

Function-preserving watermarking of AI-generated proteins

Corresponding Author: Dr David Stutz

Any redactions in this file are there to maintain patient confidentiality, the confidentiality of unpublished data, or to remove third-party material.

This file contains all reviewer reports in order by version, followed by all author rebuttals in order by version.

Attachments originally included by the reviewers as part of their assessment can be found at the end of this file.

Version 0:

Reviewer comments:

Referee #1

(Remarks to the Author)

The manuscript by Stutz et al. describes approaches to watermark either protein sequences generated by proteinMPNN or protein structures as predicted by AlphaFold 3. Both approaches are validated in silico to assess detectability and robustness. The sequence watermarking approach is also validated through in vitro validation of resequenced watermarked binders, showing that the watermarking procedure does not negatively impact the function of the proteins.

The authors chose not to provide any code, so any comments below are solely based on the data and description provided in the text, figures, and supplementary information. The underlying theory and fundamental approach are explained well and seem sound to me, but a detailed evaluation of the computational work would require more information. The authors state that code will be made available at time of publication, but for a manuscript that focuses on computational development the code is generally required in the review process to evaluate what has been done.

My main concerns regard the intended use cases, the “robustness to attacks” as the authors call it, and the implications of the true- and false positive rates in any real-world scenario.

Main concerns:

The authors motivate this work through use cases in biosecurity and research integrity. However, I fail to understand how the proposed watermarking approach could help in either case.

For biosecurity (section 5.1), the authors refer to the digital-to-physical frontier and DNA synthesis companies' conditions. How do the authors envision this would work? How would identifying an MPNN-generated sequence help distinguish what should or should not be synthesized? Would DNA synthesis companies refuse to synthesize designed sequences?

There are already many different models for sequence design and structure prediction, and the number keeps growing every week. Would the authors expect that every newly developed model adopts the watermarking strategy? Wouldn't a single model that doesn't use watermarking ruin the entire system? How would this be enforced? I understand that this paper is about technology and not about policy, but I simply do not understand how this would be helpful.

The authors suggest that “watermarking could be used to reliably establish which specific tools were used for design”. However, the structure-based watermark would only give information about the very last tool that was used in the pipeline, since all structural information from earlier steps would be lost in the last sequence-to-structure prediction step; meaning that in atypical RFDiffusion - proteinMPNN - AlphaFold pipeline, the initial backbone generation step would be lost.

In section 5.2, the authors suggest that watermarking with SynthIDBio could be used to prevent submission of AI-generated sequences or structures to databases. While this sounds like a great safeguard, I am concerned about the implications of the false positive rates (FPR) here. Throughout the manuscript, the authors set a target FPR of 0.1%, which sounds low. However, given the sheer size of the databases involved, this seems orders of magnitude too high. Given a 0.1% FPR,

about 244 experimental structures from the PDB would be flagged as AI-generated, and 200 human protein sequences from UniProt would be flagged as designed with proteinMPNN. Assuming multiple different generative models, prediction networks and sequence design tools all used independent watermarking / separate watermarking keys, I would expect that this number would increase near-linearly with the number of watermarking keys used. Would this not be prohibitive for any intended use case? This should be addressed, ideally by including some kind of benchmark on natural sequences and structures.

Overall, the authors focus on distinguishing watermarked proteinMPNN-generated sequences from non-watermarked proteinMPNN-generated sequences. Wouldn't most applications (besides IP protection) rather require distinguishing artificially generated sequences from natural sequences? How distinguishable are natural and designed sequences without any watermarking? Would proteinMPNN-generated sequences have some distinguishable properties already?

Regarding true positive rates (TPR), it would be great to have some kind of discussion about what level of TPR would be required for specific use cases (how good does it need to be in order to be useful).

Regarding the robustness of SynthIDBio-sequence, the authors consider different levels of sophistication and information. Given that proteinMPNN and AlphaFold are freely available via browser-based interfaces, I would suggest assuming that any attacker would have access to these tools. Using resequencing with proteinMPNN, it seems very easy to remove watermarks, and the only drawback would be a reduced success rate. But what if the attacker only resequences parts of the protein? For a binder, one could expect that resequencing only the non-interface residues would lead to a significantly higher success rate (=preserve the function); would this also be enough to remove the watermark? What if the attacker adds residues to either terminus of the protein that don't interfere with the function?

For SynthIDBio-structure, the authors state that the watermarking is robust to adding 0.1 Angstrom Gaussian noise. How about other ways to modify a protein structure model? What if an attacker uses for example Rosetta or a similar tool to relax the structure in torsional and/or Cartesian space? Ideally, I would have loved to give this a try, but unfortunately the authors chose not to provide any code.

The in vitro work looks sound to me, and convincingly shows that sequence watermarking does not affect binding properties in the chosen setting, although once again very little detail is provided:

- Ideally, sequences (and pdb files) for all tested designs should be provided.
- Was there any type of sequence validation after cloning/assembly?
- Was there any type of purification or quality control of the final proteins?
- Negative controls are mentioned in the text and methods section, but no result is ever reported for them.
- Raw data or at least plots of fitted data should be provided for SPR measurements.

Referee #2

(Remarks to the Author)

In this manuscript, the authors present two potential methods for watermarking AI-generated designs. The first operates by distorting the transition probabilities in protein sequence generation, the second operates by distorting the positional relations between atoms in a 3D structure model. Understanding the provenance of AI-generated biological artifacts (digital or physical), is indeed a pressing problem, and the key value of the methods proposed by the authors is that they embed information into the nature of the artifact itself.

What is not currently clear from the manuscript, however, is the cost/benefit tradeoff for doing so. The authors clearly demonstrate that in some circumstances their methods enable detection, and that in some circumstances the distortion introduced by their methods is not significant. Unfortunately, however, the current presentation in the manuscript made it difficult for me to tell whether those circumstances overlap enough to make the methods valuable. Instead, it appears to me that the size of distortion required for reliable detection may necessarily result in a major degradation of tool functionality. I may be incorrect about that, but I think it will require significant improvements in the manuscript to make that clear.

For SynthIDBio-sequence, the tradeoff between false positive rate, true positive rate, and tool performance does not look very good to me:

- If one is looking to see if any protein in a cellular organism is watermarked, there would typically be at least hundreds or thousands of proteins to consider, which means that even the 0.1% false positive rate is fairly high. The higher rates considered by the authors are essentially guaranteeing any engineered protein would be lost against a background of false positives. For checking the integrity of submissions to scientific databases, the false positive rates would need to be even lower.
- For 0.1% false positives, the true positive rates are extremely low for normal operating temperatures (0.1 - 0.3) and not great even at 0.5. Only with distorted sampling does the detection become really reliable.
- Fig. S9, on the other hand, makes it quite clear that the distorted sampling approach results in a dramatic reduction in sequence diversity, though this is not quantified.

The diversity reductions from SynthIDBio-sequence are not quantified - they are only measured indirectly through the distribution of K_D values and AlphaFold confidence values. In Section 9.5, the standard deviation of log K_D

measurements is said to be 2.5, which means that there is not much power to discriminate on a range of -10.5 to -5.5. The distribution of K_D measurements shown in Fig. S2 and the improvement rates shown in Fig. S3 thus seem to be primarily driven by the uncertainty in the in vitro assay and reflect little about the properties of the actual distribution.

I thus think that it would be extremely important to quantify the direct effect on sequence diversity, e.g., through entropy measures of the distribution over residues. Given what has been presented, I suspect that there will be quite a significant decrease in entropy for the distorted sampling, which calls the viability of the watermarking approach into doubt.

Certainly, however, the evidence presented does not support the authors' claims of "near-perfect detectability" for anything but distorted distributions.

For SynthIDBio-structure, the big question, which is not adequately addressed by the current presentation, is the degree of impact that the distortion has on the validity of model results. Using perturbations for SynthIDBio-structure range of up to 0.1 Angstroms is quite high, given that the distance between many atoms is on the order of only 1-3 Angstroms. How big an impact is this having on model quality?

Fig. 4b and Fig. S28e show that even with the minimum studied perturbation of $S=0.001$, there are quite significant changes in the distribution of feature values with respect to non-distorted AF3. Distributions for higher levels are not presented, and need to be. The summary numbers in Fig. S28d, however, already show that some properties change by an order of magnitude for $S=0.1$, which seems to me to throw the validity of the watermarked model into grave doubt.

I think it would be extremely useful to see the ROC curves not just for the case of structures perturbed with 0.1 Angstrom noise, but also for the cases that are producing graphs 4a, S28a and S28b, as well as for 0.01 Angstrom and 0.001 Angstrom noise. As it is, between the lack of an ROC curve and the figure issues detailed below, the significance of the detectability results is hard for me to evaluate.

Other significant issues:

- The cost of watermarking is not discussed:
- How much additional compute is required for design generation for SynthIDBio-sequence? Multiplying the number of designs by 2^{25} is not a small cost.
- How large are the models that need to be stored for each key for SynthIDBio-sequence?
- How much compute is required for fine-tuning the model for SynthIDBio-structure? What is the tradeoff between amount of fine tuning, performance, and cost?
- A k-mer length of $H=4$ seems quite low for ProteinMPNN - aren't k-mer lengths typically significantly longer than that? Using such a low length seems likely to have a significant impact on costs, detection, and design viability.
- The binders used for testing are all short proteins. What happens if the proteins are hundreds of amino acids long, like most natural proteins?
- Fig. 3d shows the impact of resequencing on hit rate, but we do not see the impact of resequencing on the ability to detect the watermark. My guess is that the watermark is effectively destroyed after resequencing.
- It is counterintuitive and potentially concerning that random perturbation is **required** in order to prevent SynthIDBio-structure watermarking from being destroyed by fixed point rounding.
- Only a single instance of training SynthIDBio-structure is presented. If fine-tuning is performed more than ones, to what degree will different instances be distinguishable? Multiple instances should be trained and compared in order to determine the cross-instance false positive rates.
- I do not think I understand the value that is being proposed for SynthIDBio-structure: how does knowing the tool used to generate a publicly shared molecular model help with either biosecurity or scientific integrity? "I used AlphaFold3 to make this model" seems like a statement that nobody would need to try to hide.

Minor points:

- Fig. 1a: The watermark detector is more than just a threshold - it needs sampling data and the key that was used for generation, which needs to be transmitted between the generator and detector.
- Fig. 1b: Shouldn't the two watermark detectors be the same color?
- Fig. 1b seems to conflate together the model training and model application stages. If I understood correctly, a model user should not have access to the watermark detector module, else they could use it to undo the fine tuning.
- Fig. 2a,e would be much clearer if presented as violin plots.
- Fig. 2c: Is the third line meant to be distortionary 0.1, rather than non-distortionary 0.5? It looks like it's meant to show one of each case. If you're showing multiple distortionary 0.5, there is vertical space enough to show more than 2.

- The "Detectability and accuracy" graphs in Figs. 4a and S28a are extremely difficult to read and do not appear to match one another. Separate the two metrics into different panels, use consistent axes, plot with more readily distinguishable colors, and use line graphs instead of bar graphs.
- Fig. S28a is missing the bars for 0.001 and 0.01 noise.
- Figs 4a, S28a, and S28b would be much clearer if they were converted to line graphs with the number of AAs on the x-axis.
- Figs 4b and S28e should plot distributions with lines rather than semi-opaque areas so that it's easier to see the actual values.
- The detectability rates shown in Figs. 4a (right) and S28b do not appear to be compatible with what is shown in 4a (left) and S28a. Some of that may be due to the problems in the presentation of these graphs, however.

Referee #3

(Remarks to the Author)

Overall, I think this is a well-designed study on a timely topic. There are, however, several areas where the authors could provide further explanation and context (see list below). I would also encourage the authors to consider reframing the way they present the study:

Biosecurity is an increasingly discussed topic with many proposed interventions, yet none that are 100% foolproof; and that is okay. Each intervention has its strengths and weaknesses, but the cumulative effect of layering a set of flawed guardrails can lead to a resilient system. I think it is important to frame any biosecurity intervention in this context; so, rather than focusing primarily on where SynthIDBio works well, equal attention should be given to where it does not. In other words, discussion should provide more focused discussion on the specific elements of the biosecurity environment that SynthIDBio both does and does not support, and, further, how the method fits together with other proposed interventions. I think that Section 5.1 provides an excellent starting point for this type of reframing, where a potential synergism between watermarking and DNA synthesis screening is proposed. The paper would be strengthened if use cases for watermarking (like the one in the aforementioned section) were more richly worked through, with a focus on realism and feasibility (i.e., How realistic is the use case in question?) and robustness (i.e., How resilient would watermarking be in the proposed usage?).

