FoldMark: Protecting Protein Generative Models with Watermarking

Zaixi Zhang[1], Ruofan Jin[2], Kaidi Fu[3], Le Cong[4], Marinka Zitnik[5], and Mengdi Wang[1,✉]

[1]Princeton University, NJ, USA
[2]Zhejiang University, Zhejiang, China
[3]Tsinghua University, Beijing, China
[4]Stanford University, CA, USA
[5]Department of Biomedical Informatics, Harvard Medical School, Boston, MA, USA
✉mengdiw@princeton.edu

ABSTRACT

Protein structure is key to understanding protein function and is essential for progress in bioengineering, drug discovery, and molecular biology. Recently, with the incorporation of generative AI, the power and accuracy of computational protein structure prediction/design have been improved significantly. However, ethical concerns such as copyright protection and harmful content generation (biosecurity) pose challenges to the wide implementation of protein generative models. Here, we investigate whether it is possible to embed watermarks into protein generative models and their outputs for copyright authentication and the tracking of generated structures. As a proof of concept, we propose a two-stage method FoldMark as a generalized watermarking strategy for protein generative models. FoldMark first pretrain watermark encoder and decoder, which can minorly adjust protein structures to embed user-specific information and faithfully recover the information from the encoded structure. In the second step, protein generative models are fine-tuned with watermark Low-Rank Adaptation (LoRA) modules to preserve generation quality while learning to generate watermarked structures with high recovery rates. Extensive experiments are conducted on open-source protein structure prediction models (e.g., ESMFold and MultiFlow) and de novo structure design models (e.g., FrameDiff and FoldFlow) and we demonstrate that our method is effective across all these generative models. Meanwhile, our watermarking framework only exerts a negligible impact on the original protein structure quality and is robust under potential post-processing and adaptive attacks.

Introduction

Proteins are life's essential building blocks and the basis of all living organisms. Understanding their structure is key to uncovering the mechanisms behind their function. With the advancement of generative AI[1], protein structure generative models have revolutionized both protein structure prediction[2,3] and de novo protein design[4,5], opening up a wide range of applications in bioengineering and drug discovery. For example, AlphaFold2[2] made a breakthrough by accurately predicting protein structures from amino acid sequences at near-experimental accuracy, solving a decades-old challenge in biology. Its successor, AlphaFold3[3], further improved on this by enhancing the ability to model more complex protein interactions and assemblies. Meanwhile, RFDiffusion[4] and Chroma[5] introduced diffusion-based generative models that enable the creation of novel protein structures with desired properties and functions, pushing the boundaries of de novo protein design. In recognition of the profound impact of these models, the 2024 Nobel Prize in Chemistry was awarded to David Baker "for computational protein design" and to Demis Hassabis and John M. Jumper "for protein structure prediction"[6].

However, the rapid development of protein generative models and the lack of corresponding regulations lead to copyright and biosecurity concerns. First, the ease of model sharing raises copyright concerns, such as the unauthorized use of generated structures and the redistribution of pretrained models for profit, potentially compromising the interests of the original creators[7–9]. For example, the latest AlphaFold3 is only deployed on the server with the terms saying that users "must not use AlphaFold Server or its outputs to train machine learning models" or "in connection with any commercial activities"[10]. Second, the unregularized yet powerful protein generative models are vulnerable to misuse and cause bio-security/safety concerns[11,12]. For example, protein generative models can be used to design new proteins with harmful properties, such as pathogens[13], toxins[14], or viruses[15] that can be used as bioweapons. Therefore, there is an urgent need for a reliable and efficient tool to track and audit the use of protein generative models.

Similar problems have occurred in text and image generation. For example, Large language models (LLMs) such as ChatGPT can be used to create fake news and to cheat on academic writing[16–18]. The latest text-to-image models such as

Stable Diffusion[19] and DALL·E 3[20] enable users to create photo-realistic images like deep fakes[21] for illegal purposes. As a result, there is growing consensus that the ability to detect, track, and audit the use of AI-generated content is essential for harm reduction and regulation[17,22]. Recently, the watermark becomes one of the most promising protection strategy, which embeds hidden patterns in the generated content and is imperceptible to humans, while making the embedded information algorithmically identifiable. Although watermark has been applied for LLM[23–27] and text-to-image models[28–31], extending these methods to protein structure data presents unique challenges. Unlike text and images, protein structures are highly sensitive to minute changes[2,32,33], and embedding watermarks without disrupting the biological functionality or stability of the proteins is a complex task. Moreover, protein structures exhibit complex geometrical symmetries, making traditional watermarking methods less effective due to the requirement for equivariance[4,34].

