Enhancing AlphaFold3 for Protein-Ligand Co-Folding via Reinforcement Learning

Yu Pei [1] **Yuxuan Song** [1] **Zhilong Zhang** [1] **Keyue Qiu** [1] **Hao Zhou** [1] **Wei-Ying Ma** [1]

Abstract

Protein–ligand co-folding has emerged as a powerful alternative for modeling protein-ligand complex, offering inherent flexibility and removing reliance on experimentally determined crystal structures. Recent AlphaFold3-style conditional diffusion models achieve state-of-the-art accuracy on docking benchmarks but lack mechanisms to encode physical principles and expert experience. We propose to utilize Kahneman–Tversky Optimization (KTO), a reinforcement learning method that directly integrates human and biochemical preference signals, for conditional diffusion–based co-folding models. AF3-KTO seamlessly aligns with binary docking feedback and the iterative, conditional architecture of AlphaFold3-style models, eliminating the need for a separate reward network and minimizing computational overhead. Extensive evaluations on multiple benchmarks show that KTO consistently enhances binding-pose accuracy and physical plausibility, even under imbalanced preference data.

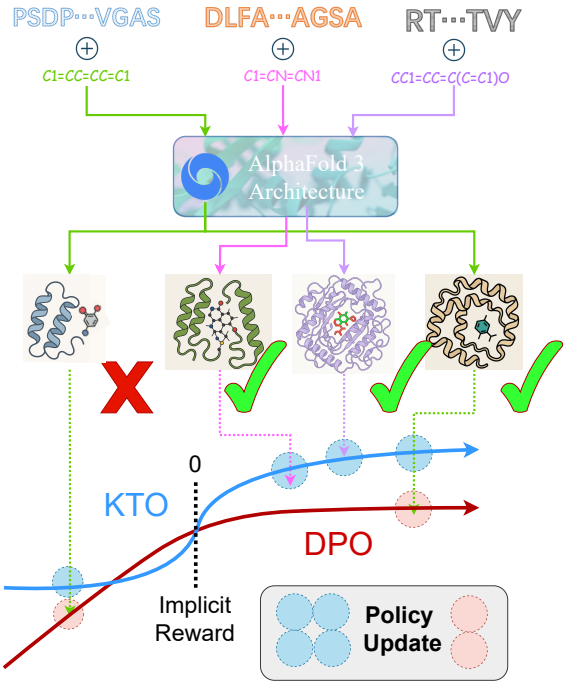

Figure 1. An overview of AF3-KTO's preference optimization pipeline. KTO fully make use of generated data in co-folding scenario.

1. Introduction

Protein–ligand co-folding is a specialized form of structure prediction that simultaneously models the three-dimensional structure of a receptor protein at the binding pose and conformation of one or more ligands, which requires the understanding of biomolecular interactions, and holds the potential of accelerating the discovery of novel therapeutics. Previous methods have approached the structure prediction problem as protein-ligand docking, yet the advanced formulation of co-folding enjoys twofold advantages over the previous one: (i) it naturally accounts for the inherent flexibility and induced-fit effects of protein–ligand complexes by generating receptor and ligand coordinates in a single, integrated process rather than relying on time-consuming iterative adjustments, which are usually conducted by traditional physics-based docking methods (Trott & Olson, 2010); and (ii) it is broadly applicable to high-throughput wet-lab experiments, since it does not depend on experimentally determined apo structures as required by deep learning-based docking models such as DiffDock (Corso et al., 2022), which may be unavailable or undiscovered.

A promising avenue to conduct AI-based protein-ligand co-folding is leveraging AlphaFold3-like pre-trained models with multiple capabilities folding bio-molecular complexes incorporating interactions with proteins, nucleic acids, and

[1]Institute for AI Industry Research (AIR), Tsinghua University. Correspondence to: Hao Zhou <zhouhao@air.tsinghua.edu.cn>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

small molecules. During this process, AlphaFold3-style architectures leverage extensive multiple sequence alignments and structural templates, rather than explicit apo structures, to achieve state-of-the-art results (Liu et al., 2024; Wohlwend et al., 2024; Team et al., 2025). These emergent capabilities stem from integrating diverse biological constraints during pre-training, novel network designs that fuse multiple representations, and iterative recycling through large-scale conditional diffusion modules. Despite these breakthroughs, AlphaFold3-like models inherit the limitations of one-shot deep learning approaches, including violation of physical principles and the absence of experiences from human expert which hinders their adoption in wet-lab settings.

In this work, we address these challenges by introducing a reinforcement-learning (RL) framework that embeds biochemical preferences as reward signals, encouraging model's generalization beyond the data distribution and promoting more physically feasible and explainable samples. Building on direct preference optimization techniques, we propose to utilize Kahneman–Tversky Optimization (KTO) in AlphaFold3 architecture, an RL method which we found tailored to protein–ligand co-folding and conditional diffusion modules. Unlike prior approaches, KTO eliminates the need for a separately trained reward model and constructs win-loss comparisons under identical conditions to minimize extra computation (Ethayarajh et al., 2024). Furthermore, by weighting training examples according to prospect theory (Kahneman & Tversky, 2013), KTO mitigates abrupt distributional shifts caused by extreme outliers while preserving the core strengths of AlphaFold3-like models.

2. Preliminaries

2.1. Diffusion Models

Diffusion models (Ho et al., 2020; Rombach et al., 2022; Esser et al., 2024; Karras et al., 2022) are trained by simulating a forward–reverse noisy Markovian perturbation process (Garcia & Rachelson, 2013). The forward process $q(x_t \mid x_{t-1})$ transits data x_0 to noisy sample x_T, while reverse process $p_\theta(x_{t-1} \mid x_t)$ reconstruct noisy to generated samples. Common training objective could be factorized as:

$$L_{\mathbf{Diff}} = E_{x_0,\epsilon,t,x_t}[w(\lambda_t)||\epsilon - \epsilon_\theta(x_t, t)||_2^2] \quad (1)$$

where θ is the main parameters of networks, $w(\lambda_t)$ is a weight function over signal noise ratio.