Within the context of that overarching comment, here are some specific thoughts on topics to include/expand upon:

Most bio design tools (BDTs) are open source, including, of course, ProteinMPNN, which was the BDT primarily used for testing SynthIDBio-sequence. How do you implement watermarking in such an environment? A malicious actor would likely just use an open-source tool that did not use watermarking.

Similar to the above, a sentence or two on the generalizability of SynthIDBio should be added. As written, it appears to me that SynthIDBio is not as easy to use as simply wrapping an existing model implementation with a watermarking implementation—some level of modification to underlying code and, at least in the case of the structure version of the tool, model fine tuning will be required. Is this correct? If so, some discussion regarding the challenges of encouraging/promoting uptake should be added, as I would imagine many (if not most) tool developers will not be motivated to do the extra work.

It is mentioned in the text that SynthIDBio introduces additional overhead to the generation pipeline, but this is not quantified anywhere. Ideally, quantification via a few metrics should be added, including GPU hours and monetary cost. I think that Figure 3B gets close to this, as it shows the decrease in pass rates when designs are filtered by g-value as well. Perhaps a companion figure could show the increase in cost or compute required to achieve the same rate of passing designs when using g-value filtering versus when not.

The sections on resilience to watermark removal need to be expanded:

In Section 4.3, information should be provided on how the resequencing attack was performed. Additional discussion should be added on how easy (or not) this would be to do. My sense is that anyone capable of using BDTs would be able to launch such an attack.

For Section 4.5, attacks using a relaxation protocol or some other energy minimizing protocol should be benchmarked. As of now, there is only brief mention in the discussion of how the authors expect such an attack to remove the watermark. Assuming the authors' hypothesis regarding these attacks is true, discussion over what this means within the broader context of biosecurity should be added.

I also have a few more minor comments:

Related work on watermarking biological designs is introduced in the discussion ("Concurrent work also considers watermarking..."), when it would more appropriately be first referenced in the introduction. The second paragraph of the Introduction seems a reasonable place to put this.

On pg. 10, there is the statement "However, the examples shown in B are meant to illustrate the worst case." In what sense

are these examples meant to illustrate the worst case? More explanation is needed.

At the top of pg. 11, there is the statement "It is also worth emphasizing that, by construction of our watermark detector, the watermark is robust to rigid transformations of the protein structure." A supplementary figure should be added to support this claim.

Referee #4

(Remarks to the Author)

I co-reviewed this manuscript with one of the reviewers who provided the listed reports.

Version 1:

Reviewer comments:

Referee #1

(Remarks to the Author)

The authors have substantially edited and appended the manuscript to address the concerns expressed by all reviewers, and have also shared code and (raw) data as requested by myself and other reviewers. I also appreciate the effort the authors have put in highlighting all edits and overall presenting a very clear and concise rebuttal.

The technical concerns raised by the reviewers have in principle been addressed. In some cases the revised manuscript clearly acknowledges weaknesses and shortcomings of the proposed procedure, which is commendable. However, this ultimately contributes to the impression that the procedure as presented here is not really useful (Except for protecting IP, which is not the focus of the manuscript). The authors themselves state in the manuscript:

"Ultimately, however, we consider this a proof-of-concept that function preserving watermarking of protein sequences and structures is possible."

This statement is undeniably true, and proof-of-concept studies are undeniably valuable contributions to science. Whether this proof-of-concept is enough to justify publication in Nature is for the editor to decide, but I would recommend treating the manuscript as exactly this, a proof-of-concept, and not a ready-to-use method (as one might have expected from these authors).

Detailed response to the rebuttal:

My (and other reviewers) concerns about FPR and TPR tradeoffs have been addressed convincingly.

The expanded description of the methods and the shared data have convincingly addressed my concerns regarding the in vitro work.

The detailed analysis of costs arising from watermarking is convincing and very much appreciated.

Other reviewers and I had been concerned that the original manuscript overestimated the robustness of watermarking against removal attempts in the form of resequencing (SynthIDBio-sequence) or relaxation (SynthIDBio-structure). The authors now acknowledge the lack of robustness in both cases, which is commendable. However, this rules out all potential use cases which would require watermarks to be irremovable (such as the cases discussed in section 5.2 of the manuscript).

Which leaves the question of how and where these watermarking procedures could be applied (except for protecting IP). The authors have amended the manuscript to address this question and expanded on it in the rebuttal, but I honestly still don't understand the underlying logic.

The authors state in the rebuttal:

"We believe that building good incentives for users to use "trusted", watermarked models and these models being state-of-the-art can go a long way in driving adoption."

What is a "trusted" model, and why should a DNA synthesis company prefer sequences obtained from a trusted model? If a malicious actor uses proteinMPNN to resequence toxins or pathogens or any "harmful" protein, why would it matter if a "trusted" version was used? How does identifying the specific resequencing tool help to distinguish between harmful and not harmful? I apologize if I am missing something obvious here, but I cannot think of a single concrete example where this identification of the resequencing tool would help to decide what should or should not be synthesized. Section 5.1 and the corresponding parts of the rebuttal sound as if the authors had a specific scenario in mind - maybe it would help to describe this in detail?

(Remarks on code availability)

I did not conduct an in-depth review of the code but from what I saw I have no complaints (only that I wish it had been provided with the initial submission).

Referee #2

(Remarks to the Author)

Thank you to the authors for the work that they have done clarifying figures and adding new information to fill in important gaps. The new version of the manuscript is significantly clearer and more convincing in a number of aspects, and a number of my concerns regarding the first version have been well addressed. Among other things, the cost charts are excellent: they give clear intuitions of tradeoffs and bottlenecks, and address all of the concerns that I previously had in that area.

Some major open questions remain, however, which I believe need to be addressed in order to determine the significance and correctness of the results in the manuscript.

Biosecurity Value of Sequence Watermarking

The authors have proposed to flip the value here to focus on watermarks as indicators of production by a trusted tool. Since such detection already requires secret key exchange, and since users of a trusted tool are incentivized to provide evidence that they should be trusted, why does sequence watermarking have any advantage over standard PKI approaches like certificate-based signing of a digest value in accompanying metadata?

Biosecurity Value of Structure Watermarking

The authors claim that watermarking will help to identify whether crystallography was actually carried out (versus just in silico work). I still do not understand how this would be true. Crystallography results would not have a watermark. Non-trusted tools produce no watermark and relatively small random perturbations destroy a watermark, so how are either of those cases distinguishable from real crystallography?

Entropy / Sequence Diversity:

The new entropy and diversity results are quite interesting and contain some surprising and counterintuitive results. I very much appreciate the authors' exploration of this space.

That said, I find the results in Figs S10 and S11 concerning. The key point I see is that distortionary 0.1 results are almost entirely non-conserved with respect to the original sequence and highly conserved with respect to one another. This, in turn, I believe means that the entropy results in Figs S8 and S9 are telling us very different things about the non-watermark and distortionary 0.1 distributions. For non-watermark, we are seeing exploration of protein space around the original sequences, while for distortionary 0.1, we are seeing exploration of protein space around a very different consensus sequence. This view is also consistent with the example sequences shown in Fig. S21.

This leads me to wonder the degree to which ProteinMPNN is even remaining a significant aspect of the design, versus the watermarking effectively replacing ProteinMPNN with a different diversity generation technique in which the primary determinant of sequence variation is the randomly chosen encryption key. What would happen if ProteinMPNN was replaced with a different tool, such as EvoDiff, RFDiffusion, but the watermarking key was kept the same? Would the difference between watermarked outputs of different tools be less than the difference between watermarked and non-watermarked outputs from the same tool?

A milder form of the same concern applies for non-distortionary 0.5.

Sequence Watermarking False Positives, False Negatives, and Natural Sequences:

The results in Figs S2 and S3 showing false positive rates against natural sequences are extremely helpful, and I thank the authors for including them. Order of magnitude lower false positive rates is very good news as a starting point.

Unfortunately, however, as the authors also note (correctly, I believe) a significant component of this lower FPR is due to the longer sequences. The authors also correctly note that adding adjacent non-watermarked material would readily spoof the system, and that for this reason it would be important to compare not whole sequences but windows.

What would happen to Figs S2 and S3 if the test were performed as an "apples to apples" comparison using a scanning window? Taking a window of length 60, matching the length of some of the engineered proteins worked with by the authors, and a 50% tiling (a bare minimum requirement for screening), a typical 400 amino acid natural protein would be tested on 13 different windows, and return a false positive if any of those 13 had a high g-value. Combine that with the fact that shorter windows will produce a broader g-value distribution, and the FPR may end up as bad or worse than the target.

I think that this is an important thing to check, if the authors wish to know what the FPR/TPR tradeoff is likely to actually be in the scenarios they envision.

Impact of Structure Watermarking on Distributions:

The authors claim that the impact of structure watermarking on accuracy is not high because Figs 4a and S41b show that the RMSD values versus ground truth are similar for both watermarked and non-watermarked structures. Figs 4b and S42-S45, however, show that in many cases these distributions are highly bimodal, and in some cases the watermarking strongly distorts or eliminates a mode relative to both ground truth and non-watermarked. Is systematic elimination of a mode of distribution really negligible in impact on accuracy, even if bulk statistics are not much affected? Perhaps these are critical points in the protein. It would be interesting to inspect a few of the affected atoms and see if there is a high impact on local regions of the structure, even if the overall structure is relatively intact. The fact that the authors acknowledge that WM-0.1 has a high impact on distributions but that the bulk metrics show barely any difference between WM values indicates that those bulk metrics are not telling the full story on accuracy.

Differentiability of Structure Watermarks:

The authors appear to have misunderstood my question about different instances of structure watermarking, as they have compared their different. My question is about what happens if we want to distinguish more than one "trusted tool" by watermark. Let's say that there are five different organizations that each run a "trusted AF3" tool, each of which has been independently fine-tuned with $s=0.001$. What does the confusion matrix look like for correctly interpreting a watermark as having been applied to Organization A vs Organization B vs Organization C, etc.?

Minor points:

- For structure, some places say "s" and some way "WM". I think they are the same, but I found it confusing.
- The s value for Fig. 4b is not given.
- Figs 3a,b, S35, S37, S38, S40, S41a: Please do not use nearly-identical shades of the same color to distinguish different conditions. There are lots of good color-blind-friendly options that will make these figures much easier to read.
- Figs 3b, S40b: Put a gap between the bar clusters. They are running into one another right now.
- Fig. S41a: The bars are out of order. Clean is equivalent to 0, so the others should be in ascending order: 0.001, 0.01, 0.1, not the current non-monotonic order.
- Fig. S41f is very interesting! I'd like to understand better what it's telling us, but the graphs and text say there is noise=0.1 added and the captions say there is no noise added. Which is it?
- Fig. S41f: Order the WM values in the caption - right now they are out of order.

(Remarks on code availability)

Referee #3

(Remarks to the Author)

We thank the authors for taking the time to address our concerns. The proposal for motivating adoption seems reasonable as a method adapting methods developed with language models to protein biology. We appreciate the more nuanced discussion regarding where SynthIDBio fits into the broader biosecurity ecosystem. Given the changes made, we feel that the work is suitable for publication. Before that, however, we want to suggest an important update. This change should not preclude publication, but addressing the following would increase the impact of the work:

The "trusted model" system that is proposed is a good idea, but it of course relies on DNA synthesis providers all agreeing to adopt a policy of preferential treatment of "trusted" orders. This should not be too difficult to achieve, so long as use of watermarking techniques does not significantly burden their customers. What would be most burdensome for customers would be a lack of options: Customers will want to be able to select from a wide array of "trusted" models. There can be wide availability of trusted models only if developers are encouraged to include SynthIDBio or similar as part of their models/generation procedures. This should also not be too difficult to get model developers to do, so long as it is easy to implement SynthIDBio as part of their projects. Thus, the impact of this work is maximized if it is SynthIDBio is made as easy as possible for developers to incorporate. Given this, the SynthIDBio code base should include the infrastructure needed to apply SynthIDBio to tools beyond those studied here. For SynthIDBio-sequence, this would ideally take the form of a class decorator, a wrapper function, a base class, etc.--something that can be dropped in to anyone else's project to apply watermarking to their model with next-to-no additional effort. This would admittedly be more difficult to do for SynthIDBio-structure, owing to the requirement for retraining, but programmatic infrastructure could still be provided to streamline its application to others' models (though we would imagine that more would be needed than, e.g., a simple decorator).