In this paper, as a proof of concept, we propose FoldMark, a generalized watermarking method for protein generative models, e.g., AlphaFold2[2] and RFDiffusion[4]. FoldMark builds upon pretrained protein generative models and generally has two training stages. In the first stage, SE(3)-equivariant watermark encoder and decoder are pretrained to learn how to embed watermark information into protein structure without compromising the structural quality. Specifically, the encoder takes the watermark code and learns to construct the watermark-conditioned structure; the decoder takes the watermarked structure and predicts the embedded watermark. In the second stage, we propose watermark LoRA[35], which flexibly encodes the given watermark code and merges it into the original model weights, without changing or adding extra model architecture. The protein generative model is fine-tuned with two objectives, named message retrieval loss and consistency loss. The message retrieval loss ensures the effective embedding of watermarks into the generated structure, allowing for successful retrieval of the embedded watermark codes. Meanwhile, the consistency loss ensures that the inclusion of the watermark has minimal impact on the overall quality of the protein structure.

In experiments, we first observe that FoldMark achieves nearly 100% bit accuracy on watermark code recovery from encoded protein structures with minimal influence on structural validity (measured by scRMSD and RMSD), which means FoldMark can reliably embed and retrieve watermarks with minimal structural deviation. Compared with baseline methods, FoldMark achieves consistent improvements and can successfully handle 16-bit watermark code. We further consider two application scenarios: detection for copyright protection and identification for tracing the creator. Specifically, in detection, FoldMark aims to verify whether a structure is generated by a certain model; in identification, FoldMark aims to find the exact user that generates the structure by watermark matching (see Fig. 1). Finally, we demonstrate the robustness of FoldMark under multiple post-processing (e.g., Cropping, Rotation+Translation, Coordinates Noising) and adaptive attacks (Finetuning attack and Multi-Message attack). Collectively, FoldMark efficiently embeds watermarks while maintaining the quality of protein structure data, offering a new approach to ensuring biosecurity in protein design in the era of generative AI.

Results

Overview of FoldMark

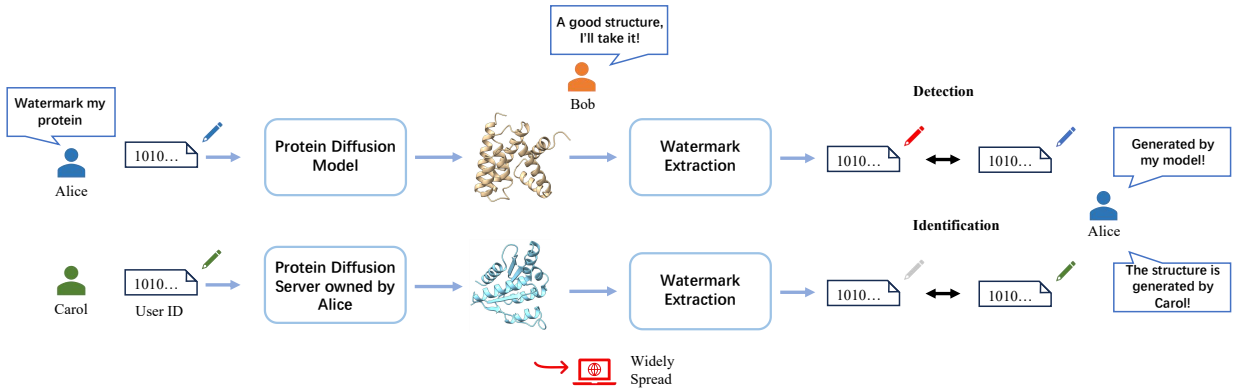

Figure 1. Illustrations of application scenarios of FoldMark. The scenario involves the model owner Alice, the thief Bob, and the user Carol. Alice is responsible for training the model, releasing the pretrained model and inference code, or deploying it on the platform for users. Bob, who downloads Alice's model and code, generates protein structures and falsely claims ownership of the copyrights. Carol registers as a user on the server and utilizes the API to generate protein structures. Alice, as the model owner, may wish to restrict the use of these generated structures, particularly in commercial contexts, to avoid copyright infringement. Using FoldMark, Alice can embed a watermark within the generated protein structures, allowing her to extract the watermark code for detection and identification of unauthorized usage, safeguarding her intellectual property.