2.2. Kahneman Tversky Optimization for Diffusion

Originally, KTO (Ethayarajh et al., 2024) is designed for language modeling (Achiam et al., 2023)(Guo et al., 2025) distribution $\pi_\theta(x \mid y)$ of sentence x given prompt y with loss function written as:

$$L_{\mathbf{KTO}} = -E_{c,x\in D}[U(w(x)(\beta \log \frac{\pi_\theta(x|y)}{\pi_{ref}(x|y)}) - Q_{ref})] \quad (2)$$

where D for a binary Dataset with desirable or undesirable samples, β for scaling factor, $w(x) \in \{+1, -1\}$ for preference signal, Q_{ref} for average implicit reward, $U(v)$ for weight function. As shown in Figure.1, the main difference between KTO and DPO are a sigmoid-like weight function (which eliminated huge policy update when implicit reward reach $-\infty$) and the separation of win-loss pairs.

Meanwhile, KTO can be easily applied to diffusion models (Li et al., 2024) using an estimated upper bound:

$$L_{\mathbf{Diff-KTO}} = -E_{c,x_0\in D,t\in[0,T]}[U(w(x_0) (\beta \log \frac{\pi_\theta(x_{t-1}|x_t)}{\pi_{ref}(x_{t-1}|x_t)}) - Q_{ref})] \quad (3)$$

where π_θ stands for model under optimization, π_{ref} for fixed pre-trained model. Q_{ref} can be calculated according to construction of "mismatched pairs" over a small batch of data $\{x^0, x^1, x^2 ... x^m\}$ where $j = (i+1) \mod m$.

$$\{(x_t^i, x_{t-1}^j)\} = \{(x_t^1, x_{t-1}^2)..(x_t^m, x_{t-1}^0)\} \quad (4)$$

$$Q_{ref} = \beta \max(0, \frac{1}{m}\sum_{i=1}^{m-1} \log \frac{\pi_\theta(x_{t-1}^j|x_t^i)}{\pi_{ref}(x_{t-1}^j|x_t^i)}) \quad (5)$$

3. Methodology: KTO for AlphaFold3

In this section, we aims to transfer the algorithm of KTO to the core conditional diffusion module of AlphaFold3 architecture and enable optimization for protein ligand co-folding task. Detailed derivation could be found in Appendix B.

For structure prediction model like AlphaFold3, we're modeling protein structure x (usually continuous atom coordinates) given protein sequence y (discrete amino acid type). Specifically as an unified framework conducting protein ligand co-folding, AlphaFold3 is predicting complex structure x given sequence of protein y_p and smiles of small molecular ligand y_m. However, input information of the whole structure prediction model consists far more than that, including Multiple Sequence Alignment (MSA) searched from homologous database y_M, as well as template searched from structure database y_T. For clarity, all input conditions are allocated in formula of y:

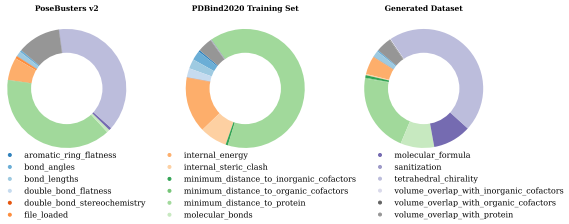

Figure 2. Distribution of invalid reasons for different dataset

$$y = \{y_p, y_m, y_M, y_T, ...\} \tag{6}$$

Since KTO has already implied into conditional task with textual prompt for example text-to-image generation, all things we're going to do are applying those conditions to Diffusion-KTO algorithm for optimization on AlphaFold3:

$$L_{\mathbf{AF3-KTO}} = -E_{c,(x_0,y)\in D,t\in[0,T]}[U(w(x_0) (\beta \log \frac{\pi_\theta(x_{t-1}|x_t,y)}{\pi_{ref}(x_{t-1}|x_t,y)}) - Q_{ref})] \tag{7}$$

Correspondingly, conditions are also added when getting the term of Q_{ref} with batch of data $\{([x^0 \mid y^0]...[x^m \mid y^m])\}$:

$$\{([x_t^i \mid y^i], [x_{t-1}^j \mid y^j])\} = \{([x_t^1 \mid y^1], [x_{t-1}^2 \mid y^2]) ...([x_t^m \mid y^m], [x_{t-1}^0 \mid y^0])\}, \quad j = (i+1) \mod m \tag{8}$$

$$Q_{ref} = \beta \max(0, \frac{1}{m}\sum_{i=1}^{m-1} \log \frac{\pi_\theta(x_{t-1}^j|x_t^i, y^i)}{\pi_{ref}(x_{t-1}^j|x_t^i, y^i)}) \tag{9}$$

Since then, we've applied KTO for AlphaFold3 architecture theoretically. In addition, we find that AF3-KTO can directly optimized using Diffusion MSE loss as part of loss function with some algebra. For detailed implement algorithm with practical setting considered, please refer to Appendix B.2.

4. Experiments

4.1. Dataset Construction

To discover the strengths and limitations of our AlphaFold3-like model in predicting protein–ligand interactions, we constructed a synthetic preference dataset derived from PDBBind2020 (Wang et al., 2004; Liu et al., 2017), which comprises 19,443 protein–ligand complexes. Figure 2 highlights a distribution shift between standard and predictive complexes, underscoring the need for a tailored dataset.