(Remarks on code availability)

Referee #4

(Remarks to the Author)

I co-reviewed this manuscript with one of the reviewers who provided the listed reports.

(Remarks on code availability)

Version 2:

Reviewer comments:

Referee #2

(Remarks to the Author)

Thank you to the authors for their continued work to address concerns regarding the manuscript.

The additional entropy/sequence diversity analysis is useful, and while I am not entirely convinced, I think there is enough detail now to allow a reader to make an informed judgement on how to interpret the results. I also appreciate the authors acknowledging that structure watermarks currently cannot differentiate between providers.

A number of the key issues in the prior version, however, still have not been addressed, leaving the significance and correctness of claims in the manuscript still in doubt in its current form.

****On biosecurity value of sequence watermarking:****

The authors argue the watermarking has advantages over PKI-based signatures for indicating authentic production of a sequence by a trustworthy source. Unfortunately, their arguments appear to confuse signatures (which are used for establishing authenticity) and encryption (which is used to avoid sharing information).

Specifically:

- "metadata is trivial to remove"

Lack of signature metadata = untrustworthy, so this doesn't impact use of signatures to establish authenticity.

- "encryption also requires coordination."

Signatures are not encryption, and are typically established unilaterally.

- "While public-key encryption makes this easier, future advancements in watermarking could enable similar strategies (as discussed in Section 5.3)."

It is not clear to me what the authors are referring to here. The only portion of Section 5.3 that references public-key methods is talking about the fact that detectors currently cannot be shared publicly, which is a significant deficiency for watermarking compared to PKI-based signatures.

- "Even if the tool user is incentivized to provide encrypted metadata ... C2PA is not reliable and commonly removed by many social media sites ... this would require all other tools used in the process to respect the metadata protocol."

The architectural design flaws in C2PA do not imply that signatures don't work; they only mean that C2PA was not designed effectively for establishing authenticity.

If a sequence or portion of a sequence was not modified, signature metadata will still be able to authenticate its provenance (even if delivered separately, as is commonly done with software).

If subsequent non-trusted tools have modified the sequence, however, then supply chain integrity has indeed been broken and the sequence should NOT be trusted.

- "A watermark is not as easily removed."

Again, lack of signature = untrustworthy, so this isn't a problem for using signatures to establish authenticity.

Moreover, in Section 4.3, the authors report that ProteinMPNN resequencing "effectively removes the watermark."

- "Tracking edits in metadata is also more involved, while watermarking can naturally be applied to parts of a sequence/structure (just like it is applied to parts of edited images [3])."

Signature metadata can easily be applied to edits or specific components as well: this is commonly done with security patches and software libraries.

In sum, it seems that, excepting the dubious case of C2PA, the authors are not familiar with how PKI-based signatures are typically actually employed in current practice for establishing authenticity. So far as I can tell, sequence watermarking with methods like SynthIDBio-sequence remains strictly inferior to PKI-based signatures for establishing authentic production of a sequence by a trustworthy source.

****On the value of structure watermarking****

The authors claim that: "SynthIDBio-structure has been shown to be robust to random perturbations, as shown in Figure 4a and the appendix."

I do not see how Fig. S45 supports this claim.

The results presented in S45 show that when random perturbations are equal to or greater than the value of s used in training, detectability becomes extremely degraded. An adversary choosing to use a random perturbation of 0.2A (still much shorter than a typical protein bond length) should thus expect to be able to remove the watermark even at $s=0.1$.

The authors claim: "With growing adoption, this additional database signal will get clearer and more reliable." But honestly generated in silico work is already generally correctly labelled by methodology, and dishonest work can readily remove the watermark according to Fig. S45.

****Data integrity problems in SynthIDBio-structure figures****

While evaluating the authors' claims about the robustness of SynthIDBio-structure to random perturbations, I found that there currently appear to be a number of inconsistencies in Fig. 4a and Fig. S45.

First, there appears to be an inconsistency between Fig. 4a(middle) and Fig. S45a. Fig. 4a(middle) is a subset of the data in Fig. S45a, so the numbers on 4a(middle) should be the same as those on equivalent bars in S45a. But:

- WM-0.001 / Noise 0.01A is 99.8% in 4a(middle) and 99.9% in S45a.

- WM-0.1 / Noise 0.01A is 99.9% in 4a(middle) and 99.8% in S45a.

This isn't a very big difference, but it's enough to indicate that something may have gone wrong in plotting, particularly given the claim in Section 4.5 that " $s=0.1$ still achieves a TPR of 88.5%", while S45a indicates that 88.5% is the value achieved by WM-0.001.

Likewise, the Noise 0.1 values in S45a should correspond to the value at FPR=0.1 in S45d. But WM-0.1 has TPR=88.5% or 0.6% in S45a and ~100% in S45d.

Fig. S45c also does not match with S45a or S45f. The authors do not label which of the panels in S45c is WM-0.01 and which is WM-0.1, and also do not say the amount of noise used in S45c. By order of the description, a reader would typically presume WM-0.01 is the one on the left, but by results it would appear that WM-0.1 is the one on the left. Whatever the case may be, however, the results span a range where the weighted means cannot match either the better performance claimed in S45a or the worse performance observed in S45f.

****Sequence watermarking false positives, false negatives, and natural sequences****

The authors appear to still be fundamentally misunderstanding the requirement for window-based detection, so I will attempt to explain again. The authors find, as predicted, that thresholds must be adjusted to achieve their target false positive rates over an `_individual_` 60-residue subsequence.

However, the authors dismiss detection based on the maximum value in any window because it: "requires careful recalibration of the background null distribution to account for the multiple hypothesis testing inherent in evaluating numerous windows per sequence."

That is exactly the point!

As the authors themselves note in Section 5.1, any approach other than maximum value means that adding flanking material "will reduce the watermarking signal proportionally based on the modification's relative size."

Consider the following construct: GFP-2A-XXX-2A-GFP, where GFP is green fluorescent protein, 2A is the standard self-cleaving peptide, and XXX is any watermarked protein shorter than GFP. This construct will produce protein XXX, but unless the max-window approach is used, XXX cannot be detected except via false positives on GFP.

Side note: the authors also do not seem to find the threshold shift from 0.679 to 1.000 notable, even though 1.000 is the maximum possible value of the system. A parameter saturating like that typically indicates a qualitative change in behavior (usually failure/loss of system control).

****Impact of structure watermarking on distributions****

The authors have fundamentally failed to address the concerns that I expressed regarding potential high impact on structure accuracy at $s=0.1$. The new supplementary material addresses only $s=0.001$, which the authors have already agreed is not

robust against small watermark removal perturbations.

****Presentation notes****

- If you are going to have multipanel figures, give a panel label to each distinct dataset in the figure and be sure you label which portion is which.

- The fact that the "s" vs. "WM" split has still not been addressed despite the authors claiming to have done so made reviewing SynthDBio-structure significantly more frustrating.

(Remarks on code availability)

Referee #3

(Remarks to the Author)

With the addition of the wrapper function/class decorator, I believe the utility of this tool is notably increased. I attempted to review the additions to the codebase, but it appears that the resource made available to reviewers has not been updated to reflect them. So long as it is confirmed that such changes have been made to the repository, my concerns have been addressed.

As a note, I agree with the shortcomings noted by other reviewers, but shortcomings are to be expected for any biosecurity intervention. Biosecurity, particularly in the face of AI advances, will ultimately have to rely on a set of "leaky" interventions that, when combined, yield a more robust system. In other words, I agree with the shortcomings noted by the other reviewers, but shortcomings are to be expected for any biosecurity intervention. Perhaps the authors can add additional remarks on this perspective--that the methods are indeed not bulletproof; they have their shortcomings, but that the methodology can be added to the constellation of methods with the hope that they can help boost robustness and can help with a "defense in depth" approach to the challenges ahead.

(Remarks on code availability)

Referee #4

(Remarks to the Author)

I co-reviewed this manuscript with one of the reviewers who provided the listed reports.

(Remarks on code availability)

Referee #5

(Remarks to the Author)

I recommend this work for publication. To my knowledge it is the first study to demonstrate that watermarking engineered protein sequences need not come at the cost of function. The study is well-designed and rigorous and the methods are thoughtful extensions of established work on provenance in general text generation.

The authors demonstrate this across both sequence design and structural prediction, showing wide applicability and a path toward hard-wiring watermarks directly into model output. Whether protein watermarking could survive without degrading functional prediction had remained an open question. Publishing this result would establish an initial baseline while leaving many avenues open for improvement. A beachhead implementation like this can inspire far more researchers to attack its weaknesses and build on its strengths.

I understand the concern that the approach is not production-ready, but that is not generally a standard to which we hold academic papers for publication. That said, the gene synthesis community has long wrestled with (and still relies on) methods that carry well-known avenues for attack or subversion. We are under no illusion that we can absolutely rule out misuse; our goal is to make it as difficult as possible. The SynthDBio methods fall squarely in this tradition. Yes, the watermark can be defeated, but defeating it requires additional knowledge and tool use, raises computational cost, and risks degrading the function of the protein output. And should top-tier models converge on a consistent watermarking strategy, it may eventually push bad actors toward less performant models. That trade-off of increasing difficulty rather than perfect security is one the DNA synthesis industry knows intimately.

Were this method deployed in protein-design models, DNA synthesis providers would gain real value from screening submissions for the watermark. An intact mark tells us two useful things at once: that a sequence region was engineered, and the customer did not remove the watermark. This is affirmative (but not conclusive) evidence they are not trying to disguise its origin. Watermark absence proves nothing, for the reasons the authors provide. But detection is cheap, and when detected it is evidence against customer deception. If the watermark identifies the specific originating tool as one known to have strong guardrails against misuse, synthesis providers may choose to relax their own computational efforts at

determining whether a novel protein encodes a harmful function (as long as the remainder of the sequence can be annotated as known sequence with low risk). In this way, watermarking could reduce the long-term cost of biosecurity screening, a cost we expect to increase over time as we expend additional computational resources to prove unknown sequences do not recapitulate harmful function.

On the specific points R2 raises: In regard to the confusion around signatures vs encryption, I don't believe the paper itself is unclear on its intent in this regard: a watermark discloses to a limited set of actors (those in possession of the key) that a sequence was engineered. Given our current technological moment, that disclosure is important. Historically, any sequence with little similarity to known sequences would have safely been assumed to have come from an organism we'd never sequenced before. Today there are two possible sources of novelty: a previously unobserved biological source or a sequence design model. I agree with the authors that watermarks can be highly complementary to other more overt annotation technologies like cryptographically signed metadata.

On 'windowed detection', I agree with R2 that detection must be local. A watermark scored over a whole sequence gets diluted whenever the watermarked stretch is only part of the construct, and the authors say as much in Section 5.1. I disagree, however, with the nature of the GFP-2A-XXX-2A-GFP counterexample. Implementing watermark detection in a synthesis screening pipeline would run the current similarity search process first. GFP and the 2A peptide are conserved, well-catalogued elements. Such a pipeline would annotate them and set them aside. The novel low-similarity remainder, XXX, would proceed to watermark detection without flanks to dilute it.

That said, local detection still matters when the watermarked piece is a minority of a novel region with no database hit, when the remainder is too short to clear a threshold, or when an attacker picks flanks designed to avoid annotation. I'd suggest the authors frame their windowed-detection discussion around this kind of annotation-guided localization, which is easier to deploy and doesn't require submitter honesty.

One final point favors the specific approach taken here. By adapting strategies originally developed for generic text, the authors open a path for sequence/structure watermarking to benefit over the long term from future advances in the far larger universe of text and image provenance. Many more researcher-hours will be spent solving those broader sociotechnical problems than will ever be spent on protein-specific watermarking, but biotech may still inherit that progress provided those newer techniques can be adapted to proteins as this paper demonstrates.

(Remarks on code availability)

Open Access This Peer Review File is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

In cases where reviewers are anonymous, credit should be given to 'Anonymous Referee' and the source.

The images or other third party material in this Peer Review File are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain

permission directly from the copyright holder.

To view a copy of this license, visit <https://creativecommons.org/licenses/by/4.0/>

Response to Reviews for Manuscript

“Function-preserving biological watermarking of AI-generated protein sequences and structures”

We appreciate the thorough reviews by all reviewers and the additional points raised by the editor. R2 and R3 agree that this is a timely study because establishing provenance for biological structures and sequences is a pressing problem. Moreover, R1 and R3 positively highlighted our experimental setup and approach as “sound” and “well-designed”.

In the following we want to address the reviewers’ comments in detail. We start by clarifying common concerns shared among reviewers before addressing individual concerns. We also refer to the updated manuscript and supplementary material where we highlight all edits.

