAO1 ($n = 4,989$), *Klebsiella pneumoniae* ($n = 4,837$), *Salmonella* Typhimurium LT2 ($n = 1,616$), *Vibrio cholerae* O1 ($n = 3,422$), *Yersinia pestis* CO92 ($n = 3,535$), *Helicobacter pylori* ($n = 1,372$), *Campylobacter jejuni* ($n = 1,426$), *Neisseria meningitidis* ($n = 1,857$), *Legionella pneumophila* ($n = 2,416$), *Bordetella pertussis* ($n = 3,047$), *Shigella flexneri* ($n = 1,546$), and *Proteus mirabilis* ($n = 3,318$). Gram-positive organisms included: *Staphylococcus aureus* NCTC8325 ($n = 2,720$), *Streptococcus pneumoniae* ($n = 1,946$), *Streptococcus pyogenes* ($n = 1,607$), *Enterococcus faecalis* ($n = 3,141$), *Listeria monocytogenes* ($n = 2,755$), *Bacillus subtilis* 168 ($n = 4,047$), *Clostridium difficile* ($n = 3,617$), and *Mycobacterium tuberculosis* ($n = 2,008$). Numbers in parentheses indicate proteins retained after virulence-associated filtering. A total of 59,298 candidate proteins were collected, reduced to 58,399 after sequence deduplication. Candidate negative proteins were filtered to remove any sequences with annotations suggestive of virulence association. Exclusion criteria included 40

keyword patterns encompassing: toxins and cytolysins (toxin, hemolysin, cytolysin, leukocidin, enterotoxin, exotoxin), adherence factors (adhesin, pilus, fimbriae, curli), invasion machinery (invasin, secretion system, T3SS, T4SS, T6SS, effector), immune evasion factors (capsule, LPS, complement evasion), motility and chemotaxis systems, biofilm components, quorum sensing regulators, iron acquisition systems (siderophore), and antibiotic resistance determinants. Additionally, candidate sequences identical to any VFDB entry were excluded using MD5 hash comparison to prevent information leakage. The complete list of keywords used to exclude potential virulence-associated proteins from the negative dataset: virulence, virulent, pathogen, pathogenic, toxin, toxic, hemolysin, cytolysin, leukocidin, adhesin, adhes, invasion, invasin, invade, secretion system, type III, type IV, type VI, T3SS, T4SS, T6SS, effector, translocat, pilus, pili, fimbr, curli, capsule, capsular, lipopolysaccharide, LPS, endotoxin, flagell, motility, chemotaxis, biofilm, quorum sens, autoinducer, siderophore, iron acqui, iron uptake, antibiotic resist, beta-lactam, carbapenem, immune evasion, complement, phagocyt, superantigen, enterotoxin, exotoxin, protease, collagenase, hyaluronidase. At this stage, sequences were retained only if they met the following criteria: length between 20 and 3,000 amino acids, and fewer than 5% ambiguous residue characters (X).

Dataset Balancing and Stratification To match biophysical property distributions between positive and negative datasets, we implemented proportional stratified sampling based on sequence length. To ensure compatibility with protein language models, sequences exceeding 1024 amino acids were excluded from both positive and negative datasets prior to balancing. This filter retained 93.3% of VFDB sequences (27,083 of 29,042) while removing very long multi-domain proteins that would require truncation during model training. The length distribution of the filtered VFDB sequences was divided into decile bins, and the proportion of sequences in each bin was calculated. Negative examples were then sampled from each length bin proportionally to match the VFDB distribution, with functional diversity maintained within each stratum. The final balanced dataset contained 27,083 positive (virulence-associated) and 27,083 negative (non-virulence) protein sequences, totaling 54,166 sequences. Length distributions were well-matched between classes (positive: 380.1 ± 209.2 residues; negative: 370.8 ± 197.9 residues). Negative sequences were distributed across functional categories: unassigned function ($n = 12,445$), cellular processes ($n = 4,655$), unknown/hypothetical proteins ($n = 3,576$), genetic information processing ($n = 3,483$), signal transduction ($n = 2,751$), stress response ($n = 106$), and metabolism ($n = 67$). Negative examples were sampled from 20 bacterial species.