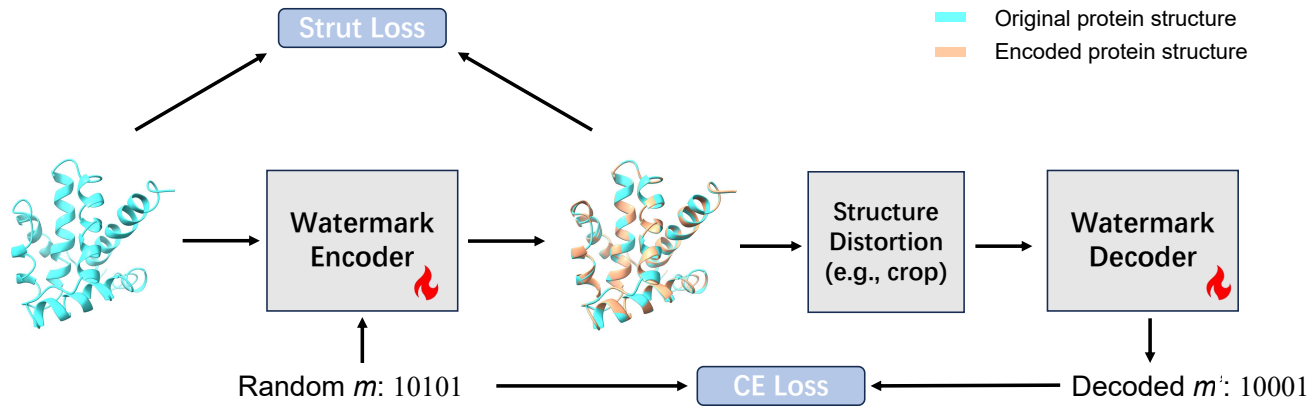

Figure 2. Pretraining stage of FoldMark.

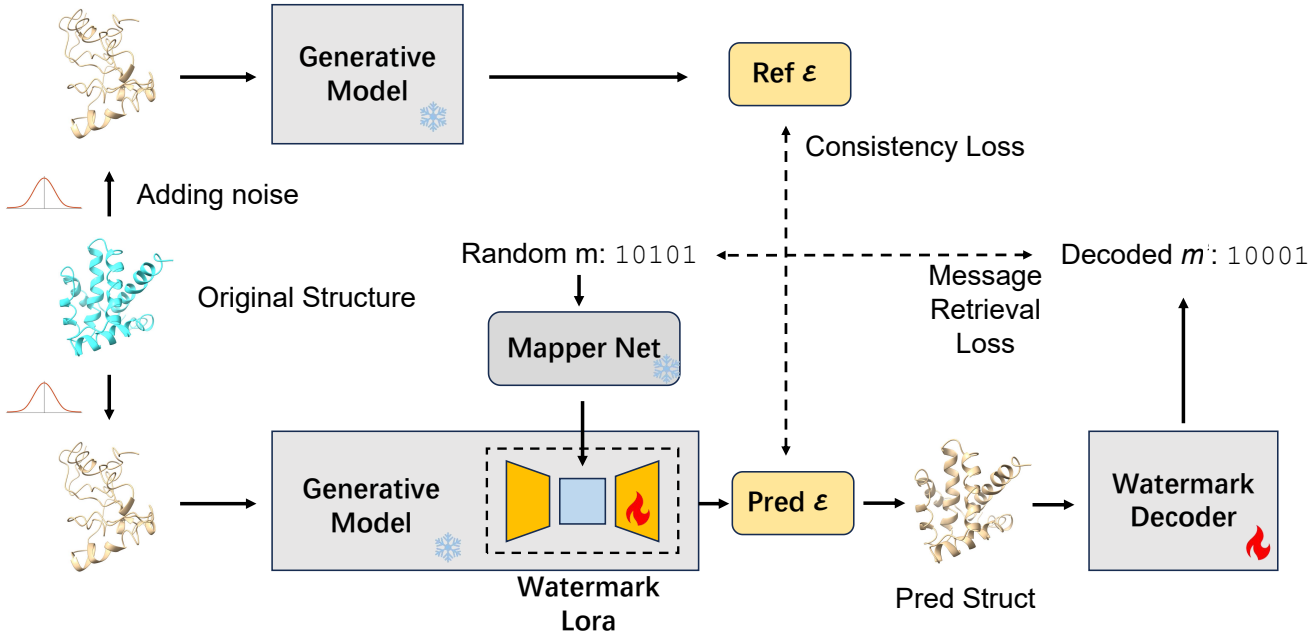

Figure 3. Finetuning stage of FoldMark.