Response to common questions

Release of code and in vitro data

We will provide Google Drive access to reviewers which includes the following:

- In vitro data: all protein sequences, the K_D measurements, as well as the K_D curve fits per sequence and the g-values computed on each sequence. This will be made publicly available post decision.
- SynthIDBio-sequence code: a ZIP file with code for SynthIDBio-sequence, running on CPU. This will be made publicly available post decision.
- SynthIDBio-structure model: a ZIP file containing model parameters of SynthIDBio-structure trained with $s = 0.001$ that are fully compatible with the AF3 open-source repository [1]. Note that this excludes the detector. This will be made available as per the current AF3 weight request process.

In all three cases we include README files outlining file formats and/or code/model usage and installation.

We have given the editor access to a Google Drive directory with all of the above and kindly ask the editor to share with all reviewers.

[1] <https://github.com/google-deepmind/alphafold3>

Relevance of watermarking for biosecurity

We thank the reviewers for their measured comments regarding the biosecurity utility of SynthIDBio. As noted across the received feedback, the success of the resequencing attack and the availability of browser-based instances of ProteinMPNN severely impact the biosecurity utility of SynthIDBio-sequence if used in a “punitive” implementation where watermark detection leads to action taken against the user if, e.g., the watermark cannot be detected or validated. While our initial submission included acknowledgement of the challenges posed by the resequencing attack, we have completely rewritten and expanded Section 5.1 to more clearly acknowledge these challenges. Specifically, we include a more concrete proposal around “trusted tools” which we believe to be significantly more grounded in real-world use cases.

We imagine SynthIDBio-sequence as a part of a multi-layered suite of interventions in support of nucleic acid synthesis screening. The goal of such interventions is to defend the

digital-to-physical interface through limiting access to synthesis if nucleic acids are deemed harmful. If submitted orders have low sequence homology to sequences in reference databases, synthesis providers face the difficult decision of whether to fill orders or not. This may involve expert judgement and increased operational costs for, e.g., background checks and more complex screening. Establishing provenance of AI-generated sequences may support these assessments.

A punitive implementation would involve taking action based on the detection of the SynthIDBio-sequence watermark that involves additional work for the user or additional scrutiny when detecting the watermark. For example, watermark detection could trigger requesting additional design details. This disincentivizes usage by good actors and, given widely available open-source alternative tools, bad actors would just avoid using watermarked tools or apply the resequencing attack to remove the watermark.

Now, we explicitly propose using the watermark as a verification signal that “trusted tools” were used in design. With growing calls to incorporate guardrails into design tools, the watermark could verify that a trusted tool with appropriate guardrails was used. For synthesis providers, this signal may lead to lower operational costs by avoiding more computationally expensive screening strategies, staff time spent on in-depth investigations for ambiguous orders, and/or extensive background checks on customers for orders with the watermark. Passing these cost savings on to customers would incentivize good actors to use trusted tools. This increased usage would drive model developers to integrate SynthIDBio into their models. Note that in this context, robustness of the watermarking scheme may actually be counter productive. If watermarking is to serve as verification of using trusted tools, we may want the watermark to be removable even by smaller modifications.

In this context, we also added a separate Section 5.3 on technical challenges in encouraging this adoption and discussion around choosing appropriate operating points in terms of FPR and TPR.

Encouraging watermark adoption, especially in the context of open weight models

Both R1 and R3 had questions on how we envision adoption of watermarking to work considering that there are already numerous open source sequence and structure models and the number will likely keep growing. This is closely connected to a question by R3 on the generalizability of SynthIDBio and what changes in existing tools are needed to integrate SynthIDBio. Moreover, R1 raised the point that some coordination is needed: “Wouldn’t a single model that doesn’t use watermarking ruin the entire system? How would this be enforced?”

We want to emphasize that our paper will not be able to definitively answer these questions. We primarily present a proof of concept that function-preserving watermarking with high detectability is possible. Nevertheless, we want to comment on these concerns in the following and also added a discussion to Section 5.3.

Given a pool of multiple open and closed folding and sequence models, there are several considerations in how to encourage users to add watermarks to their generations. First, we believe that demonstrating watermarking capabilities on state-of-the-art models and making the corresponding code and/or models available will help ensure that users will always have an incentive to use the best models available. Beyond this, we can also create additional incentives to use watermarking. For example, following our proposal on biosecurity above where we may incentivize “good actors” to use watermarking to ensure quicker processing or similar advantages within the AI for biology ecosystem. In this context, the availability of

non-watermarked models is not a problem and we expect that incentivizing users to use “trusted”, watermarked models will in turn incentivize model providers to provide watermarking tools.

In terms of the generality of SynthIDBio, it is worth emphasizing that SynthIDBio-sequence and -structure have different trade-offs. This is particularly important in the context of open weight models but also affects their generality. SynthIDBio-sequence is based on a sampling algorithm. This means that no changes to the underlying model, ProteinMPNN, are required in order to apply SynthIDBio-sequence. Thus, we expect that SynthIDBio-sequence can easily be applied to other or future sequence models that predict per-amino-acid token distributions irrespective of their architecture or how they are trained. However, this also means that the watermarking mechanism can be removed by the user in an open-source/open-weight setting where the user has control over the sampling algorithm used. This is in contrast to SynthIDBio-structure which embeds the information of how to watermark in the model weights via fine-tuning. While this requires fine-tuning, which is expensive and depends on the model architecture and loss formulations, it means that SynthIDBio-structure ensures users cannot remove the watermarking mechanism, even when the model weights are shared openly. In the case of SynthIDBio-structure, our approach is largely adapted to AlphaFold3’s diffusion module. Because diffusion models are common for these types of prediction tasks, we expect this approach to generalize well. We believe that other choices for these trade-offs are also possible; we expect that an approach based on fine-tuning ProteinMPNN to learn how to embed a watermark could be possible. Similarly, there exist sampling-based watermarking approaches for image diffusion models [1] that could be adapted to AlphaFold3.

Beyond these trade-offs, it is worth noting that sharing the ability to detect SynthIDBio watermarks requires sharing secrets – the secret key for SynthIDBio-sequence or the detector for SynthIDBio-structure. As both do, to some extent, enable forging watermarks, this limits sharing with trusted partners. Future work on public-key watermarking would be required to share detectors more widely [2,3,4].

[1] Yang, Z., Zeng, K., Chen, K., Fang, H., Zhang, W.M., & Yu, N.H. (2024). Gaussian Shading: Provable Performance-Lossless Image Watermarking for Diffusion Models. 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 12162-12171.

[2] Collomosse, J.P., Parsons, A., & Potel, M. (2024). To Authenticity, and Beyond! Building Safe and Fair Generative AI Upon the Three Pillars of Provenance. IEEE Computer Graphics and Applications, 44, 82-90.

[3] Fairoze, J., Garg, S., Jha, S., Mahlouljifar, S., Mahmood, M., & Wang, M. (2023). Publicly Detectable Watermarking for Language Models. IACR Commun. Cryptol., 1, 31.

[4] Fairoze, J., Ortiz-Jimenez, G., Vecerik, M., Jha, S., & Goyal, S. (2025). On the Difficulty of Constructing a Robust and Publicly-Detectable Watermark. ArXiv, abs/2502.04901.

Distortionary SynthIDBio-sequence does not affect protein function

R2 raised several concerns about the impact of distortionary watermarking in SynthIDBio-sequence, suggesting it leads to degraded tool functionality. In this context, we want to clarify what distortionary watermarking entails. In LLM watermarking, which SynthID-text was designed for, the more entropy there is in the prediction, the easier it is to watermark. If the entropy is low, distortionary watermarking offers a way to improve detectability. Here, non-distortionary watermarking refers to the property that, in expectation over watermarking keys, the LLM’s distribution over output sequences remains the same. As such, the concept of distortion is primarily a statistical concept. As soon as you fix a key (to deploy the watermarking scheme), however, there will naturally be a shift in this distribution (in order to detect the

watermark). This means that there may exist watermarking schemes that are extremely distortionary while minimally affecting the semantics of the generated text. An easily detectable watermark for a conversational AI would be to never use the word “apartment” and always use the word “flat” instead. This is clearly distortionary in that certain words are not selected. However, it does not affect the understood meaning of the word (semantics). Similarly, the results of our *in vitro* assessment of distortionary and non-distortionary watermarking approaches demonstrate little support for distortionary watermarking negatively affecting protein function (the “semantics” for amino acid sequences).

SynthIDBio-structure robustness to relaxation

Both R1 and R3 raised the question of whether SynthIDBio-structure watermarks are robust to relaxation. We conducted extra experiments and found that relaxation indeed destroys the watermark. Using amber relaxation [1] (3 outer iterations, no max for inner iterations, stiffness 10, tolerance 2.39), the predicted logits largely become negative, overlapping with a typical non-watermarked logit distribution. This means there is no meaningful detectability of the watermark after relaxation with Amber. We clarified this in Section 4.5.

However, similar to work on watermarking images or video [2, 3], we expect that this can be improved by explicitly taking relaxation into account during training. This means ideally implementing a differentiable variant or proxy to relaxation that is added right before the detector during training. We also expect a straight-through estimator could work, i.e., running relaxation before calling the detector, but ignoring relaxation in the backward pass when computing gradients.

[1] Hornak, V. et al. Comparison of multiple Amber force fields and development of improved protein backbone parameters. *Proteins* 65, 712–725 (2006).

[2] Goyal, S., Bunel, R., Stimberg, F., Stutz, D., Ortiz-Jimenez, G., Kouridi, C., Vecerík, M., Hayes, J., Rebuffi, S., Bernard, P., Gamble, C., Horv'ath, M.Z., Kaczmarczyk, F., Kaskasoli, A., Petrov, A., Shumailov, I., Thotakuri, M., Wiles, O., Yung, J., Ahmed, Z., Martin, V., Rosen, S., Savcak, C., Senoner, A., Vyas, N., & Kohli, P. (2025). SynthID-Image: Image watermarking at internet scale. ArXiv, abs/2510.09263.

[3] Fernandez, P., ElSahar, H., Yalniz, I.Z., & Mourachko, A. (2024). Video Seal: Open and Efficient Video Watermarking. ArXiv, abs/2412.09492.

Alteration to data for Figure 2

Since our original submission, we updated our analyses depicted in Figure 2A and B, as our CRO conducted further quality assurance reviews on the binding data. For SC2RBD, they identified higher-than-expected non-specific interaction during their initial SPR run. Incorporating this insight led to 40 SC2RBD binders affinity measurements changing and the re-classification of 12 SC2RBD binders as non-binders. VEGF-A and PD-L1 analyses were not impacted. Importantly, **our main claim of binding affinity being statistically indistinguishable between watermarked and non-watermarked remains unchanged**. This update included the below changes and we updated Section 4.1 accordingly.

1. Refinement of binding constants (K_D) for confirmed binders:

For 40 binders, minor adjustments to kinetic constants (k_{on} , k_{off} , and K_D) occurred as the dataset transitioned from preliminary results to the final release. These shifts are attributed to two specific quality control measures performed by the CRO:

- Replicate consolidation: In cases of insufficient loading or instrument noise, replaced suboptimal replicates with fresh experimental runs.
- Fit re-parameterization: For samples where the residuals between the theoretical fit and measured data exceeded standard thresholds, the fitting algorithm was re-parameterized. This involved tuning blank subtraction, T_0 alignment, and the removal of specific outlier injection cycles.

This re-parameterization significantly improved the accuracy of the kinetic modeling, particularly for binders with fast on- and off-rates. An example of this is shown in the figure below (update from left to right). In the preliminary fit, the association phase showed high residuals; the refined fit adjusted the k_{off} approximately 3.5 faster (from 3.2×10^{-4} to 1.13×10^{-5}), resulting in a fit that more accurately models the experimental data.