H Computational Environment

All data processing was performed using Python 3.11 with the following packages: Biopython 1.81 for sequence parsing, pandas 2.0 for data manipulation, NumPy 1.24 for numerical operations, SciPy 1.11 for statistical testing, and requests 2.31 for API interactions. UniProt data was accessed via the UniProt REST API (`https://rest.uniprot.org/`).

I Proximal Policy Optimization with KL Penalty and Length Regularization

I.1 Length Regularization

Protein function depends on length, so SAFEPRO uses a two-regime reward r_{len}. For short sequences ($|x| < L_{\min} = 400$), $r_{\text{len}}(x) = -1 + |x|/L_{\min}$, linearly interpolating between -1 (empty) and 0 (threshold). For sequences at or above the reward threshold ($|x| \geq L_{\min}$), $r_{\text{len}}(x) = \min(|x|/\bar{L}, 1) \times 0.2$, where $\bar{L}$ is the running mean length (initialized to 512). The positive length bonus is capped at 0.2.

I.2 PPO Procedure

Reinforcement learning has proven effective for post-training large generative models, yielding gains in reasoning and generalization beyond what supervised fine-tuning alone achieves Peng et al. [2025], Wu et al. [2026]. We fine-tune the ProGen2 policy π_θ using Proximal Policy Optimization (PPO) Schulman et al. [2017], which alternates between collecting rollouts under the current policy and performing multiple gradient steps on the collected data while constraining how far the policy

moves per update. We augment the standard PPO objective with a token-level KL penalty against a frozen reference policy π_{ref} to prevent the policy from drifting into degenerate sequence distributions.

Rollout collection. At each iteration, the policy generates a batch of N sequences $\{x_n\}_{n=1}^{N}$ autoregressively from a fixed context c, without a hard minimum-length constraint. Short sequences are penalized only through r_{len} (Appendix I.1). Generation runs under `torch.no_grad()` using the rollout policy $\pi_{\theta_{\text{old}}}$, and the sampled sequences are held fixed for the subsequent optimization epochs. Before any parameter update, separate scoring forward passes through the policy and frozen reference compute the per-token log probabilities below. On calibration iterations, the policy scoring pass enables gradients for MGDA; reference scoring remains under `torch.no_grad()`:

$$\ell_t = \log \pi_\theta(x_t \mid x_{<t}), \qquad \ell_t^{\text{ref}} = \log \pi_{\text{ref}}(x_t \mid x_{<t}), \tag{10}$$

and the value estimate $V_\phi(x_{<t})$ for the prefix before generating x_t, from a separate value head V_ϕ attached to the corresponding policy hidden state. The fixed context c is implicit in all prefix notation.

Per-token reward with KL penalty. A token-level KL penalty discourages distributional drift from the reference but does not guarantee diversity or prevent all forms of reward hacking:

$$\tilde{r}_t = -\beta\,(\ell_t - \ell_t^{\text{ref}}), \tag{11}$$

where β is the KL coefficient. The biological reward $\mathcal{R}(x_n)$ from Eq. (1) is treated as a terminal signal, added only to the final token:

$$r_t = \begin{cases} \tilde{r}_t & t < T, \\ \tilde{r}_T + \mathcal{R}(x_n) & t = T. \end{cases} \tag{12}$$

This design reflects the episodic nature of protein generation: intermediate tokens receive only the KL shaping signal, while the full biological evaluation lands at sequence completion.