Watermarking State-of-the-art protein generative models

In Table. 1 and 2, we showed experiments of watermarking unconditional protein structure generative models (i.e., FoldFlow[36], FrameDiff[37], and FrameFlow[38]) and protein structure prediction models (i.e., MultiFlow[39] and ESMFold[40]). We vary the watermark code length from 4 to 32 and measure the bit prediction accuracy (BitAcc) and the structural validity (scRMSD and RMSD). To benchmark the performance of FoldMark, we also adapt two watermark methods from the image domain, WaDiff[29] and AquaLoRA[30] for comparison. Generally, the performance degrades with the increase of watermarking capacity, i.e., more watermark bits. On most cases with less than 16 bits, FoldMark achieves nearly 100% bit accuracy on watermark code recovery from encoded protein structures with minimal influence on structural validity (measured by scRMSD and RMSD). Therefore, FoldMark is a generalized and effective method for protein generative model protection.

Applications in detection and user identification

As shown in Figure 1, we show two applications of FoldMark. The scenario involves Alice, the model owner responsible for training, releasing the pretrained model, and deploying the inference code on the platform. Bob, a thief, downloads Alice's model and code to generate protein structures, falsely claiming ownership of the copyrights. Carol registers as a user on the server and utilizes the API to generate protein structures. In the detection, the successful extraction of a watermark from structures serves as proof of Alice's rightful ownership of the copyright and indicates that the structure is artificially generated.

In user identification, Alice assigns a unique watermark to each user. By extracting the watermark from generated structures, it becomes possible to trace it back to Carol by comparing it with the watermark database and regarding the most similar user id. Traceability goes beyond detection, enabling copyright protection for different users by identifying the source of infringement. In Table 3, we show the identification accuracy for different generative models with different numbers of users. While FoldMark achieves strong performance with small groups of users, it becomes much more challenging for identification among a larger number of users (e.g., 10^6).

Table 1. Watermarking unconditional protein structure generative models

Watermark Method		4bit		8bit		16bit		32bit	
		BitAcc ↑	scRMSD ↓	BitAcc ↑	scRMSD ↓	BitAcc ↑	scRMSD ↓	BitAcc ↑	scRMSD ↓
FoldFlow	No watermark	-	1.926	-	1.926	-	1.926	-	1.926
	WaDiff	88.2%	2.107	85.0%	2.115	80.1%	2.397	64.3%	2.630
	AquaLoRA	73.5%	2.056	72.6%	2.210	71.7%	2.446	62.8%	2.718
	FoldMark(Ours)	**99.9%**	**1.937**	**99.7%**	**1.980**	**98.9%**	**2.114**	**94.5%**	**2.307**
FrameDiff	No watermark	-	2.850	-	2.850	-	2.850	-	2.850
	WaDiff	76.8%	2.919	73.3%	3.235	62.2%	3.810	50.4%	4.058
	AquaLoRA	64.3%	3.150	59.1%	3.431	56.2%	3.890	51.6%	4.179
	FoldMark(Ours)	**98.7%**	**2.795**	**98.3%**	**2.914**	**88.4%**	**3.045**	**82.0%**	**3.428**
FrameFlow	No watermark	-	1.855	-	1.855	-	1.855	-	1.855
	WaDiff	77.1%	1.883	76.4%	2.270	63.5%	2.456	54.6%	2.823
	AquaLoRA	63.6%	1.920	61.4%	2.317	54.5%	2.680	52.1%	2.953
	FoldMark(Ours)	**99.6%**	**1.860**	**99.5%**	**1.939**	**96.7%**	**2.019**	**95.4%**	**2.192**

Table 2. Watermarking protein structure prediction models

Watermark Method		4bit		8bit		16bit		32bit	
		BitAcc ↑	RMSD ↓	BitAcc ↑	RMSD ↓	BitAcc ↑	RMSD ↓	BitAcc ↑	RMSD ↓
MultiFlow	No watermark	-	3.344	-	3.344	-	3.344	-	3.344
	WaDiff	64.3%	4.342	62.0%	4.583	56.4%	5.746	52.1%	5.643
	AquaLoRA	65.0%	3.626	58.9%	4.010	51.2%	4.398	50.4%	5.277
	FoldMark(Ours)	**99.7%**	**3.520**	**98.6%**	**3.534**	**97.0%**	**3.570**	**89.5%**	**3.688**
ESMFold	No watermark	-	2.241	-	2.241	-	2.241	-	2.241
	WaDiff	79.5%	**2.426**	77.1%	2.643	72.6%	2.779	59.0%	2.800
	AquaLoRA	76.4%	2.446	71.8%	2.547	66.3%	2.605	54.4%	2.820
	FoldMark(Ours)	**94.6%**	2.453	**93.2%**	**2.506**	**86.9%**	**2.571**	**85.0%**	**2.680**

Robustness against post-processing and adaptive attacks

In real applications, the malicious user may take post-processing or design adaptive attacks to bypass the safeguarding of FoldMark. Here, we consider three common post-processing methods for the protein structure and two adaptive attacks in Table. 4. Adaptive attacks involve fine-tuning the watermarked model using clean protein data to erase the watermark, or performing a multi-message attack, where additional watermarks are injected to obscure the original ones. We can observe that FoldMark is robust to cropping, translation, and rotation because the watermark information is encoded into each residue and the watermark decoder is SE(3) invariant. Due to the integrated design and data augmentation, FoldMark is resistant to finetuning and multi-message injection.