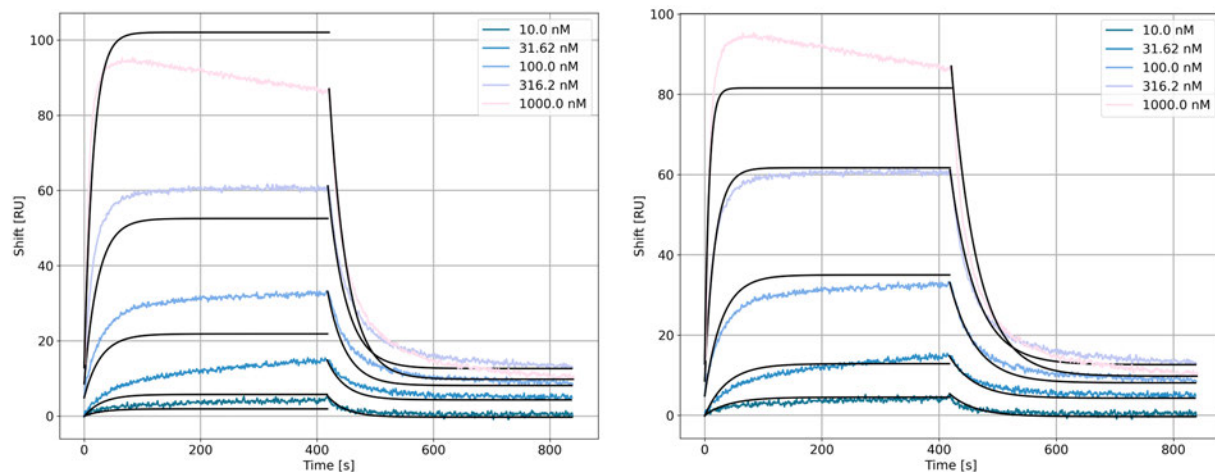


Figure: An example for updated SPR fits from left to right. The graph plots the response units (RU) against time (s) for five different concentrations of the soluble protein partner (10.0–1000.0 nM). In the original fit (left), the association phase showed high residuals; the updated fit (right) adjusted the k_{off} approximately 3.5 faster (from 3.2×10^{-4} to 1.13×10^{-5}). Left: MAE = 4.6 (rel. MAE = 0.05), Right: MAE = 2.38 (rel. MAE = 0.02). Overall, we see a clear improvement from the re-fit on the right, with the MAE dropping by 2.22 (about 48% lower error) and the relative MAE reduced by ~60%.

These binders being refined and re-classified had a minor impact on population affinity measurements, with the mean SC2RBD binding affinity (K_D) of non-watermarked binders changing from 2.36×10^{-7} to 2.29×10^{-7} , distortionary 0.1 from 2.08×10^{-7} to 2.33×10^{-7} , and non-distortionary 0.5 from 2.47×10^{-7} to 2.87×10^{-7} . The mean affinity and significance testing of K_D distributions remained statistically indistinguishable between all watermarked and non-watermarked designs. **The paper's main claim of statistically indistinguishable affinity distributions remains unchanged.** See Figures 2A, S14, S15 and S18.

2. Reclassification of 12 binders to non-binders

12 binders were reclassified as non-binders due to additional experimentation confirming that the binding behavior observed was a consequence of non-specific interactions rather than genuine binding. The 12 binders that were reclassified included 5 distortionary 0.1 and 7 non-distortionary 0.5. The raw value changes for Figure 2B are that the hit rates for binders with a $K_D < 1 \times 10^{-6}$ changed from 65% to 63% for distortionary 0.1, 61% to 58% for non-distortionary 0.5, and was unchanged for non-watermarked (68%). For distortionary 0.1 watermarking, there

were no changes in significance tests across affinity threshold hit rates. For non-distortionary 0.5 there was a single statistically distinguishable ($p < 0.05$) hit rate difference with non-watermarked 0.1 (at the weakest affinity threshold of $1e-6$), the stronger affinity thresholds ($K_D < 1e-7$) remained statistically indistinguishable between non-distortionary 0.5 and non-watermarked 0.1. An updated version of Figure 2B incorporates the new data.

Response to reviewer 1

Availability of code

The authors chose not to provide any code, so any comments below are solely based on the data and description provided in the text, figures, and supplementary information. The underlying theory and fundamental approach are explained well and seem sound to me, but a detailed evaluation of the computational work would require more information. The authors state that code will be made available at time of publication, but for a manuscript that focuses on computational development the code is generally required in the review process to evaluate what has been done.

Please see our response “Release of code and in vitro data” where we now share code and in vitro study data.

Our proposal of using SynthIDBio for biosecurity

The authors motivate this work through use cases in biosecurity and research integrity. However, I fail to understand how the proposed watermarking approach could help in either case.

For biosecurity (section 5.1), the authors refer to the digital-to-physical frontier and DNA synthesis companies’ conditions. How do the authors envision this would work? How would identifying an MPNN-generated sequence help distinguish what should or should not be synthesized? Would DNA synthesis companies refuse to synthesize designed sequences?

Please see our response “Relevance of watermarking for biosecurity” above and the comprehensive discussion in Section 5.1. Generally, we believe that a punitive implementation in which detection of the watermark leads to additional work for either user and/or synthesis provider is not helpful and thus our proposal is about using the watermark to verify the usage of “trusted tools” with guardrails. Then, SynthIDBio could be an additional layer in the screening process, also see our response “SynthIDBio as additional layer for biosecurity” for R3.

Watermark adoption with a growing number of available models

There are already many different models for sequence design and structure prediction, and the number keeps growing every week. Would the authors expect that every newly developed model adopts the watermarking strategy? Wouldn’t a single model that doesn’t use watermarking ruin the entire system? How would this be enforced? I understand that this paper is about technology and not about policy, but I simply do not understand how this would be helpful.

Please see our response “Encouraging watermark adoption, especially in the context of open weight models”. We believe that building good incentives for users to use “trusted”, watermarked models and these models being state-of-the-art can go a long way in driving

adoption. Specific to nucleic acid synthesis screening, please see our response “Relevance of watermarking for biosecurity” above for a proposal. Concretely:

- *Would the authors expect that every newly developed model adopts the watermarking strategy?* We expect that model developers will be incentivized to support watermarking if there is sufficient interest by users for watermarked, trusted tools.
- *Wouldn't a single model that doesn't use watermarking ruin the entire system?* No, there may still be tools that do not integrate watermarking and these may continue to be used for various reasons. However, with strong enough incentives, we can imagine that a significant volume of users may migrate to tools that support watermarking.
- *How would this be enforced?* No explicit enforcement is needed if incentives are properly constructed.

Watermarking to establish specific BDT used

The authors suggest that “watermarking could be used to reliably establish which specific tools were used for design”. However, the structure-based watermark would only give information about the very last tool that was used in the pipeline, since all structural information from earlier steps would be lost in the last sequence-to-structure prediction step; meaning that in atypical RFDiffusion - proteinMPNN - AlphaFold pipeline, the initial backbone generation step would be lost.

This is a good question and we believe there are two related but distinct questions here. First, how can we reliably add watermarks that identify specific tools when multiple tools are combined. And second, in the concrete example of a protein design pipeline, using RFDiffusion - ProteinMPNN - AlphaFold3, why we did not consider watermarking the initial backbone.

On the first question, watermarks on different modalities are usually complementary. This means that the SynthIDBio-watermarked sequence predicted by ProteinMPNN will stay watermarked even when folded with SynthIDBio-structure. However, if we were to use, e.g., different, watermarked sequence models on top of each other, we can currently not explicitly identify these models and would indeed only detect the last model that was used. However, in the context of image and audio watermarking, there is recent work [1,2] that shows that watermarks can co-exist to some extent (albeit reducing detectability). This indicates that using watermarks to identify the usage of specific tools, even if tools are combined, may be possible depending on the modality (i.e., structure rather than sequence).

On the second question, the output of a state-of-the-art protein design pipeline is primarily the sequence. More importantly, the backbone does not represent something that can exist in the real world with some biological function. Thus, the backbone is less relevant in terms of the applications we discuss in the manuscript. As a result, we did not consider watermarking the backbone.

[1] Petrov, A., Agarwal, S., Torr, P.H., Bibi, A., & Collomosse, J.P. (2025). On the Coexistence and Ensembling of Watermarks. ArXiv, abs/2501.17356.

[2] Liu, H., Guo, M., Jiang, Z., Wang, L., & Gong, N.Z. (2024). AudioMarkBench: Benchmarking Robustness of Audio Watermarking. ArXiv, abs/2406.06979.

Reasonable TPRs and FPRs

In section 5.2, the authors suggest that watermarking with SynthIDBio could be used to prevent submission of AI-generated sequences or structures to databases. While this sounds like a great safeguard, I am concerned about the implications of the false positive rates (FPR) here. Throughout the manuscript, the authors set a target FPR of 0.1%, which sounds low. However,

given the sheer size of the databases involved, this seems orders of magnitude too high. Given a 0.1% FPR, about 244 experimental structures from the PDB would be flagged as AI-generated, and 200 human protein sequences from UniProt would be flagged as designed with proteinMPNN. Assuming multiple different generative models, prediction networks and sequence design tools all used independent watermarking / separate watermarking keys, I would expect that this number would increase near-linearly with the number of watermarking keys used. Would this not be prohibitive for any intended use case? This should be addressed, ideally by including some kind of benchmark on natural sequences and structures.

We agree that choosing an appropriate trade-off of TPR and FPR is essential not only for scientific integrity but also for biosecurity, especially in light of the implementation suggested in the revised Section 5.1.

First, we want to emphasize that SynthIDBio-sequence and -structure have slightly different trade-offs when selecting TPR and FPR thresholds. For SynthIDBio-sequence, the TPR can be fixed (e.g., at 100%) via g-value filtering for a specific target FPR. Note that this is a TPR measured on unmodified sequences (i.e., we assume no attackers at this point). More stringent FPR thresholds will result in lower pass rates, increasing the per-sample cost for protein design (as explored in the new Section A.6). We tested SynthIDBio-sequence on significantly lower FPRs with high TPR even for 0.005% FPR in Figure S4. Due to the mechanism used for watermarking (i.e., tournament sampling), there are only a few knobs to control this trade-off, namely temperature and distortionary or non-distortionary watermarking as we detail in Section 2. For SynthIDBio-structure, the trade-off is more controllable since it is a learned model. Similar to image watermarking [1], there is a well-known trade-off between detectability, robustness and impact on quality which we expect is fairly controllable via adapting training. In the models we trained, we still see very good TPR at significantly lower FPR, for example, the $s = 0.001$ model obtained 98.99% TPR at 0.01% FPR. Note that we added additional results in Figure 3 for lower FPRs as well as Figure S4. In both cases, the trade-off can further be controlled by adding an option to abstain, i.e., not make a decision, as outlined in the SynthID-text paper.

That being said, we expect that choosing an appropriate FPR threshold will ultimately be the responsibility of the database provider in the scientific integrity use case or the synthesis provider in the biosecurity use case. These providers are motivated to choose an operating point with very high TPR at very low FPR. This is to avoid false negatives and false positives: In the scientific integrity use case, false negatives would lead to misinformation, while false positives would primarily lead to extra work performing verification and a risk of spurious accusations of AI assistance. For biosecurity, false positives would avoid more intensive screening (in our proposal in Section 5.1), which itself poses a biosecurity risk. At the same time, however, providers also are motivated to incentivize adoption of watermarking because the more users use the watermark, the higher the benefit (e.g., in reduced screening cost in biosecurity). Very strict requirements in terms of TPR/FPR will usually make the tool harder to use. For example, in SynthIDBio-sequence, guaranteeing 100% TPR via g-value thresholding for extremely low FPR will result in significant reductions of pass rates. This may result in additional overhead for the user (even though this is limited as per our analysis in Section A.5). If this cost becomes too high, the user is disincentivized to use watermarking – this happens if, e.g., the additional compute cost for SynthIDBio-sequence outweighs the gained speed or cost shared by the synthesis provider due to using a “trusted” tool. Ultimately, this means that the provider will have to carefully choose TPR/FPR requirements to control false negatives/false positives while ensuring adoption by users.

Furthermore, to specifically address the concern regarding the scale of global databases, we have expanded our analysis of SynthIDBio-sequence to include thresholds calibrated on 46.4 million sequences from the UniRef50 dataset. We demonstrate that SynthIDBio-sequence supports the extremely rigorous FPR thresholds (down to $10e-06$) required for database-wide deployment without compromising economic viability. See supplementary Section A.2.

Finally, we included part of this discussion in a new section about adoption of watermarking in Section 5.3.

[1] Goyal, S., Bunel, R., Stimberg, F., Stutz, D., Ortiz-Jimenez, G., Kouridi, C., Vecerik, M., Hayes, J., Rebuffi, S., Bernard, P., Gamble, C., Horv'ath, M.Z., Kaczmarczyk, F., Kaskasoli, A., Petrov, A., Shumailov, I., Thotakuri, M., Wiles, O., Yung, J., Ahmed, Z., Martin, V., Rosen, S., Savcak, C., Senoner, A., Vyas, N., & Kohli, P. (2025). SynthID-Image: Image watermarking at internet scale. ArXiv, abs/2510.09263.