Advantage estimation via GAE. Advantages are estimated using Generalized Advantage Estimation (GAE) Schulman et al. [2015], which computes a weighted sum of TD errors backwards from the terminal step:

$$\delta_t = r_t + \gamma\,\mathbf{1}[t < T]\,V_\phi(x_{\le t}) - V_\phi(x_{<t}), \qquad \hat{A}_t = \sum_{l=0}^{T-t} (\gamma\lambda)^l\,\delta_{t+l}, \tag{13}$$

with $\gamma = 1.0$ and $\lambda = 0.95$, using value predictions evaluated at the rollout parameters. The bootstrap value at sequence completion is zero, so $\delta_T = r_T - V_\phi(x_{<T})$. The discount $\gamma = 1.0$ treats the full sequence as a single undiscounted episode. The GAE parameter λ controls the bias-variance trade-off: $\lambda < 1$ reduces variance at the cost of some bias by down-weighting distant TD errors. The GAE returns $G_t = \hat{A}_t + V_\phi(x_{<t})$ serve as regression targets for the value head. All computed quantities, log probabilities ℓ_t, values $V_\phi(x_{<t})$, advantages $\hat{A}_t$, and returns G_t, are stored in a rollout buffer and held fixed across PPO epochs.

Clipped surrogate objective. Given the stored rollout data, PPO performs E epochs of mini-batch gradient updates. At each step, a fresh forward pass through the *current* parameters θ computes new log probabilities ℓ_t^{new}. The probability ratio:

$$\rho_t = \frac{\pi_\theta(x_t \mid x_{<t})}{\pi_{\theta_{\text{old}}}(x_t \mid x_{<t})} = \exp\!\big(\ell_t^{\text{new}} - \ell_t\big) \tag{14}$$

measures how much the policy has changed since the rollout. The policy loss is the clipped surrogate:

$$\mathcal{L}^{\text{clip}}(\theta) = -\mathbb{E}_t\Big[\min\Big(\rho_t\,\hat{A}_t,\ \text{clip}(\rho_t,\,1-\varepsilon,\,1+\varepsilon)\,\hat{A}_t\Big)\Big], \tag{15}$$

with clip threshold $\varepsilon = 0.1$. The clip operates as a pessimistic bound: when $\hat{A}_t > 0$ (the action was good), increasing ρ_t beyond $1+\varepsilon$ provides no additional gradient, preventing overconfident exploitation of a single good sample; when $\hat{A}_t < 0$ (the action was bad), decreasing ρ_t below $1-\varepsilon$ provides no additional gradient, preventing catastrophic suppression. Advantages are normalized to zero mean and unit variance within each mini-batch before computing the objective to stabilize gradient magnitudes.

Value and entropy losses. The value head is trained with a clipped regression loss that mirrors the policy clipping, preventing the value function from updating too aggressively. Writing $v_t = V_\phi(x_{<t})$ for the current prediction and $v_t^{\text{old}} = V_{\phi_{\text{old}}}(x_{<t})$ for the stored rollout prediction:

$$\mathcal{L}^{\text{vf}}(\phi) = \mathbb{E}_t\Big[\max\Big(\big(v_t - G_t\big)^2,\ \big(\text{clip}(v_t, v_t^{\text{old}} - \varepsilon, v_t^{\text{old}} + \varepsilon) - G_t\big)^2\Big)\Big]. \tag{16}$$

An entropy bonus encourages exploration and prevents the policy from collapsing to a deterministic output prematurely:

$$\mathcal{L}^{\text{ent}}(\theta) = -\mathbb{E}_t\big[H\big(\pi_\theta(\cdot \mid x_{<t})\big)\big] = \mathbb{E}_t\left[\sum_v \pi_\theta(v \mid x_{<t}) \log \pi_\theta(v \mid x_{<t})\right]. \tag{17}$$

The combined training objective is:

$$\mathcal{L}(\theta, \phi) = \mathcal{L}^{\text{clip}}(\theta) + c_v\, \mathcal{L}^{\text{vf}}(\phi) + c_e\, \mathcal{L}^{\text{ent}}(\theta), \tag{18}$$

with value coefficient $c_v = 0.5$ and entropy coefficient $c_e = 0.01$. Gradients are clipped to a maximum norm of 1.0 before each optimizer step.