Discussion

In this paper, our study demonstrates the feasibility of embedding watermarks into protein generative models and their outputs through our proposed method, FoldMark. This two-stage approach successfully preserves the quality of protein structures while embedding user-specific information for copyright authentication and tracking. Extensive experiments on various protein structure prediction and design models confirm the effectiveness and robustness of FoldMark against post-processing and

Table 3. Performance of FoldMark under post-processing and adaptive attacks. Protein post-processing include structure cropping (keeping 50% of the whole sequence), randomly translating & rotating the whole structure, and adding Gaussian noise to the coordinates (strength 0.2). Adaptive attacks include fine-tuning the watermarked model with clean protein data to erase the watermarking capability and multi-message attack that try to inject additional watermarks to cover the original ones. We conduct experiments on the 16bits setting.

Model	**No Attack**	**Cropping**	**Trans&Rotate**	**Noising**	**Finetune**	**Multi-Msg**
FoldFlow	0.989	0.961	0.990	0.910	0.920	0.947
FrameDiff	0.884	0.860	0.882	0.793	0.769	0.860
FrameFlow	0.967	0.906	0.960	0.871	0.870	0.948
MultiFlow	0.970	0.864	0.972	0.826	0.924	0.950
ESMFold	0.869	0.829	0.874	0.805	0.856	0.862

Table 4. Performance of FoldMark user identification accuracy.

Model	10^3 **users**	10^4 **users**	10^5 **users**	10^6 **users**
FoldFlow	0.970	0.970	0.943	0.900
FrameDiff	0.705	0.393	0.309	0.225
FrameFlow	1.000	0.992	0.980	0.931
MultiFlow	1.000	0.996	0.940	0.817
ESMFold	0.903	0.824	0.450	0.334

adaptive attacks, with minimal impact on the original structure quality. This provides a potential solution for addressing ethical concerns, such as copyright protection, in the application of generative AI to protein design.

There are a few limitations that we would like to address in the future. First, our approach struggles with significant structural modifications such as large-scale domain movements or extreme conformational changes, as the watermark's resilience is limited. Currently, the watermark pretraining process is decoupled from the fine-tuning of protein generative models, and future improvements in building end-to-end watermark pipelines could enhance robustness against such structural changes. Additionally, advanced users may apply protein generative models not only for de novo design but also for structure editing, functional optimization, or motif scaffolding. At present, our watermarking technique does not sufficiently address these types of complex modifications, limiting its effectiveness in more advanced usage scenarios. Finally, as the complexity or length of the generated protein increases, we observe some performance degradation in watermark retrieval accuracy. We plan to address this limitation in future work by optimizing our method to scale effectively with larger and more intricate protein structures.

Methods

Figures 2 and 3 provide an overview of our method, FoldMark, which consists of two main stages: Watermark Encoder/Decoder Pretraining and Prior-Preserving Finetuning. The pretraining stage enables the watermark encoder and decoder to learn how to embed watermark information into the structure space and accurately extract it. The finetuning stage equips pretrained protein generative models with watermarking capabilities while preserving their original generative performance (prior-preserving). FoldMark is a versatile method that can be applied to various mainstream protein structure generative models. We use a diffusion-based model as an example and present the details of FoldMark below.

Watermark Encoder/Decoder Pretraining

We first train a watermark encoder $\mathcal{W}$ and decoder $\mathcal{D}$ such that $\mathcal{D}$ can correctly retrieve the watermark message $\mathbf{m}$ embedded by $\mathcal{W}$.

$$\mathcal{L}_{Pretrain} = \mathbb{E}_{\mathbf{x},\mathbf{m},f}[\mathcal{L}_{BCE}(\mathcal{D}(f(\mathcal{W}(\mathbf{x},\mathbf{m}))),\mathbf{m}) + \gamma\|\mathcal{W}(\mathbf{x},\mathbf{m}) - \mathbf{x}\|_2], \tag{1}$$

where $\mathbf{x}$ represents the protein structure data and $\mathbf{m}$ denotes the string of binary watermark code. $\gamma > 0$ is a hyperparameter to control the strength of structure adjustment for watermarking. f represents a randomly selected structure distortion as data augmentation. The pool of data augmentation includes random rotation/translation, adding Gaussian noise to protein coordinates, and randomly cropping the protein structure. $\mathcal{L}_{BCE}(\mathcal{D}(f(\mathcal{W}(\mathbf{x},\mathbf{m}))),\mathbf{m})$ and $\|\mathcal{W}(\mathbf{x},\mathbf{m}) - \mathbf{x}\|_2$ correspond to the CE Loss and Struct Loss in Figure 2 respectively.