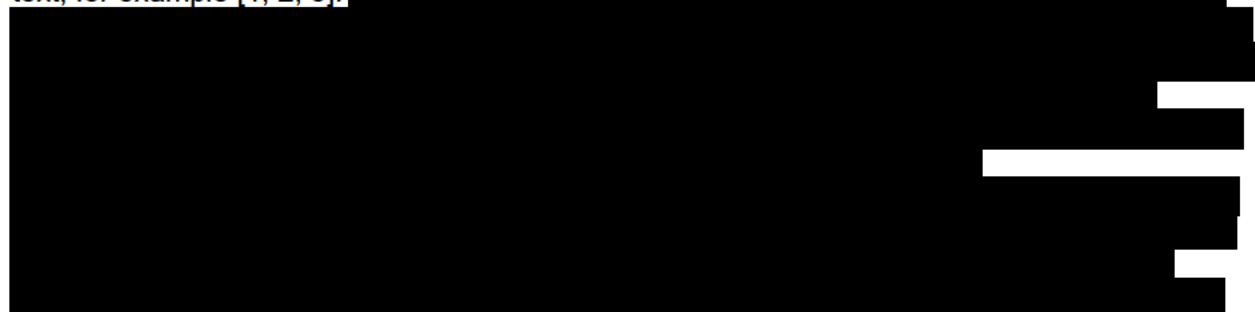
Watermarking vs. “detecting AI-generated sequences”

Overall, the authors focus on distinguishing watermarked proteinMPNN-generated sequences from non-watermarked proteinMPNN-generated sequences. Wouldn't most applications (besides IP protection) rather require distinguishing artificially generated sequences from natural sequences? How distinguishable are natural and designed sequences without any watermarking? Would proteinMPNN-generated sequences have some distinguishable properties already?

In general, we see two problems with post-hoc detection of AI generated sequences/structures vs. actually watermarking them.

First, detection will ultimately not provide the provenance information we need. Specifically, in our proposal of how watermarking can be beneficial for biosecurity, we would want to establish a set of “trusted” tools. Distinguishing between the usage of different tools is key to making this work. As is apparent, distinguishing different tools is significantly harder than “just” detecting AI generated vs. benign.

Second, we expect that detection may become more and more difficult given the current speed of developments in AI for biology. This has been shown repeatedly in the case of AI generated text, for example [1, 2, 3].



Given several concerns by reviewers about the need for high TPR at extremely low FPRs, we do not believe that this level of detectability, both for sequences and structures, is sufficient. Thus, we do not believe that this is a promising direction for the use cases we have in mind, including biosecurity and scientific integrity.

[1] Fraser, K.C., Dawkins, H., & Kiritchenko, S. (2024). Detecting AI-Generated Text: Factors Influencing Detectability with Current Methods. ArXiv, abs/2406.15583.

- [2] Ghosal, S.S., Chakraborty, S., Geiping, J., Huang, F., Manocha, D., & Bedi, A.S. (2023). Towards Possibilities & Impossibilities of AI-generated Text Detection: A Survey. ArXiv, abs/2310.15264.
- [3] Tang, R., Chuang, Y., & Hu, X. (2023). The Science of Detecting LLM-Generated Text. Communications of the ACM, 67, 50 - 59.
- [4] Krishna, Rohith et al. "Generalized Biomolecular Modeling and Design with RoseTTAFold All-Atom." bioRxiv (2023): n. pag.

Reasonable TPR

Regarding true positive rates (TPR), it would be great to have some kind of discussion about what level of TPR would be required for specific use cases (how good does it need to be in order to be useful).

Please see our response "Reasonable TPRs and FPRs" above. We also added extensive discussion to Section 5.3.

Robustness of SynthIDBio-sequence

Regarding the robustness of SynthIDBio-sequence, the authors consider different levels of sophistication and information. Given that proteinMPNN and AlphaFold are freely available via browser-based interfaces, I would suggest assuming that any attacker would have access to these tools. Using resequencing with proteinMPNN, it seems very easy to remove watermarks, and the only drawback would be a reduced success rate. But what if the attacker only resequences parts of the protein? For a binder, one could expect that resequencing only the non-interface residues would lead to a significantly higher success rate (=preserve the function); would this also be enough to remove the watermark? What if the attacker adds residues to either terminus of the protein that don't interfere with the function?

There are numerous approaches for attacking SynthIDBio-sequence that will have variable rates of success. Beyond the partial resequencing and terminus addition attacks mentioned by the reviewer, other attacks we considered included addition of glycines into the center of predicted loops (disrupting detectability of adjacent H-grams), interweaving two watermarked sequences of fixed lengths, and adding a restriction site plus unrelated residues to the 3' end of a synthesis order and digesting the gene prior to protein production. These methods would need to be evaluated for their expected impact on *in vitro* pass rates and the additional compute costs incurred by an attacker. Results with the naive substitution method (Figures S37 and S38) demonstrate that relatively minor modifications to the watermarked sequence can result in dramatic decreases in laboratory confirmation. We believe that evaluating all of these is beyond the scope of the manuscript but would be a necessary component of operationalizing this technology. We have added a reference to this effect in Section 5.5.

Regarding the potential impact of partial resequencing and terminus addition, our substitution-based attacks in A.9 are a reasonable proxy for the impact of any attack on detectability as they showcase the proportional decrease in signal as the watermarking signal is diluted through removal of watermarked residues or disruption of the H-gram used to generate the detector. The partial resequencing technique is likely to be especially potent as it would interpolate between our substitution-based attack results and the full resequencing attack shown in Figure S39B. We also updated Figure S39B to make the results of the resequencing attack more clear.

Lastly, the proposed implementation in Section 5.1 decreases the motivation for attacking SynthIDBio-sequence.

Robustness of SynthIDBio-structure

For SynthIDBio-structure, the authors state that the watermarking is robust to adding 0.1 Angstrom Gaussian noise. How about other ways to modify a protein structure model? What if an attacker uses for example Rosetta or a similar tool to relax the structure in torsional and/or Cartesian space? Ideally, I would have loved to give this a try, but unfortunately the authors chose not to provide any code.

Please see our response “SynthIDBio-structure robustness to relaxation”. Also note that we are making the SynthIDBio-structure model available, though without the detector.

Release of in vitro data and additional details

The in vitro work looks sound to me, and convincingly shows that sequence watermarking does not affect binding properties in the chosen setting, although once again very little detail is provided:

- *Ideally, sequences (and pdb files) for all tested designs should be provided.*
- *Was there any type of sequence validation after cloning/assembly?*
- *Was there any type of purification or quality control of the final proteins?*
- *Negative controls are mentioned in the text and methods section, but no result is ever reported for them.*
- *Raw data or at least plots of fitted data should be provided for SPR measurements.*

Please see our response “Release of code and in vitro data” where we provide all sequences as well as fits.

We’ve greatly expanded the description of the *in vitro* methods. Specific to the reviewer’s questions, post-assembly 10-15% of sequences were randomly selected for verification via batch sanger sequencing, all of which had their sequence identity confirmed. There were no isolated purification steps for the expressed binders. However, the expression system should provide for purified protein. As the binders adhere to the Strep-Tactin XT chip via the C-terminal affinity tag, the washing steps of the SPR should remove any impurities.

In addition, we added results for the negative controls in the main paper, Section 4.1: “All negative controls (n = 9) had no detectable binding.”

Response to reviewer 2

SynthIDBio couples good detectability with function preserving watermarking

What is not currently clear from the manuscript, however, is the cost/benefit tradeoff for doing so. The authors clearly demonstrate that in some circumstances their methods enable detection, and that in some circumstances the distortion introduced by their methods is not significant. Unfortunately, however, the current presentation in the manuscript made it difficult for me to tell whether those circumstances overlap enough to make the methods valuable. Instead, it appears to me that the size of distortion required for reliable detection may necessarily result in a major degradation of tool functionality. I may be incorrect about that, but I think it will require significant improvements in the manuscript to make that clear.

The reviewer is concerned about a lack of overlap between circumstances when SynthIDBio watermarks are detectable and when they are not distortionary. We can answer this concern directly and quantitatively. First, we want to emphasize that “distortionary” watermarking is a

statistical statement and not one about protein function as detailed in our response “Distortionary SynthIDBio-sequence does not affect protein function”. It is more meaningful to consider whether the watermarks are “perturbative” in the sense of reducing hit rates or binding affinities. Our measurements show clearly that these practical outcomes are unaffected by watermarking, i.e. the watermarks are non-perturbative across all three targets, see Figure 2. In this case we can achieve 100% TPR at 0.1% FPR.

Our responses below provide additional evidence:

- We can get good detectability at even lower FPRs.
- While we use “distortionary” watermarking at low temperature, this is primarily a statistical concept without impact on protein function.
- Watermarking does not affect the diversity of generated sequences; even with distortionary watermarking.

Lower FPRs

- If one is looking to see if any protein in a cellular organism is watermarked, there would typically be at least hundreds or thousands of proteins to consider, which means that even the 0.1% false positive rate is fairly high. The higher rates considered by the authors are essentially guaranteeing any engineered protein would be lost against a background of false positives. For checking the integrity of submissions to scientific databases, the false positive rates would need to be even lower.

To demonstrate that SynthIDBio works well at lower FPRs, we re-plotted our in silico results, as shown in Figure 3. As expected and detailed in our response “Temperature and distortionary watermarking” below, TPR at lower FPR drastically reduces, especially for temperatures 0.1 and 0.3 without distortionary watermarking. However, it is important to remember that we can always fix the TPR to 100% via g-value filtering. This means we can pretty much freely choose FPR, while fixing TPR at 100%, by spending additional compute. As the additional compute cost is actually limited, see our response “Cost of SynthIDBio-sequence” below, we can run SynthIDBio-sequence even at very low FPR without incurring significant additional cost.

We also included additional results on our in vitro designs in Figures S2 and S3, considering 46.4M wild-type sequences from UniRef. Among others, these results show that our FPR rates, calibrated following Table S2, are extremely conservative. Considering 46.4M UniRef sequences, the UniRef50 measured FPR is usually half to a full magnitude lower than ours (i.e., 0.001% instead of 0.01%). This is because we chose our original calibration set to be rather conservative, including many short sequences that exhibit higher variation in average g-values. This shows that the actual FPR on submissions to scientific databases will be much smaller.

Note that for SynthIDBio-structure, detectability is incredibly high as illustrated by the near perfect ROC curves in our response “More ROC curves” below.

Note that, in order to evaluate lower FPRs, we re-plotted Figure 3 with 10k candidates for each of the 23 targets (instead of 5k previously). As a result, the reported numbers changed very slightly (e.g., non-distortionary 0.5 watermarking pass rate at 0.1% changed from -41.7% to -41.0%). We refer to more details in the respective answers below.

Temperature and distortionary watermarking

- For 0.1% false positives, the true positive rates are extremely low for normal operating temperatures (0.1 - 0.3) and not great even at 0.5. Only with distorted sampling does the detection become really reliable.

This is correct for the tournament sampling approach we use for watermarking. However, it ignores the fact that we can use g-value filtering to guarantee a TPR of 100%, which we also did in our in vitro validation study. In this case, we trade a lower pass rate for 100% TPR irrespective of the temperature or distortionary/non-distortionary watermarking. Because the additional monetary/compute cost of a lower pass rate is minimal, see our response “Cost of SynthIDBio-sequence”, we believe that the TPR without g-value filtering is less relevant.

We also want to comment on what “distortionary” watermarking actually means. Temperature reflects an integral trade-off in tournament sampling: if there is low entropy (at low temperature) there is little room to include a watermark. This is equivalent to image watermarking, for example: it is difficult to watermark an all-black image (zero entropy), while it is easy to hide the watermark in images with rich texture. In LLM watermarking, this trade-off can be controlled by using distortionary watermarking schemes. Here, non-distortionary refers to the property that, in expectation over watermarking keys, the LLM’s distribution over output sequences remains the same. Meaning it is primarily a statistical property that is unrelated to protein function. Also see our response “Distortionary SynthIDBio-sequence does not affect protein function” at the beginning of this response for a more thorough discussion of distortionary watermarking.

Moreover, we show that increased temperature is unproblematic in terms of protein function. We commented on this in the main paper Section 2..

Impact on sequence diversity

- Fig. S9, on the other hand, makes it quite clear that the distorted sampling approach results in a dramatic reduction in sequence diversity, though this is not quantified.

The diversity reductions from SynthIDBio-sequence are not quantified - they are only measured indirectly through the distribution of K_D values and AlphaFold confidence values.