Reference model and precision. The reference model π_{ref} is the original ProGen2 checkpoint, kept fully frozen in FP16 on a dedicated GPU throughout training. It anchors the policy through the KL term in Eq. (11), discouraging distributional drift without guaranteeing the absence of reward hacking. The policy π_θ is maintained in FP32 to ensure that `.backward()` on calibration iterations produces accurate gradient vectors for MGDA, as described in Section 3.4.

J Implementation Details

J.1 Models

The policy is ProGen2-medium ($D = 764{,}803{,}616$ parameters) Nijkamp et al. [2023] maintained in FP32 throughout training. The frozen reference is the same checkpoint loaded in FP16 on a dedicated GPU. The CTAF reward model uses an ESM2-650M backbone Lin et al. [2022] with task-specific expert MLP modules ($d_{\text{exp}} = 1024$, head hidden dimension $d_{\text{head}} = 1440$) and $H_a = 4$ cross-task attention heads, trained separately on toxicity and virulence classification datasets. The pLDDT reward is computed by ESMFold Lin et al. [2023] in a cached scoring server.

J.2 Training Configuration.

CTAF experiments. All models, including our CTAF and the baselines, are trained under identical optimization settings to ensure a fair comparison. We employ a binary cross-entropy loss with logits (`BCEWithLogitsLoss`) using class-frequency-balanced positive weighting to mitigate class imbalance. Optimization is performed with AdamW using a weight decay of 0.01, a batch size of 64, and a maximum sequence length of $1{,}024$ residues. The learning rate follows a linear warmup schedule followed by cosine annealing, and gradients are clipped at an ℓ_2 norm of 1.0 to stabilize training. For each model, we perform a grid search over the number of training epochs in $\{10, 20, 30\}$ and the peak learning rate in $\{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$, selecting the configuration that maximizes the combined validation F1 score, $\frac{1}{2}(F1_{\text{tox}} + F1_{\text{vir}})$.

Alignment experiments. We use the same training settings for SAFEPRO and baselines. We use AdamW with grid search on policy learning rates from $[1 \times 10^{-6}, 5 \times 10^{-6}, 1 \times 10^{-7}, 5 \times 10^{-7}]$ and value-head learning rates from $[1 \times 10^{-6}, 5 \times 10^{-6}, 1 \times 10^{-7}, 5 \times 10^{-7}]$, weight decay 0.01, with cosine annealing after a 5% linear warmup. PPO uses clip $\epsilon = 0.1$, value coefficient 0.5, entropy coefficient 0.01, KL coefficient $\beta = 0.3$, and GAE with $\lambda = 0.95$, $\gamma = 1.0$. Each rollout batch contains $N = 8$ sequences of up to 512 tokens, with length-reward threshold $L_{\min} = 400$. MGDA calibration runs every $\tau = 10$ iterations with EMA $\alpha = 0.95$, Gram regularization $\epsilon_G = 10^{-6}$, TCS top fraction $\xi = 0.05$, and TCS minimum parameters $M_{\min} = 1000$.

Weight post-processing. For both TCS-MGDA and standard MGDA, we first solve the sub-problem on the ordinary simplex, then apply EMA, $\tilde{\mathbf{w}}_t = \alpha\, \mathbf{w}^* + (1 - \alpha)\, \mathbf{w}_{t-1}$, with $\alpha = 0.95$. The weight floor of 0.05 is applied only after EMA as post-processing to obtain $\mathbf{w}_t$; it is not a constraint in the MGDA sub-problem. Rewards and GAE advantages are recomputed using these post-processed weights.