Our analysis of the primary PB-invalid failure modes reveals two key shortcomings of existing datasets:

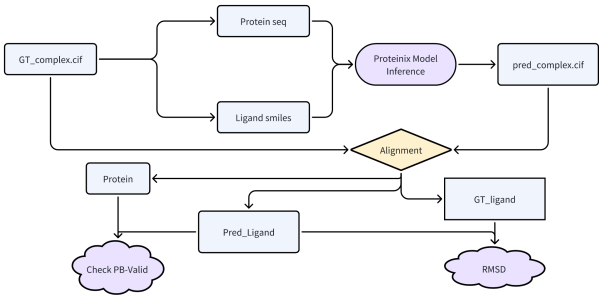

Figure 3. Evaluation pipeline of AF3-KTO

Overlap with Pre-training Data: AlphaFold3 was pre-trained on the entire PDB (Sidi & Keasar, 2020) and multiple distilled resources (AlphaFold2, MGnify (Richardson et al., 2023), OpenProteinSet (Ahdritz et al., 2023), UniClust30 (Mirdita et al., 2017), etc.), which include most proteins and many common ligands. Consequently, directly using PDBBind for preference learning suffers from redundancy and limited novelty.

Unmatched Error Modes: Ground-truth complexes do not capture prediction-specific errors (e.g., incorrect chirality or altered secondary structure). Moreover, generating negative examples from model outputs shifts the dominant failure modes in training samples.

As we apply the KTO optimization framework to this dataset, all samples from generated dataset are used fully, labeling as positive any complex with RMSD<2Å or deemed PB-valid, and negative otherwise. Details of the KTO hyperparameters are given in Appendix C.

4.2. Evaluation Pipeline

Following AlphaFold3 its reproducing works, we evaluate protein–ligand co-folding on the PoseBuster v2 dataset (Buttenschoen et al., 2024), which contains complexes resolved in 2021 or later to avoid overlap with our training set. Two complementary metrics quantify model performance:

RMSD Success rate: Predicted and reference complexes are first superimposed by aligning protein backbones with TM-align (Zhang & Skolnick, 2005). The ligand RMSD is then computed, and predictions with RMSD<2Å are counted as successful.

PB-Valid: Each predicted complex must pass a suite of 19 physics-based checks (bond lengths and angles, ligand strain energy, chirality, steric clashes, etc.). Only complexes that satisfy all checks are considered valid.

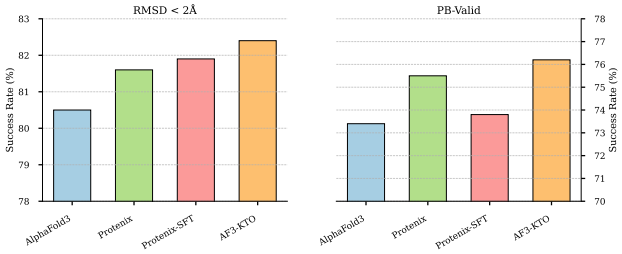

Figure 4. Protein ligand co-folding results AF3-KTO compared with baselines

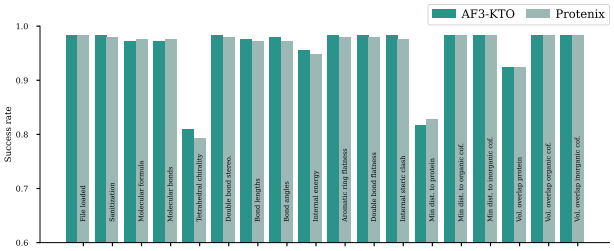

Figure 5. Detailed validity check between AF3-KTO and Protenix

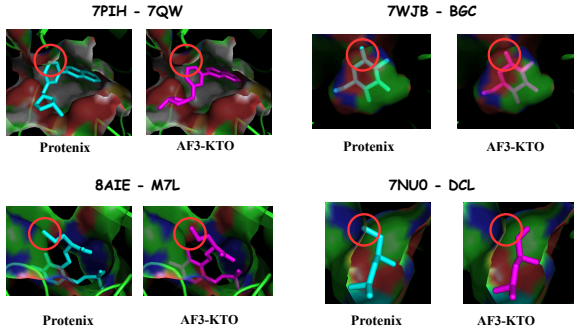

Figure 6. AF3-KTO solves structural invalid issues

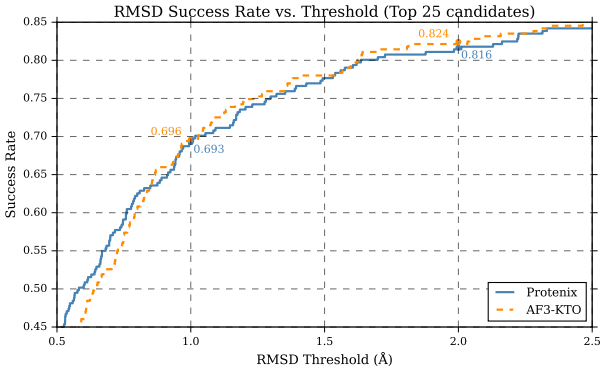

Figure 7. RMSD success rate curve with threshold

4.3. Main Results

Figure 4, 5 and 7 compares AF3-KTO preference optimized model against the baseline AlphaFold-like pre-training models. Across both RMSD success rate and PB-Valid metrics, our method achieves substantially higher scores, demonstrating that Kahneman Tversky Optimization markedly enhances predictive accuracy and physical plausibility. Qualitative examples in Figure 6 further illustrate how KTO corrects common geometric and physical errors.

4.4. Ablation Study

To isolate the contributions of KTO, we performed a controlled ablation in which the Protenix model was fine-tuned using only positive examples (RMSD<2Å and PB-Valid), under identical training steps. Table 1 reports results for varying numbers of generated candidates. In every configuration, AF3-KTO outperforms Protein-SFT, confirming that KTO-driven preference learning yields superior co-folding performance. Notably, the decline in PB-Valid rate at limited candidate counts suggests a trade-off between candidate volume and physical reliability.