I thus think that it would be extremely important to quantify the direct effect on sequence diversity, e.g., through entropy measures of the distribution over residues. Given what has been presented, I suspect that there will be quite a significant decrease in entropy for the distorted sampling, which calls the viability of the watermarking approach into doubt.

We thank the reviewer for this comment and appreciate that the original Figure S9 (now Figure S21) may have given the inadvertent impression that distortion decreases sequence diversity. In contrast, both non-distortionary 0.5 and distortionary 0.1 watermarking settings do not decrease sequence diversity compared to non-watermarking as detailed in the new supplementary section A.4. We demonstrate this effect using Shannon entropy values before any filtering has been applied (Figure S5), Shannon entropy values (Figure S6) after filtering has been applied, sequence similarity with parent consensus sequence after filtering (Figure S7), and sequence similarity compared to parent consensus sequence after filtering (Figure S8). Across all comparisons, both non-distortionary 0.5 and distortionary 0.1 are roughly equivalent in their increase in diversity over non-watermarking. While the addition of the g-value filter itself likely decreases sequence diversity, our hypothesis is that the introduction of watermarking at the generation step enables a much higher rate of sampling of low probability tokens during resequencing. This, in turn, allows for the protein design process to explore more distant sequence spaces initially, and ultimately enables a more diverse population of sequences.

Statistical power analysis

In Section 9.5, the standard deviation of log K_D measurements is said to be 2.5, which means that there is not much power to discriminate on a range of -10.5 to -5.5. The distribution of K_D measurements shown in Fig. S2 and the improvement rates shown in Fig. S3 thus seem to be primarily driven by the uncertainty in the in vitro assay and reflect little about the properties of the actual distribution.

The power analysis mentioned in Section 9.5 was purely to determine sample size before performing the experiment. Retrospectively, given the actual data, the statistical power is much higher and we are confident in our conclusions. In particular, assuming a standard deviation in log K_D of 2.5, a two sample t-test would have allowed us to get ~80% power using 90 designs per target and mode (watermarked, non-watermarked). These were assumptions made before the actual *in vitro* study to guide the number of designs used for validation given practical considerations such as provider availability and cost. More importantly, the assumed standard deviation of 2.5 applies to the mean binding affinity. Thus, this was a conservative assumption: across targets depicted in Figure 2A, we find standard deviations of log K_D to be 0.326, 0.301, 0.289, and 0.286 for 0.1 distortionary watermarking, non-watermarked, 0.5 non-distortionary watermarking, and parent sequences, respectively. This means the actual power of our test is significantly higher (re-running the same analysis gives a power close to 100%). This makes us very confident that SynthIDBio-sequence is actually function-preserving.

As we appreciate that Section 9.5 caused some confusion, we removed details on this power analysis as it is not relevant to the results presented in the paper.

Near-perfect detectability

Certainly, however, the evidence presented does not support the authors' claims of "near-perfect detectability" for anything but distorted distributions.

As detailed in our response "Temperature and distortionary watermarking", we show that distortionary watermarking does not affect protein function. Moreover, in our response "Impact on sequence diversity", we showed that distortionary watermarking does not negatively affect sequence diversity (diversity actually improves). We show near-perfect detectability for temperature 0.5 without distortionary watermarking, with 100% TPR in our in vitro study, and we are able to guarantee 100% TPR with g-value filtering for very low FPR. Overall, we believe that we provide sufficient evidence for this claim.

Clarifying the distortion used in SynthIDBio-structure

For SynthIDBio-structure, the big question, which is not adequately addressed by the current presentation, is the degree of impact that the distortion has on the validity of model results. Using perturbations for SynthIDBio-structure range of up to 0.1 Angstroms is quite high, given that the distance between many atoms is on the order of only 1-3 Angstroms. How big an impact is this having on model quality?

Training SynthIDBio-structure with $s=0.1$ means that we inject noise of 0.1 Angstrom during training (to improve robustness), not that there has to be a change of 0.1 Angstrom in all the atom coordinates (or all atom-atom distances). For example, in Figure S42, we show that many feature distributions are actually not impacted. Those atom-atom distances that are impacted often show a change smaller than 0.1 Angstrom. Moreover, in our response "Impact of feature distributions for higher s " below, we further show that there is a negligible shift in atom-atom distances for the majority atom-atom pairs, especially for $s < 0.1$. Nevertheless, we do not recommend using $s = 0.1$ in practice and mostly included it for ablation purposes. This is why the majority of our results are conducted on $s = 0.001$. For $s = 0.001$, we are convinced that

Figures 4 and S42-45 provide sufficient evidence that SynthIDBio-structure (a) does not impact key metrics such as IDDT, TM score, or RMSD, and (b) shows negligible impact on feature distributions.

Impact of feature distributions for higher s

Fig. 4b and Fig. S28e show that even with the minimum studied perturbation of $S=0.001$, there are quite significant changes in the distribution of feature values with respect to non-distorted AF3. Distributions for higher levels are not presented, and need to be. The summary numbers in Fig. S28d, however, already show that some properties change by an order of magnitude for $S=0.1$, which seems to me to throw the validity of the watermarked model into grave doubt.

We want to clarify what original Figure S28E, now distributed over Figures S42 to S45, is meant to illustrate: by the nature of having a highly-detectable watermark, there has to be a shift in feature distributions between SynthIDBio-structure and AF3. However, what actually matters is the shift relative to the ground truth distribution. We believe that, especially for $s = 0.001$, the shift relative to the ground truth is largely indistinguishable from AF3, meaning that SynthIDBio-structure does not introduce additional error over the error AF3 already has.

To support this argument, we added additional figures for all settings of s to the supplementary information, Figures S42 to S45. We believe that, in most cases, the change in feature distributions is negligible, in particular for the $s=0.001$ model. The $s=0.1$ model, in contrast, does indeed result in larger shifts that go beyond the inherent error in AF3. We included $s > 0.001$ in our experiments to illustrate this trade-off, which is a common trade-off in watermarking.

We also want to clarify the plot in original Figure S28D (now Figure S41E): Here, we show relative changes of SynthIDBio-structure compared to the *error* between AF3 and the ground truth. If AF3 makes good predictions (e.g., atom-atom distances are close to the ground truth), the error is small which means that the percentage increase shown in Figure S41D may be high even though the absolute changes are very small. As a result, these percentages do not tell us whether SynthIDBio-structure does increase error on top of the error AF3 already induces (relative to the ground truth) which is ultimately what is relevant.

More ROC curves

I think it would be extremely useful to see the ROC curves not just for the case of structures perturbed with 0.1 Angstrom noise, but also for the cases that are producing graphs 4a, S28a and S28b, as well as for 0.01 Angstrom and 0.001 Angstrom noise. As it is, between the lack of an ROC curve and the figure issues detailed below, the significance of the detectability results is hard for me to evaluate.

Below, you can find ROC curves evaluated with less or no noise at test time. We did not include these results in the original manuscript as well as the revised version since most of the ROC curves show near-perfect detectability, even when plotting TPR against log FPR. However, we are happy to include them in the final supplementary information if the reviewer prefers. These plots show that ROC curves are not sensitive enough to assess SynthIDBio-structure's detectability as detectability is already very high, especially when evaluation without noise or with noise with standard deviation < 0.1 . We did, however, add ROC curves for our evaluation by structure length, evaluated at noise with standard deviation 0.1, in Figure S41F. Again, evaluated without noise, these ROC curves would be similar to those below.

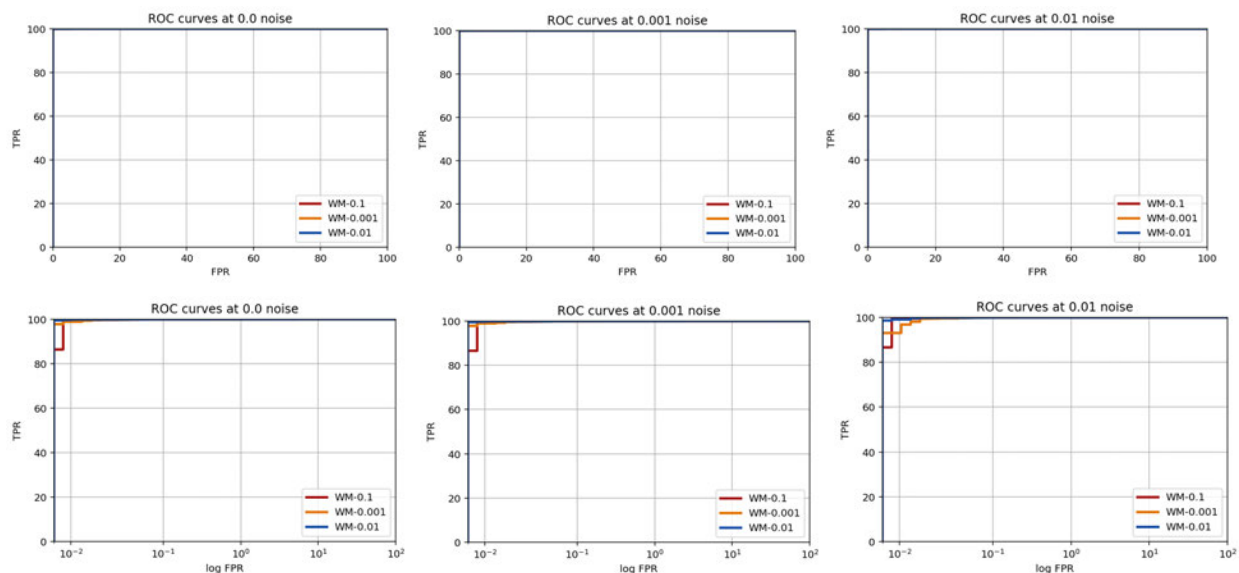


Figure: ROC curves for all models, $s = 0.001, 0.01, 0.1$, evaluated at different noise levels ranging from no noise (left) to noise of standard deviation 0.01 (right). Top: regular ROC curve plotting FPR (x-axis) against TPR (y-axis). Bottom: plotting log FPR (with pyplot's symlog set to $\text{linthresh}=0.05$) to illustrate the high detectability. Going below 0.01% FPR is not reasonable due to the limited sample size we evaluated; it can be seen that the ROC curve becomes extremely coarse below 0.01% FPR (1 FP per 10000 samples).

Cost of SynthIDBio-sequence

- How much additional compute is required for design generation for SynthIDBio-sequence? Multiplying the number of designs by 2^{25} is not a small cost.

The additional cost for watermarked sequence design is not 2^{25} . We want to stress that the number of layers (25) used in SynthIDBio-sequence have no impact on additional cost. In fact, following the SynthID-text paper [1], the additional overhead on sampling is negligible. The original paper cites a latency increase of only 0.57% in total with similar settings which we expect to be lower for SynthIDBio-sequence due to the significantly smaller vocabulary size. This is because tournament sampling does not need to actually sample 2^{25} candidate tokens to perform the tournament; it can be implemented as a transformation on top of the predicted logits. We refer to the open-source SynthID-text implementation [2] for details and clarified this in Section 2.

Beyond the additional compute needed for sampling, SynthIDBio-sequence also induces additional cost due to a higher number of candidates being required. In Figure 3B, we show the pass rate reduction incurred by watermarking without any g-value threshold and with g-value thresholds for different FPRs. For example, considering 0.5 distortionary watermarking, with a 0.1% FPR threshold, the pass rate reduces by 41% across a range of targets (see Table S3). This means the following: if X is the pass rate for unwatermarked designs and X' is the pass rate of watermarked designs, then $1 - X'/X$ is the reduction in pass rate we expect. This means that we need to generate $100\% / (100\% - 41\%) = 1.695$ times the candidates to obtain the same number of passed designs. The required compute scales accordingly. We added a plot illustrating this in Figure S35B.

Based on the reduced pass rate, we added a concrete cost analysis to Section A.5. We demonstrate that SynthIDBio-sequence watermarking imposes a manageable cost on protein design pipelines. This is largely due to the cost of structural filtering via AF3 largely overshadowing the cost for additional resequencing needed due to reduced pass rate. More importantly, we found that the FPR stringency does have negligible effect on the additional cost. At our standard operating threshold (0.1% FPR), the estimated cost per successful binder increases from a baseline of ~\$0.12 to ~\$0.15 for non-distortionary schemes and ~\$0.18 for distortionary schemes. This marginal increase is driven entirely by the initial reduction in AF3 validation rates rather than the computational burden of additional ProteinMPNN inferences. This is largely because running ProteinMPNN is significantly cheaper than structure validation via AF3; the latter is, however, only required for designs that pass the g-value threshold.