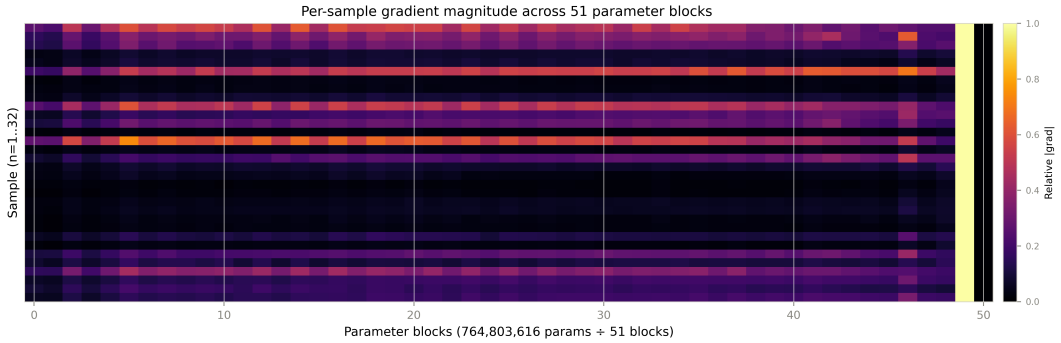

Figure 5: Raw per-sample log-probability gradients $|\nabla_n|$ across 51 parameter blocks for N=32 sequences generated by the `progen2-medium` policy. **Each row is independently normalized by that sample's own maximum block magnitude**, i.e. entry (n, b) plots $\|\nabla_n^{(b)}\|_2 \,/\, \max_{b'} \|\nabla_n^{(b')}\|_2 \in [0, 1]$. Per-row normalization removes the large magnitude differences *across* samples so that the *within-sample* distribution of gradient signal is directly comparable: bright = blocks carrying most of that sample's gradient, dark ≈ zero for that sample.

Computational overhead. On calibration iterations, MGDA adds one `.backward()` call per rollout sample using the gradient-enabled policy scoring forward pass, not the no-gradient generation pass, plus the $N \times N$ gradient inner-product matrix ($N = 8$, negligible) and the linear-time TCS partial sort over D parameters. The policy is kept in FP32 on the calibration pass specifically to ensure that `.backward()` lands on the true parameter `.grad` buffers rather than scaled mixed-precision copies; the reference model remains in FP16 throughout. Total MGDA overhead is under 5% of per-iteration wall time. All experiments are conducted on three NVIDIA RTX A6000 GPUs with 48 GB of memory each. Training the CTAF reward model requires approximately 9 GPU-hours, while the full PPO alignment stage requires approximately 170 GPU-hours.

K Per-Sample Gradient Sparsity

The mean-aggregated view in Figure 2 of the main body shows that average gradient mass concentrates on a single output-projection block. Here we complement that view with the raw per-sample picture, which clarifies *why* the mean alone is dominated by one block and rules out the simplest possible workaround, namely pruning a fixed inactive subset of parameters offline.

Per-sample gradients are sparse and sample-specific. Figure 5 shows the per-sample gradient magnitude across parameter blocks, row-normalized so that bright regions indicate where each sample's gradient concentrates. Two patterns stand out. First, within any single sample signal is confined to a small number of blocks while the remainder is essentially dark. Second, the bright columns are not aligned across rows: samples n=5, n=9, and n=13 distribute gradient broadly across the backbone, while n=3, n=7, n=16–25 activate only the final output-projection block (the tall yellow stripe near block 49) and leave the backbone essentially untouched. The active parameters are therefore *sequence-dependent*, not a fixed dead set that could be pruned offline. These observations motivate sample-dependent parameter selection for weight calibration; they do not by themselves establish inter-objective conflict.

L MGDA Dynamics and the Cosine Gap

This section evaluates the optimization behavior of TCS-MGDA along two diagnostics: (i) the regularized quadratic objective $q^\star$, related to the squared gradient norm by Eq. (9); and (ii) the per-calibration cosine-similarity gap $\Delta\rho_t = \rho(G_t) - \rho(G_t^{\mathcal{M}})$ between the full-parameter Gram matrix G_t and its TCS-restricted counterpart $G_t^{\mathcal{M}}$. We compare SAFEPRO against an MGDA baseline that shares the PPO loop, reward stack, and calibration cadence (τ=10 over 1,000 PPO iterations) but solves the MGDA sub-problem on the full Gram matrix G [Désidéri, 2012, Sener and Koltun, 2018].