Table 1. Ablation study on SFT data and number of candidates

Model		Protenix	SFT	**Ours**
RMSD<2Å	Top1	71.7%	72.1%	**73.9%**
	Top5	77.9%	78.0%	**79.0%**
	Top25	81.6%	81.9%	**82.4%**
PB-Valid	Top1	38.3%	34.1%	**38.6%**
	Top5	61.0%	61.4%	**62.1%**
	Top25	75.5%	73.8%	**76.2%**

5. Conclusion

In this paper, we proposed a novel approach of conducting advanced preference optimization method KTO on the framework of AlphaFold3. Based on pre-trained model, we made a step forward to improve generative qualities with human preference, which is an Out of Distribution (OOD) ability captured by reinforcement learning. In addition, AF3-KTO has clear implement theoretically and huge generalization potential which could perform on any labeled dataset with any conditions of protein-ligand pairs.

Impact Statement

Our work introduces Kahneman–Tversky Optimization (KTO), a reinforcement-learning framework tailored for protein–ligand co-folding. AF3-KTO offers a computationally efficient, data-efficient pathway to improve the physical realism and predictive confidence of AI-driven co-folding. In the near term, this advance promises to accelerate early-stage drug discovery and reduce the experimental burden on wet-lab teams. In the longer term, our approach paves the way for broader adoption of RL based structural biology methods, fostering more reliable design of therapeutics and other bio-molecular applications with the attendant responsibilities to ensure ethical deployment and guard against misuse.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Ahdritz, G., Bouatta, N., Kadyan, S., Jarosch, L., Berenberg, D., Fisk, I., Watkins, A., Ra, S., Bonneau, R., and AlQuraishi, M. Openproteinset: Training data for structural biology at scale. *Advances in Neural Information Processing Systems*, 36:4597–4609, 2023.

Buttenschoen, M., Morris, G. M., and Deane, C. M. Posebusters: Ai-based docking methods fail to generate physically valid poses or generalise to novel sequences. *Chemical Science*, 15(9):3130–3139, 2024.

Corso, G., Stärk, H., Jing, B., Barzilay, R., and Jaakkola, T. Diffdock: Diffusion steps, twists, and turns for molecular docking. *arXiv preprint arXiv:2210.01776*, 2022.

Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Ethayarajh, K., Xu, W., Muennighoff, N., Jurafsky, D., and Kiela, D. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

Garcia, F. and Rachelson, E. Markov decision processes. *Markov Decision Processes in Artificial Intelligence*, pp. 1–38, 2013.

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Kahneman, D. and Tversky, A. Prospect theory: An analysis of decision under risk. In *Handbook of the fundamentals of financial decision making: Part I*, pp. 99–127. World Scientific, 2013.

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35: 26565–26577, 2022.

Li, S., Kallidromitis, K., Gokul, A., Kato, Y., and Kozuka, K. Aligning diffusion models by optimizing human utility. *arXiv preprint arXiv:2404.04465*, 2024.

Liu, L., Zhang, S., Xue, Y., Ye, X., Zhu, K., Li, Y., Liu, Y., Gao, J., Zhao, W., Yu, H., et al. Technical report of helixfold3 for biomolecular structure prediction. *arXiv preprint arXiv:2408.16975*, 2024.

Liu, Z., Su, M., Han, L., Liu, J., Yang, Q., Li, Y., and Wang, R. Forging the basis for developing protein–ligand interaction scoring functions. *Accounts of chemical research*, 50(2):302–309, 2017.

Mirdita, M., Von Den Driesch, L., Galiez, C., Martin, M. J., Söding, J., and Steinegger, M. Uniclust databases of clustered and deeply annotated protein sequences and alignments. *Nucleic acids research*, 45(D1):D170–D176, 2017.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36: 53728–53741, 2023.

Richardson, L., Allen, B., Baldi, G., Beracochea, M., Bileschi, M. L., Burdett, T., Burgin, J., Caballero-Pérez, J., Cochrane, G., Colwell, L. J., et al. Mgnify: the microbiome sequence data analysis resource in 2023. *Nucleic acids research*, 51(D1):D753–D759, 2023.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Sidi, T. and Keasar, C. Redundancy-weighting the pdb for detailed secondary structure prediction using deep-learning models. *Bioinformatics*, 36(12):3733–3738, 2020.

Team, B. A. A., Chen, X., Zhang, Y., Lu, C., Ma, W., Guan, J., Gong, C., Yang, J., Zhang, H., Zhang, K., et al. Protenix-advancing structure prediction through a comprehensive alphafold3 reproduction. *bioRxiv*, pp. 2025–01, 2025.

Trott, O. and Olson, A. J. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461, 2010.

Wallace, B., Dang, M., Rafailov, R., Zhou, L., Lou, A., Purushwalkam, S., Ermon, S., Xiong, C., Joty, S., and Naik, N. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024.

Wang, R., Fang, X., Lu, Y., and Wang, S. The pdbbind database: Collection of binding affinities for protein- ligand complexes with known three-dimensional structures. *Journal of medicinal chemistry*, 47(12):2977–2980, 2004.

Wohlwend, J., Corso, G., Passaro, S., Reveiz, M., Leidal, K., Swiderski, W., Portnoi, T., Chinn, I., Silterra, J., Jaakkola, T., et al. Boltz-1: Democratizing biomolecular interaction modeling. *bioRxiv*, pp. 2024–11, 2024.

Zhang, Y. and Skolnick, J. Tm-align: a protein structure alignment algorithm based on the tm-score. *Nucleic acids research*, 33(7):2302–2309, 2005.

A. A Brief Discussion of Existing Works

In this section, We discuss the connection AF3-KTO between existing work, especially on the reinforcement learning side.

A.1. Reinforcement Learning from Human Feedback

Using a single signal as guidance may not always lead to the desired generative model, particularly when the metrics involved are subjective, difficult to define, or not fully understood. To improve generative models, there is a growing need for them to develop an implicit understanding of desirable objectives—learning human or natural preferences rather than relying solely on explicit numerical signals. In the context of Reinforcement Learning from Human Feedback (RLHF), a reward model is typically trained to assess human preferences, which then informs the policy gradient process. However, training reward models presents challenges, such as resolving preference conflicts and dealing with the scarcity of preference data.