Prior-preserving Finetuning

Instead of finetuning all the parameters of the generative model, we selectively fine-tune part of the protein generative model with watermark LoRA and the watermark decoder as shown in Figure 3. The other parameters including the watermark embedder $\mathcal{P}$ and the reference model are kept unchanged. We discuss the details of watermark module in the next subsection.

Here we take the diffusion-based protein generative model (e.g., FrameDiff[37] and RFDiffusion[4]) as an example to construct the fine-tuning loss. The diffusion model typically involves two critical components known as the forward and backward process, where the forward process gradually noises the original protein structure $\mathbf{x}_0$ into $\mathbf{x}_t$ for $t \in \{1, \cdots, T\}$ and the model learns to predict the original structure $\epsilon_\theta(\mathbf{x}_t)$ based on $\mathbf{x}_t$. There are two losses in the fine-tuning: the consistency loss for regularization and the message retrieval loss to encourage correct watermark retrieval. In the consistency loss $\mathcal{L}_c$, the prediction of the fine-tuned model is compared with the original pretrained model so that the finetuned model weights will not deviate too much from the original ones. For the watermark retrieval loss $\mathcal{L}_m$, we take a single reverse step with respect to $\mathbf{x}_t$ to obtain $\hat{\mathbf{x}}_t = (\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(\mathbf{x}_t))/\sqrt{\bar{\alpha}_t}$, and then feed it into the decoder to predict the watermark code. In sum, we incorporate both optimization objectives above and formulate the prior-preserving finetuning loss as:

$$\mathcal{L}_{Finetune} = \mathbb{E}_{\mathbf{x},t,\mathbf{m}}[\mathcal{L}_c(\epsilon_\theta(\hat{\mathbf{x}}_t), \epsilon_{\theta_{ref}}(\mathbf{x}_t)) + \eta \cdot \frac{t-T}{T}\mathcal{L}_m(\mathcal{D}(\hat{\mathbf{x}}_t), \mathbf{m})], \tag{2}$$

where η controls the trade-off between consistency loss $\mathcal{L}_c$ and watermark retrieval loss $\mathcal{L}_m$. We place an additional weight $\frac{t-T}{T}$ for the retrieval loss because the generated structure contains more information of watermark as $t \to 0$ and we observe better performance in experiments.

Watermark LoRA

To save the computation costs of fine-tuning and flexibly embed watermark information in the generation process, we proposed the Watermark LoRA strategy. The computation formula for LoRA can be then written as $\Delta\mathbf{W} = \mathbf{A} \times \mathbf{E} \times \mathbf{B}$, where $\mathbf{A} \in \mathbb{R}^{n\times r}$ and $\mathbf{B} \in \mathbb{R}^{r\times m}$, $\mathbf{E} \in \mathbb{R}^{r\times r}$ as the embedding matrix. Here, r is the rank of the matrix ($r \ll n, m$). When $\mathbf{E}$ is an identity matrix, the computation yields the same result with the vanilla LoRA[35].

To input the watermark information into the fine-tuned model, we apply a mapper that transforms a watermark code of length l into a vector of length r. Specifically, for the i-th bit of the code, we use a embedding vector $\mathbf{I}_i$ and $\mathbf{0}$ to represent the binary states 1 and 0, with $\mathbf{I}_i, \mathbf{0} \in \mathbb{R}^r$. The mapping function $h_i : \{0,1\} \to \mathbb{R}^r$ is defined as

$$h_i(b_i) = \begin{cases} \mathbf{I}_i, & \text{if } b_i = 1, \\ \mathbf{0}, & \text{otherwise.} \end{cases} \tag{3}$$

For a watermark code $\mathbf{m} = \{b_0, b_1, \ldots, b_l\}$, the embedding matrix $\mathbf{E}$ is constructed as

$$\mathbf{S} = \text{diag}\left(\mathbf{1} + \frac{1}{\sqrt{l}}\sum_{i=1}^{l} h_i(b_i)\right). \tag{4}$$

In the generation process, when adding a watermark to the model for a given watermark code, we pass the watermark through the aforementioned mapper to obtain the embedding matrix $\mathbf{E}$. We then calculate the added model weights $\Delta\mathbf{W}$ and merge it into the original model weights by calculating $\mathbf{W}_{\text{watermarked}} = \mathbf{W} + \alpha\Delta\mathbf{W}$.