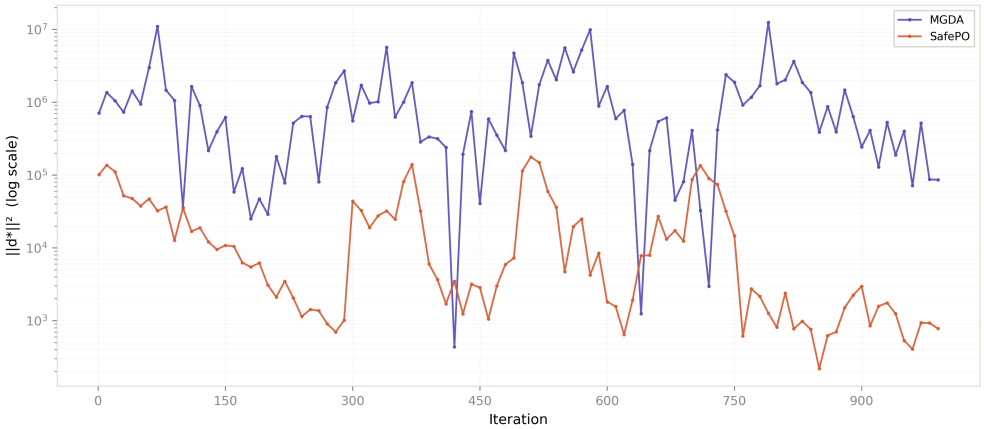

Figure 6: Regularized quadratic objective $q^\star = (\mathbf{w}^*)^\top G^\star \mathbf{w}^*$ on a logarithmic scale. The plot retains the original $\|\mathbf{d}^\star\|^2$ axis label; the exact correction is given in Eq. (9). The MGDA baseline (purple) fluctuates over several orders of magnitude. SAFEPRO (orange) shows an overall early decline, larger mid-training fluctuations, and lower values in the final quarter.

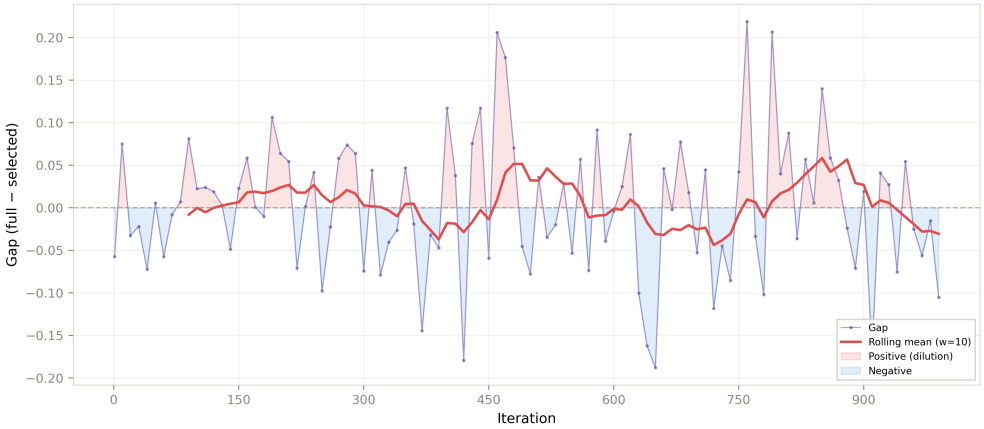

Figure 7: Cosine-similarity gap diagnostic $\Delta\rho_t = \rho(G_t) - \rho(G_t^{\mathcal{M}})$ per calibration, with a 10-calibration rolling mean (red).