A.2. Direct Preference Optimization

One of the most significant architectural advancements in RLHF is the elimination of explicit reward models. Building on Proximal Policy Optimization (PPO)(Schulman et al., 2017), Direct Preference Optimization (DPO) method(Rafailov et al., 2023) have enabled direct training by modifying the objective function during training, rather than relying on a separate reward model to guide the optimization process. This shift represents a substantial step forward, reducing the complexity of RL and making it more scalable. There are two insights of innovation in the scope if DPO:

- Unification of objectives across the entire training pipeline, from pre-training to fine-tuning and preference optimization.
- Accommodation of DPO for diverse data structures to enhance data availability and efficiency.

A.3. Kahneman Tversky Optimization

KTO(Ethayarajh et al., 2024) is a more broadly practical preference optimization method built upon Direct Preference Optimization (DPO)(Rafailov et al., 2023) with superior data efficiency and more reasonable weight function. It draws on the well-known "Prospect Theory" in economics, originally proposed by Kahneman and Tversky in 1992(Kahneman & Tversky, 2013). Prospect Theory characterizes a rational decision-maker's behavior under uncertainty: when gains are large, the marginal increase in perceived value slows down; likewise, when losses are large (i.e., substantial negative returns), the marginal decrease in perceived value also decelerates. Only when gains and risks are of similar magnitude does perceived value change markedly: at which point humans tend to "take a gamble" rather than accept the given outcome. The resulting value–gain curve is thus well approximated by a sigmoid function.

In this context, the log-likelihood loss employed by DPO didn't match human prospect. although the log-likelihood grows slowly under large positive gains, it still tends toward negative infinity for large negative losses, rendering this behavior inherently risk-averse. Such imbalance is a primary source of the persistent fixed bias seen in DPO. Consequently, the key innovation of KTO is to substitute DPO's log-likelihood with an approximately symmetric weighting function that better mirrors human preference judgments, thereby further reducing dependence on tightly paired positive and negative samples and requiring only binary labels for each datum.

A.4. Aligning Diffusion Models by RLHF

Reinforcement Learning from Human Feedback (RLHF) was initially developed to align Large Language Models (LLMs) with human preferences, particularly for text generation tasks like summation or dialogue complement. However, it has since been adapted to optimize the performance of Diffusion Models. Most studies treat the diffusion process as a Markov Decision Process (MDP) and aim to find a trajectory that maximizes the reward. From this prospective, method Diffusion-DPO(Wallace et al., 2024) proposed an upper bound of the original DPO(Rafailov et al., 2023) loss to estimate implicit reward along the diffusion trajectory, which is further inherited in study Diffusion-KTO (Li et al., 2024).

At last, AF3-KTO follows the theoretical framework of Diffusion-KTO (Li et al., 2024), transferring the task from text-to-image to protein-ligand co-folding. The process includes insertion of complex while exquisite AlphaFold3 architecture and enabling practical algorithm with min convertion of backbone (2).

B. Mathematical Derivations

B.1. A More Accurate Setting for AlphaFold3 Architecture

In this part, we dive into the implement of conditional diffusion module of AlphaFold3. Actually, the input condition of module are multiple representations: (i) token level single conditioning s_i; (ii) token level pair conditioning input z_{ij}; (iii) atom level single conditioning c_i; (iv) atom level pair conditioning p_{ij}; Again we assemble then into a formula of y:

$$y = \{s_i, z_{ij}, c_i, p_{ij}\} \tag{10}$$

We're goint to use $\hat{x_\theta}(x_t, t, y)$ to represent the denoised results after the whole diffusion modulre, it consist of: (i) adding noise with a random sample t by input coodinates x_0 to get input of network x_t; forwarding process of network backbone (the diffusion conditioning extraction, atom attention encoder, diffusion transformer and atom attention decoder), which is Algorithm 20 in the supplying material in AlphaFold3 paper, resulting in $\hat{x_\theta}$;

Given the forwarding function of diffusion module, the diffusion MSE loss of AlphaFold3 could be calculated as:

$$L_{\mathbf{AF3-MSE-Diffusion}} = \frac{1}{3}\Omega(x) \quad ||x_0^{aligned} - \hat{x_\theta}(x_t, t, y)||_2^2, \quad x \in \mathbb{R}^3 \tag{11}$$

where $x^{aligned}$ means the ground truth complex is aligned with denoised complex in the context of protein backbone, Ω is a weighted function related to biological type of data sample:

$$\Omega = 1 + f^{is\_dna} \cdot 5 + f^{is\_rna} \cdot 5 + f^{is\_ligand} \cdot 10 \tag{12}$$

B.2. Directly Apply KTO into AlphaFold3 Architecture

Lemma B.1. *For an unmatched pair (x^i, x^j) each under diffusion process $(x_t^i, x_t^j), t \in [0, T]$:*

$$-E_{x_{t-1}^j \sim q(x_{t-1}^j | x_0^j, y^j)} \log \frac{\pi_\theta(x_{t-1}^j \mid x_t^i, y^i)}{\pi_{ref}(x_{t-1}^j \mid x_t^i, y^i)} \leq ||x_0^j - \hat{x_\theta}(x_t^i, t, y^i)||_2^2 - ||x_0^j - \hat{x_{ref}}(x_t^i, t, y^i)||_2^2 \tag{13}$$

where $\hat{x_\theta}(x_t^i, t, y^i)$ are implement of diffusion module including predicting distribution $p_\theta(x_{t-1}^i \mid x_t^i, y^i)$ and reconstruct denoised sample $\hat{x_\theta}$.