In the default setting, $\mathbf{I}_i$ is initialized from a standard normal distribution. However, to better maximize the differences between vectors, we propose using orthogonal initialization. Specifically, for any $\mathbf{I}_j$ and $\mathbf{I}_k$, where $j, k \in [0, l]$, we ensure that $\mathbf{I}_j \cdot \mathbf{I}_k = \mathbf{0}$. During model training, we fixed the mapper's weights, as fine-tuning them showed no significant improvement.

Experimental Settings

Datasets. We trained watermark encoders/decoders and fine-tuned protein generative models using the monomers from the PDB[41] dataset, focusing on proteins ranging in length from 60 to 512 residues with a resolution better than 5 Å. This initial dataset consisted of 23,913 proteins. Following previous work[37], we refined data by applying an additional filter to include only proteins with high secondary structure content. For each monomer, we used DSSP[42] to analyze secondary structures, excluding those with over 50% loops. This filtering process resulted in 20,312 proteins.

Implementations. Our FoldMark model is pretrained for 20 epochs and fine-tuned for 10 epochs with Adam[43] optimizer, where the learning rate is 0.0001, and the max batch size is 64. We use the batching strategy from FrameDiff[44] of combining proteins with the same length into the same batch to remove extraneous padding. In the watermark LoRA, the rank is set as 16 in the default setting. γ and η are set as 2. We report the results corresponding to the checkpoint with the best validation loss. It takes less than 48 hours to finish the whole training process on 1 Tesla A100 GPU. More hyperparameter settings are listed in Table. **??**.

Baselines. To the best of our knowledge, FoldMark is the first watermarking method specifically designed for protein structure generative models. For comparison, we adapted two state-of-the-art watermarking methods originally developed for image generation: WaDiff[29] [1] and AquaLoRA[30] [2]. Both baseline models were designed for image diffusion models, such as Stable Diffusion[19]. Since most protein generative models are also diffusion-based, we applied the recommended hyperparameters from the original works.

Data availability

This study's training and test data are available at Zenodo (https://github.com/zaixizhang/FoldMark). The project website for FoldMark is at http://home.ustc.edu.cn/~zaixi/projects/FoldMark.

Code availability

The source code of this study is freely available at GitHub (https://github.com/zaixizhang/FoldMark) to allow for replication of the results of this study.

References

1. Stokel-Walker, C. & Van Noorden, R. What chatgpt and generative ai mean for science. *Nature* **614**, 214–216 (2023).
2. Jumper, J. *et al.* Highly accurate protein structure prediction with alphafold. *Nature* **596**, 583–589 (2021).
3. Abramson, J. *et al.* Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature* 1–3 (2024).
4. Watson, J. L. *et al.* De novo design of protein structure and function with rfdiffusion. *Nature* **620**, 1089–1100 (2023).
5. Ingraham, J. B. *et al.* Illuminating protein space with a programmable generative model. *Nature* **623**, 1070–1078 (2023).
6. Committee, T. N. The nobel prize in chemistry 2024 - press release (2024). URL https://www.nobelprize.org/prizes/chemistry/2024/press-release/. Accessed: 2024-10-15.
7. Jo, A. The promise and peril of generative ai. *Nature* **614**, 214–216 (2023).
8. Epstein, Z. *et al.* Art and the science of generative ai. *Science* **380**, 1110–1111 (2023).
9. Meskó, B. & Topol, E. J. The imperative for regulatory oversight of large language models (or generative ai) in healthcare. *NPJ digital medicine* **6**, 120 (2023).
10. Alphafold terms of use (2024). URL https://alphafoldserver.com/terms. Accessed: 2024-10-15.
11. Bloomfield, D. *et al.* Ai and biosecurity: The need for governance. *Science* **385**, 831–833 (2024).
12. Baker, D. & Church, G. Protein design meets biosecurity (2024).
13. Wang, Y., Pruitt, R. N., Nuernberger, T. & Wang, Y. Evasion of plant immunity by microbial pathogens. *Nature Reviews Microbiology* **20**, 449–464 (2022).
14. Jurėnas, D., Fraikin, N., Goormaghtigh, F. & Van Melderen, L. Biology and evolution of bacterial toxin–antitoxin systems. *Nature Reviews Microbiology* **20**, 335–350 (2022).
15. Tan, C. W. *et al.* A sars-cov-2 surrogate virus neutralization test based on antibody-mediated blockage of ace2–spike protein–protein interaction. *Nature biotechnology* **38**, 1073–1078 (2020).
16. Bergman, A. S. *et al.* Guiding the release of safer e2e conversational ai through value sensitive design. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue* (Association for Computational Linguistics, 2022).
17. Mitchell, E., Lee, Y., Khazatsky, A., Manning, C. D. & Finn, C. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning*, 24950–24962 (PMLR, 2023).
18. Wu, J., Guo, J. & Hooi, B. Fake news in sheep's clothing: Robust fake news detection against llm-empowered style attacks. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3367–3378 (2024).
19. Rombach, R., Blattmann, A., Lorenz, D., Esser, P. & Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695 (2022).