Regularized quadratic objective $q^\star$. Figure 6 reports $q^\star = (\mathbf{w}^*)^\top G^\star \mathbf{w}^*$ on a logarithmic scale. This quantity includes the diagonal-regularization contribution $\epsilon_G \|\mathbf{w}^*\|_2^2$, which lies in $[\epsilon_G/K, \epsilon_G]$ for simplex weights. With $\epsilon_G = 10^{-6}$, the distinction does not affect the reported numerical precision, but the two quantities are not mathematically identical. SAFEPRO shows an overall decrease from approximately 10^5 to 10^3 over iterations 1–290, interrupted by local increases; larger fluctuations across iterations 300–750; and lower values thereafter, with median 9.3×10^2 over the final 25 % of training. The MGDA baseline shows persistent fluctuations over several orders of magnitude, with median 6.3×10^5. These trajectories describe the behavior of the respective calibration objectives, rather than establish convergence in the full parameter space.

Cosine gap. Figure 7 reports the per-calibration gap $\Delta\rho_t$ together with a 10-calibration rolling mean. Across the 101 calibrations, 52 % of gaps are positive, indicating higher average regularized cosine similarity in the full Gram matrix than in the selected subset at those steps. This diagnostic motivates the switching rule, but does not establish lower variance of the restricted estimator or explain the performance gap causally. Because the statistic uses regularized Gram matrices, it also depends on gradient scale relative to ϵ_G, not solely on raw-gradient angles.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction (Section 1) state four contributions—the SAFE-PRO framework, the CTAF reward model, the TCS-MGDA weight calibration method, and the released virulence dataset—each of which is substantiated by the corresponding experimental sections (Sections 4.2–4.4) and supported by the reported numerical improvements ($+5.21$ composite gain, harmful-fraction reduction from 89% to 34%). The scope (single backbone, sequence-level reward modeling) and limitations are explicitly acknowledged in Section 5 and Section A.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section A explicitly enumerates three limitations: (i) validation on a single backbone (ProGen2-medium) without empirical evidence of multi-backbone generalization, (ii) the difficulty of modeling virulence as a multi-factorial phenotype from a sequence-level classifier rather than functional assays, and (iii) the dual role of CTAF as both reward model and evaluator, which means reported safety gains partly reflect reward-model consistency rather than ground-truth biological validation.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not present formal theorems or theoretical results requiring proofs. Its contributions are methodological (CTAF reward model, TCS-MGDA calibration) and empirical, with the underlying MGDA framework [Désidéri, 2012, Sener and Koltun, 2018] cited from prior work.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper specifies the base policy (ProGen2-medium), reward backbone (ESM2-650M), and structural evaluator (ESMFold) in Appendix J, along with all CTAF architectural details (Section 3.2), the full TCS-MGDA procedure with pseudocode (Algorithm 1), and the complete PPO formulation (Appendix I). Hyperparameters, optimizer settings, grid-search ranges, dataset construction (including VFDB processing, keyword filters, and length-stratified balancing in Appendix G), and the composite evaluation protocol (Section 4) are also reported.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We release anonymized code for CTAF, TCS-MGDA, and the KL-regularized PPO training loop, together with the curated $54{,}166$-sequence virulence dataset, in the supplemental material. Appendix G documents dataset construction in full (VFDB processing, exclusion keywords, length-stratified balancing, MD5-based leakage removal), and Appendix J specifies all hyperparameters, optimizer settings, grid-search ranges, and the computational environment needed to reproduce the main experimental results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Data splits are reported in Section 4.1 and Appendix G (toxicity train/val/test sizes with CD-HIT-2D at 40% identity; balanced virulence dataset construction). Appendix J specifies optimizers (AdamW), learning-rate grids ($\{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$ for CTAF;