Proof.

$$\begin{aligned}
LHS &= -E_{x_{t-1}^j \sim q(x_{t-1}^j | x_0^j, y^j)} \log \frac{\pi_\theta(x_{t-1}^j \mid x_t^i, y^i)}{\pi_{ref}(x_{t-1}^j \mid x_t^i, y^i)} && (14)\\
&= E_{x_{t-1}^j \sim q(x_{t-1}^j | x_0^j, y^j)} [\log \frac{q(x_{t-1}^j \mid x_0^j, y^j)}{\pi_\theta(x_{t-1}^j \mid x_t^i, x^i)} - \log \frac{q(x_{t-1}^j \mid x_0^j, y^j)}{\pi_{ref}(x_{t-1}^j \mid x_t^i, y^i)}] && (15)\\
&= D_{KL}[q(x_{t-1}^j \mid x_0^j, y^j) \mid\mid \pi_\theta(x_{t-1}^j \mid x_t^i, y^i)] - D_{KL}[q(x_{t-1}^j \mid x_0^j, y^j) \mid\mid \pi_{ref}(x_{t-1}^j \mid x_t^i, y^i)] && (16)\\
&\leq ||x_0^j - \hat{x_\theta}(x_t^i, t, y^i)||_2^2 - ||x_0^j - \hat{x_{ref}}(x_t^i, t, y^i)||_2^2 && (17)\\
&= RHS && (18)\\
& && (19)
\end{aligned}$$

□

Specificly, when data pairs are matched ($i = j$), that formula comes into exactly diffusion MSE loss:

$$-E_{x_{t-1} \sim q(x_{t-1} | x_0, y)} \log \frac{\pi_\theta(x_{t-1} \mid x_t, y)}{\pi_{ref}(x_{t-1} \mid x_t, y)} \leq ||x_0 - \hat{x_\theta}(x_t, t, y)||_2^2 - ||x_0 - \hat{x_{ref}}(x_t, t, y)||_2^2 = L_\theta - L_{ref} \tag{20}$$

Algorithm 1 Training algorithm of AF3-KTO

Input: a batch of noisy data with conditions $\{[x_t^1 \mid y^1]...[x_t^m \mid y^m]$, size m. $x \in \mathbb{R}^3$, $y = \{s_i, z_{ij}, c_i, p_{ij}\}$
Model: $L_\theta(x_{label}, x_t, y) = \frac{1}{3}\omega(x)\ ||x_{label} - x_\theta(\hat{x_t}, t, y)||_2^2$ and $L_{ref}(x_{label}, x_t, y) = \frac{1}{3}\omega(x)\ ||x_{label} - x_\theta(\hat{x_t}, t, y)||_2^2$
Construct mismatch pairs $\{([x_t^i \mid y^i], [x_t^j \mid y^j])\}$, $j = i + 1 \mod m$.
for i, j **in mismatch pairs do**
  $L_{mis-k} \leftarrow \frac{3}{\Omega}[L_\theta(x_0^j, x_t^i, y^i) - L_{ref}(x_0^j, x_t^i, y^i)]$
end for
$Q_{ref} = \max(0, \frac{1}{m}\sum_{k=1}^{m-1} L_{mis-k})$
for $k = 1$ **to** m **do**
  label $w(x_0^k) = 1 or -1$
  $L_{match} \leftarrow \frac{3}{\Omega}L_\theta(x_0^k, x_t^k, y^k) - L_{ref}(x_0^k, x_t^k, y^k)$
  $L_{AF3-KTO} \leftarrow U(w(x_0^k)\beta\ (L_{match} - Q_{ref}))$
  Update $\hat{x_\theta}(x_t, t, y)$ through gradient back propagation
end for

where L_θ representing the diffusion MSE loss of optimizing diffusion module, while L_{ref} is the pre-trained diffusion module.

Then we're able to derive a clarified version of AF3-KTO loss (Equation 7) given diffusion module of AlphaFold3. Firstly, let's take a look at a full version of our loss:

$$L_{\mathbf{AF3-KTO}} = -E_{c,(x_0^i,y^i)\in D, t\in[0,T]}[U(w(x_0)\beta\ (\log\frac{\pi_\theta(x_{t-1}|x_t,y)}{\pi_{ref}(x_{t-1}|x_t,y)}) - Q_{ref})] \tag{21}$$

$$= -E_{c,(x_0^k,y^k)\in\{([x^0|y^0]...[x^m|y^m])\}, t\in[0,T]}[U(w(x_0^k)\beta\ (\log\frac{\pi_\theta(x_{t-1}^k|x_t^k,y)}{\pi_{ref}(x_{t-1}^k|x_t^k,y)}) - \max(0, \frac{1}{m}\sum_{i=1}^{m-1}\log\frac{\pi_\theta(x_{t-1}^j|x_t^i,y^i)}{\pi_{ref}(x_{t-1}^j|x_t^i,y^i)}))] \tag{22}$$

where we're optimizing k^{th} data over a batch size of m with its unmatched pair (i, j). Since both x_{t-1}^j and are acquired by forward diffusion process:

$$L_{\mathbf{AF3-KTO}} \tag{23}$$

$$= -E_{c,(x_0^k,y^k)\in\{([x^0|y^0]...[x^m|y^m])\}, t\in[0,T],\ x_{t-1}^j\sim q(x_{t-1}^j|x_0^j,t),\ x_{t-1}^k\sim q(x_{t-1}^k|x_0^k,t)}$$
$$[U(w(x_0^k)\beta\ (\log\frac{\pi_\theta(x_{t-1}^k|x_t^k,y)}{\pi_{ref}(x_{t-1}^k|x_t^k,y)}) - \max(0, \frac{1}{m}\sum_{i=1}^{m-1}\log\frac{\pi_\theta(x_{t-1}^j|x_t^i,y^i)}{\pi_{ref}(x_{t-1}^j|x_t^i,y^i)}))] \tag{24}$$