[1] https://github.com/rmin2000/WaDiff
[2] https://github.com/Georgefwt/AquaLoRA

20. Betker, J. *et al.* Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf* **2**, 8 (2023).

21. Westerlund, M. The emergence of deepfake technology: A review. *Technology innovation management review* **9** (2019).

22. Zhang, T. Deepfake generation and detection, a survey. *Multimedia Tools and Applications* **81**, 6259–6276 (2022).

23. Kirchenbauer, J. *et al.* A watermark for large language models. In *International Conference on Machine Learning*, 17061–17084 (PMLR, 2023).

24. Liu, A. *et al.* An unforgeable publicly verifiable watermark for large language models. In *The Twelfth International Conference on Learning Representations* (2023).

25. Zhang, R., Hussain, S. S., Neekhara, P. & Koushanfar, F. {REMARK-LLM}: A robust and efficient watermarking framework for generative large language models. In *33rd USENIX Security Symposium (USENIX Security 24)*, 1813–1830 (2024).

26. Liu, A. *et al.* A survey of text watermarking in the era of large language models. *ACM Computing Surveys* (2024).

27. Dathathri, S. *et al.* Scalable watermarking for identifying large language model outputs. *Nature* **634**, 818–823 (2024). URL https://doi.org/10.1038/s41586-024-08025-4.

28. Fernandez, P., Couairon, G., Jégou, H., Douze, M. & Furon, T. The stable signature: Rooting watermarks in latent diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 22466–22477 (2023).

29. Min, R., Li, S., Chen, H. & Cheng, M. A watermark-conditioned diffusion model for ip protection. *ECCV* (2024).

30. Feng, W. *et al.* Aqualora: Toward white-box protection for customized stable diffusion models via watermark lora. *arXiv preprint arXiv:2405.11135* (2024).

31. Yang, Z. *et al.* Gaussian shading: Provable performance-lossless image watermarking for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12162–12171 (2024).

32. Liu, Y. *et al.* De novo protein design with a denoising diffusion network independent of pretrained structure prediction models. *Nature Methods* 1–10 (2024).

33. Van Kempen, M. *et al.* Fast and accurate protein structure search with foldseek. *Nature biotechnology* **42**, 243–246 (2024).

34. Lu, W. *et al.* Dynamicbind: Predicting ligand-specific protein-ligand complex structure with a deep equivariant generative model. *Nature Communications* **15**, 1071 (2024).

35. Hu, E. J. *et al.* Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685* (2021).

36. Bose, A. J. *et al.* Se(3)-stochastic flow matching for protein backbone generation. In *The International Conference on Learning Representations (ICLR)* (2024).

37. Yim, J. *et al.* Se (3) diffusion model with application to protein backbone generation. *ICML* (2023).

38. Yim, J. *et al.* Fast protein backbone generation with se (3) flow matching. *arXiv preprint arXiv:2310.05297* (2023).

39. Campbell, A., Yim, J., Barzilay, R., Rainforth, T. & Jaakkola, T. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. *ICML* (2024).

40. Lin, Z. *et al.* Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **379**, 1123–1130 (2023).

41. Berman, H. M. *et al.* The protein data bank. *Nucleic acids research* **28**, 235–242 (2000).

42. Kabsch, W. & Sander, C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers: Original Research on Biomolecules* **22**, 2577–2637 (1983).

43. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

44. Yim, J. *et al.* Improved motif-scaffolding with se (3) flow matching. *arXiv preprint arXiv:2401.04082* (2024).

Author contributions statement

Z.X.Z., M.Z., and M.D.W. designed the research, Z.X.Z. and K.D.F. conducted the experiments, Z.X.Z., L.C, M.Z., and M.D.W. analyzed the results. Z.X.Z., R.F.J., L.C., M.Z., and M.D.W. wrote the manuscript. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Mengdi Wang.

Supplementary Information

Ablation studies of FoldMark are here.
Watermark Encoder-Decoder architecture