$\{1\times10^{-6}, 5\times10^{-6}, 1\times10^{-7}, 5\times10^{-7}\}$ for the policy and value head), schedulers (cosine annealing with 5% warmup), batch sizes, model-selection criteria, and all PPO/MGDA hyperparameters ($\beta = 0.3$, $\varepsilon = 0.1$, $\gamma = 1.0$, $\lambda = 0.95$, $\tau = 10$, $\alpha = 0.95$, $\xi = 0.05$, $M_{\min} = 1000$).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Table 2 reports per-sequence standard deviations over $N = 1{,}000$ generated sequences for pLDDT, $P(\text{toxic})$, and $P(\text{virulent})$. Statistical significance is established via Wilcoxon signed-rank tests for CTAF vs. baseline reward models (Section 4.2, all toxicity comparisons $p < 0.001$) and Mann-Whitney U tests for SAFEPRO vs. standard MGDA and the strongest fixed-weight baseline (Section 4.4, $p < 0.001$ on toxicity and virulence; $p = 0.003$ on pLDDT vs. standard MGDA), chosen because the underlying score distributions are non-Gaussian and bounded in $[0, 1]$.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Appendix J reports the hardware configuration (three NVIDIA RTX A6000 GPUs, 48 GB memory each) and approximate wall-clock cost for the main experiments. The same appendix also discusses the per-iteration MGDA overhead (under 5% of wall time), and Section 3.4 quantifies the memory savings from the Gram-matrix factorization ($\mathcal{O}(N^2)$ vs. $\mathcal{O}(KD)$ $\sim$1.2 GB).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics. The work is explicitly motivated by responsible deployment of protein language models: it proposes a safety-alignment framework that reduces toxicity and virulence in generated sequences, uses only publicly available datasets (UniProt, VFDB, bacterial reference proteomes), and discusses dual-use concerns and limitations transparently in Sections 1 and 5.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Section 1 frames the work around the dual-use concern of protein language models and motivates SAFEPRO as an upstream alignment layer that complements downstream sequence screening. The positive impact (reducing toxic and virulent generations while preserving structural quality, with empirical gains demonstrated in Section 4.4) is discussed alongside limitations on virulence reduction and the gap between reward-model consistency and ground-truth biological validation (Section 5 and Section A).

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: The released artifacts pose limited additional misuse risk: (i) the virulence dataset is curated entirely from already-public sources (VFDB [Liu et al., 2022] and UniProt reference proteomes), with no novel sequences introduced; (ii) the CTAF reward model is a sequence-level *classifier* of toxicity and virulence intended to support safety screening, not a generator of harmful sequences; and (iii) the alignment framework itself is designed to *reduce* the rate of harmful outputs from PLMs, with empirical evidence in Section 4.4 showing the harmful fraction among designable sequences drops from 89% to 34%. We do not release any aligned model weights that could be reverse-aligned, and the trained policy (π_θ) is not part of the supplemental release.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All existing assets are credited via citation: ProGen2 [Nijkamp et al., 2023], ESM2 [Lin et al., 2022], and ESMFold [Lin et al., 2023] for models; the toxicity benchmark of Zhu et al. [Zhu et al., 2025], VFDB [Liu et al., 2022], and UniProt [The UniProt Consortium, 2023] for data sources. ProGen2, ESM2 and ESMFold, VFDB, and UniProt are used in accordance with their respective licenses, and dataset versions and access endpoints are documented in Appendix G.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper introduces two new assets: the curated $54{,}166$-sequence virulence dataset and the SAFEPRO codebase (CTAF, TCS-MGDA, KL-regularized PPO), both released anonymously in the supplemental material. Appendix G documents the dataset's construction pipeline, source organisms, length distributions, functional category breakdown, exclusion keywords, and leakage-removal procedure, while Appendix J and Section 3 document the model architectures and training procedures.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core methods of this research do not involve general-purpose LLMs as an important or non-standard component. The paper studies *protein* language models (ProGen2 and ESM2), which are the subject of study rather than tools used in method development; no LLMs were used in a way that affects the core methodology, scientific rigor, or originality of the research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.