$$\leq -E_{c,(x_0^k,y^k)\in\{([x^0|y^0]...[x^m|y^m])\}, t\in[0,T]}[U(w(x_0^k)\beta\ (E_{x_{t-1}^k\sim q(x_{t-1}^k|x_0^k,t)}\log\frac{\pi_\theta(x_{t-1}^k|x_t^k,y)}{\pi_{ref}(x_{t-1}^k|x_t^k,y)}$$
$$- E_{x_{t-1}^j\sim q(x_{t-1}^j|x_0^j,t)}\max(0, \frac{1}{m}\sum_{i=1}^{m-1}\log\frac{\pi_\theta(x_{t-1}^j|x_t^i,y^i)}{\pi_{ref}(x_{t-1}^j|x_t^i,y^i)}))] \tag{25}$$

$$\leq E_{c,(x_0^k,y^k)\in\{([x^0|y^0]...[x^m|y^m])\}, t\in[0,T]}[U(w(x_0^k)\beta\ (L_\theta^k - L_{ref}^k) - \max(0, \frac{1}{m}\sum_{i=1}^{m-1}\{||x_0^j - \hat{x_\theta}(x_t^i,t)||_2^2 - ||x_0^j - \hat{x_{ref}}(x_t^i,t)||_2^2\}))] \tag{26}$$

where L_θ^k and L_{ref}^k is the diffusion MSE loss with input k^{th} data, $||x_0^j - \hat{x_\theta}(x_t^i,t)||_2^2$ and $||x_0^j - \hat{x_{ref}}(x_t^i,t)||_2^2$ are the diffusion MSE loss between i^{th} input data and j^{th} labels. In all, the algorithm of AF3-KTO are be concluded in Algorithm 2.

Algorithm 2 Training algorithm of AF3-KTO

Input: a batch of noisy data with conditions $\{[x_t^1 \mid y^1]...[x_t^m \mid y^m]$, size m. $x \in \mathbb{R}^3$, $y = \{s_i, z_{ij}, c_i, p_{ij}\}$
Model: $L_\theta(x_{label}, x_t, y) = \frac{1}{3}\omega(x)\ ||x_{label} - x_\theta(\hat{x_t}, t, y)||_2^2$ and $L_{ref}(x_{label}, x_t, y) = \frac{1}{3}\omega(x)\ ||x_{label} - x_\theta(\hat{x_t}, t, y)||_2^2$
Construct mismatch pairs $\{([x_t^i \mid y^i], [x_t^j \mid y^j])\}$, $j = i + 1 \mod m$.
$x \leftarrow [y_p, y_m]$

Figure 8. Loss Function in KTO Optimization Process

C. Experimental Details

C.1. Training Configurations

We train our AF3-KTO model on synthetically generated datasets, where each sample is randomly sub-sampled according to a user-defined positive-sample ratio. Compared to the hyper-parameter settings used in Diffusion-KTO for text-to-image tasks (Li et al., 2024), we make the following adjustments in AF3-KTO :

1. **Learning rate**: reduced from 5×10^{-4} to 1×10^{-4}
2. **DPO KL penalty** (β): from 100 to 10
3. **Positive sampling ratio**: from 0.5 to 0.9

All other training hyper-parameters follow the Protenix conditional diffusion backbone without modification. Both the KTO-optimized model and baseline model (and vaiants fine-tuned on only positives samples) are trained for 2,000 steps; we select and report the best-performing checkpoint for each to minimize the risk of over-fitting.

Figure 8 illustrates the training loss curves for AF3-KTO , demonstrating that KTO effectively conducts tractable and efficient preference optimization on the AlphaFold3-inspired conditional diffusion architecture.

C.2. Baselines

We compare AF3-KTO against three strong baselines:

- **AlphaFold3**: The state-of-the-art conditional diffusion–based predictor for protein structure, recently shown to surpass specialized end-to-end docking models in accuracy and robustness.
- **Protenix**: A representative reproduction of the AlphaFold3 architecture, whose performance on protein ligand is enhanced by extra data collected.
- **Protenix-SFT**: Protenix fine-tuned on all positive examples from our synthetic dataset.

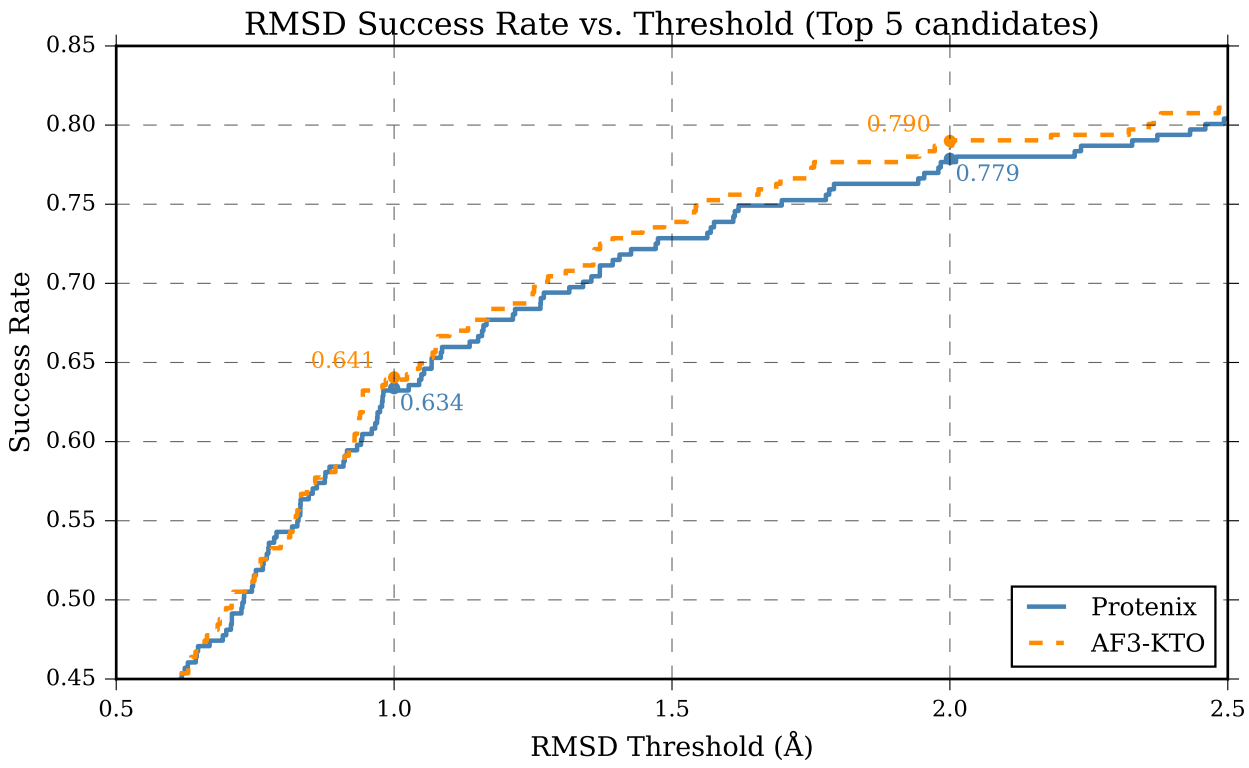

Figure 9. RMSD success rate curve with threshold on Top5 candidates

C.3. Analysis of optimization results

In Figure 4, we compare root-mean-square deviation (RMSD) and PB-Valid (physical plausibility) metrics across models. AF3-KTO consistently outperforms both the pretrained-only AlphaFold3(Protenix) and the supervised Protenix-SFT baselines, achieving higher accuracy and improved physical plausibility. This confirms that human-preference optimization can correct physically implausible predictions without compromising binding-pose accuracy.

Figure 9 and 10 examines RMSD success rates when selecting only the top-k candidate poses. Consistent with the ablation results in Table 1, AF3-KTO maintains its lead over baselines across different candidate thresholds. Notably:

- RMSD performance degrades more slowly than PB-Valid as k decreases, indicating that AlphaFold3-style models already achieve high accuracy on high-confidence poses.

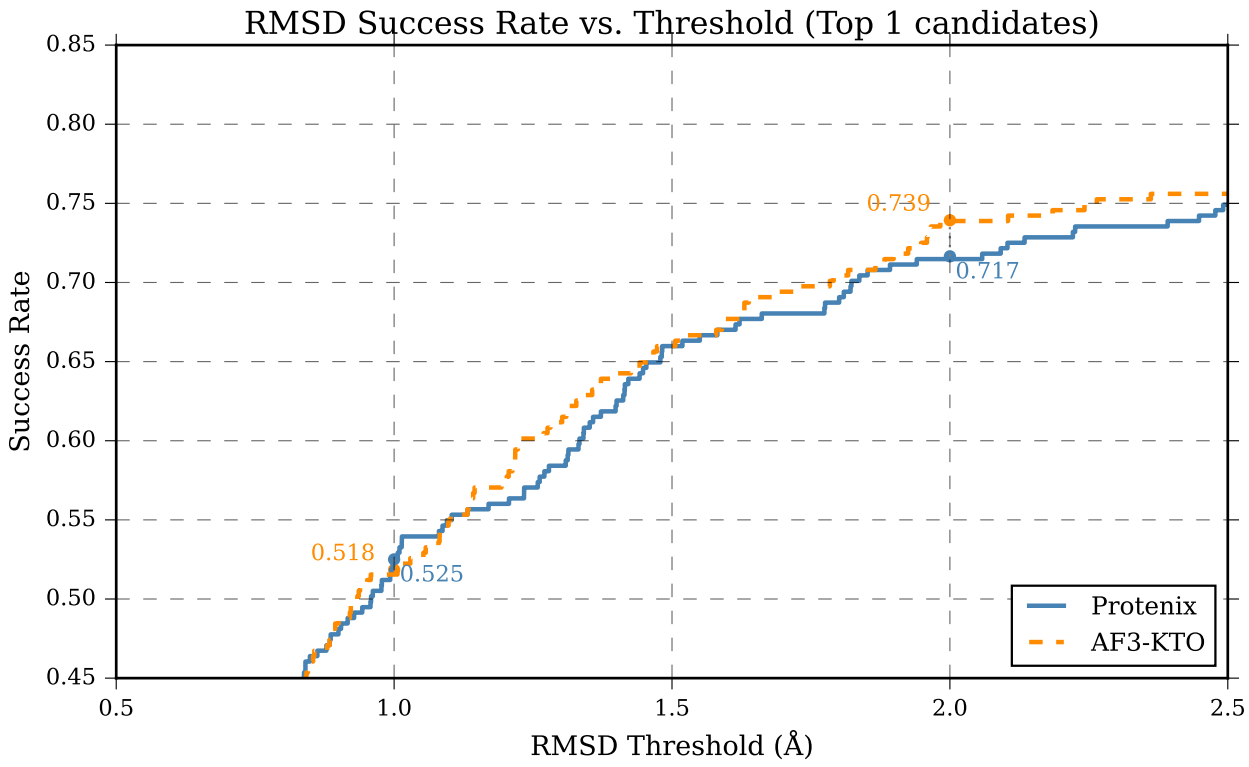

Figure 10. RMSD success rate curve with threshold on Top1 candidates

- Even when optimizing only for RMSD<2Å, KTO yields marginal gains at tighter thresholds (e.g. RMSD<1Å), suggesting an overall distributional shift toward lower RMSD values.

Finally, Figure 6 presents representative ligand–protein complexes before and after KTO optimization. While preserving the global ligand geometry, KTO substantially reduces steric clashes with the receptor, illustrating how preference-based optimization effectively mitigates common docking errors and enhances the reliability of predicted binding poses.