[1] Dathathri, S., See, A., Ghaisas, S., Huang, P., McAdam, R., Welbl, J., Bachani, V., Kaskasoli, A., Stanforth, R., Matejovicova, T., Hayes, J., Vyas, N., Merey, M.A., Brown-Cohen, J., Bunel, R., Balle, B., Cemgil, T., Ahmed, Z., Stacpoole, K., Shumailov, I., Baetu, C., Goyal, S., Hassabis, D., & Kohli, P. (2024). Scalable watermarking for identifying large language model outputs. *Nature*, 634, 818 - 823.

[2] <https://github.com/google-deepmind/synthid-text>

- How large are the models that need to be stored for each key for SynthIDBio-sequence?

Considering Figure S1, both the sampling algorithm and the detection algorithm only require storing the secret watermarking key (an integer value). This watermarking key is then used to derive individual per-layer keys, in our case for 25 layers, k_l in Section 2 of the manuscript. This means the additional storage required on top of ProteinMPNN, which we do *not* change, is negligible (i.e., an additional integer).

Cost of SynthIDBio-structure

- How much compute is required for fine-tuning the model for SynthIDBio-structure? What is the tradeoff between amount of fine tuning, performance, and cost?

For the 3 models included in the paper, $s = 0.001, 0.01$, and 0.1 , each fine-tuning run used 256 A100 GPUs for roughly 1 day. In fact, our fine-tuning setup is very close to the “Fine tuning 3” stage reported in the AF3 supplementary information [1], fine-tuning for roughly 1 day, using the same other parameters, except that we reduced the sampling weight for disorder PDB distillation to 0.01 (from 0.02). Compared to the overall cost of training AF3 (~20 days), this constitutes an extremely small extra cost ($1/20$). We did not perform exhaustive ablation into the trade-off of performance and training time but did notice that training considerably shorter than that leads to a noticeable decrease in performance judging from the continuous evaluation on ground truth structures shown below. The error only really reduces in the later part of training. Because identifying ground truth structures as non-watermarked is considerably easier than identifying non-watermarked structures predicted by AlphaFold3, we expect that reducing training time will likely result in a degradation of performance. We emphasize that SynthIDBio-structure does *not* add additional compute requirements or time at inference; meaning the inference time reported in the AlphaFold3 supplementary information is accurate. We clarified this in Section 10.2.

[1] Abramson, J., Adler, J., Dunger, J. et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature* 630, 493–500 (2024).
<https://doi.org/10.1038/s41586-024-07487-w>

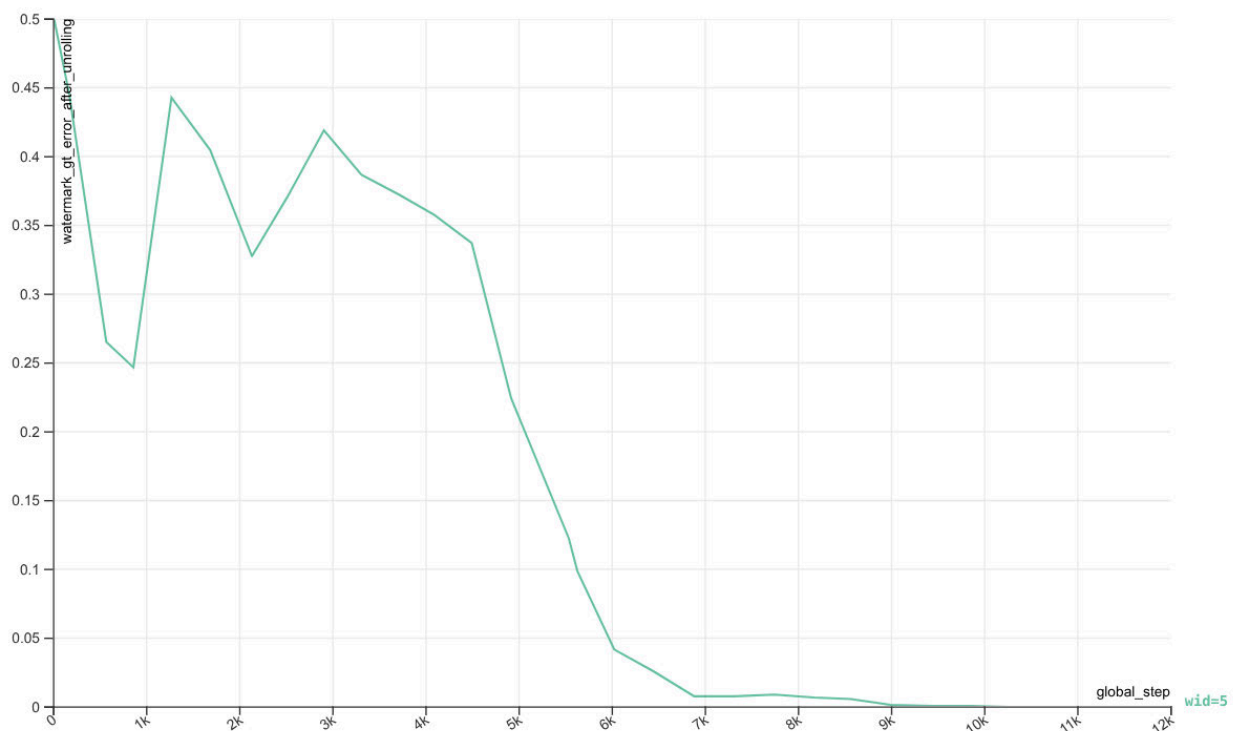


Figure: Error in detecting watermarked structures vs. ground truth structures (y-axis) by the detector throughout training steps (x-axis). Because distinguishing SynthIDBio-structure predicted structures from AF3 predicted structures is significantly harder, and the error curve only reduces towards the latter part of the training, training for significantly shorter is likely not advisable.

Clarifying the H-gram context

- A k -mer length of $H=4$ seems quite low for ProteinMPNN - aren't k -mer lengths typically significantly longer than that? Using such a low length seems likely to have a significant impact on costs, detection, and design viability.

The preceding context used by ProteinMPNN for resequencing is not impacted by the introduction of H-grams. These values are used to perturb the SynthID-text hash functions, along with the input token and fixed seed. With an H-gram length of 4 applied to an amino acid sequence, there are 160,000 (20^4) potential perturbations. In terms of the number of unique values, this is between nucleotide k -mer lengths of 8 and 9.

Longer proteins

- The binders used for testing are all short proteins. What happens if the proteins are hundreds of amino acids long, like most natural proteins?

While we have focused on watermarking protein binders with SynthIDBio-sequence to enable wet lab validation of our designs, previous work on SynthID-text [1] has established that longer watermarked texts are easier to detect. This is intuitive as there are more tokens to watermark and the scores for unwatermarked sequences more tightly associate with the null expectation of 0.5. The latter can be seen in Figure S34B. We have added a reference to SynthID-text in Section 2.1 to clarify this.

[1] Dathathri, S., See, A., Ghaisas, S., Huang, P., McAdam, R., Welbl, J., Bachani, V., Kaskasoli, A., Stanforth, R., Matejovicova, T., Hayes, J., Vyas, N., Merey, M.A., Brown-Cohen, J., Bunel, R., Balle, B., Cemgil, T., Ahmed, Z., Stacpoole, K., Shumailov, I., Baetu, C., Goyal, S., Hassabis, D., & Kohli, P. (2024). Scalable watermarking for identifying large language model outputs. *Nature*, 634, 818 - 823.

Effect of resequencing

- Fig. 3d shows the impact of resequencing on hit rate, but we do not see the impact of resequencing on the ability to detect the watermark. My guess is that the watermark is effectively destroyed after resequencing.

It is correct that resequencing effectively destroys the watermark. We have made this more explicit within the main text: "Resequencing reduces the watermark detectability to its FPR detection rate, effectively removing the watermark". In addition, we have made Figure S39B more clear by replacing the attacked and pre-attached g-value histogram plot with a paired plot showing the 1-1 relationship between each sequence's g-value before (x-axis) and after the attack (y axis).

Requirement of random perturbation during SynthIDBio-structure training

*- It is counterintuitive and potentially concerning that random perturbation is *required* in order to prevent SynthIDBio-structure watermarking from being destroyed by fixed point rounding.*

To explain, consider that we want to save a folded protein as a CIF file. Following the specification [1], atom coordinates are often saved with a fixed number of decimals. During fine-tuning, AlphaFold3's diffusion model learns to add imperceptible noise to the coordinates that can be detected via the watermark detector. There is no constraint on how small this noise is. Without any regularization, we found the model hides the watermark in very small changes that do not survive rounding to 3 decimal points. This is because that's what it is trained to do: maximize quality (via the regular AlphaFold3 losses) while adding a detectable watermark. One approach to encourage the model to learn a watermark that survives this rounding is to add uniform noise in the order of $1e-3$. Note that we show that for $s = 10e-3$, this has negligible impact on quality (in terms of RMSD and many other metrics). There may be other approaches to encourage the model to do this, but we found adding uniform noise to be the easiest. We clarified this in Sections 3 and 10.2.

[1]

<https://pdb101.rcsb.org/learn/guide-to-understanding-pdb-data/beginner%E2%80%99s-guide-to-pdbx-mmCIF>

Different SynthIDBio-structure instances

- Only a single instance of training SynthIDBio-structure is presented. If fine-tuning is performed more than ones, to what degree will different instances be distinguishable? Multiple instances should be trained and compared in order to determine the cross-instance false positive rates.

Actually, we trained 3 instances for $s = 0.001$, 0.01 , and 0.1 . We recommend $s = 0.001$ and most results are thus focused on this setting. We used the exact same setup for all three settings of s , except for varying s (i.e., seeds, data ordering, loss weights, etc. were all the same). This means that the different instances were not explicitly trained to learn different watermarks. Thus, we see mixed results when evaluating the corresponding detectors on the different watermarks. For example, evaluating the $s = 0.001$ model on watermarks from the $s = 0.01$ and 0.1 model shows that watermarks from the $s = 0.01$ model are partially detected. This

makes sense as $s = 0.01$ will result in a slightly stronger watermark, and due to the same training setup, these watermarks may share characteristics. In contrast, running the $s = 0.1$ model on the $s = 0.01$ and 0.001 watermarks shows that these are complementary.

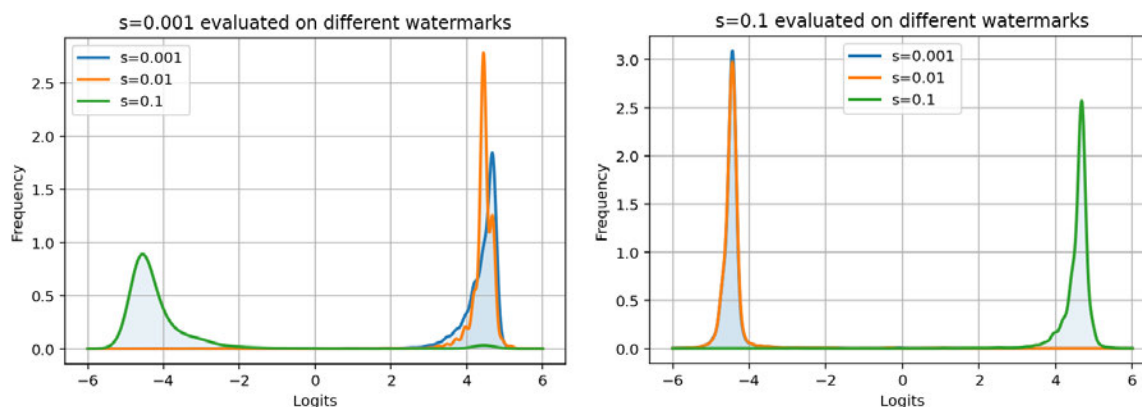


Figure: Logit distributions of the SynthIDBio-structure detector for $s = 0.001$ (left) and $s = 0.1$ (right) when evaluated on structures predicted by SynthIDBio-structure with different values of s .

It is important to note that these interferences are partly expected due to the extremely similar training setup (including same seeds, datasets, etc.). We believe that SynthIDBio-structure can easily be adapted to avoid this if needed: During training, we already inject both ground truth structures as well as predictions from the baseline AF3 model as negatives. We can easily add additional examples to encourage the model to train an “orthogonal” watermark pattern that does not interfere. This is common practice in watermarking.

Value of SynthIDBio-structure

- I do not think I understand the value that is being proposed for SynthIDBio-structure: how does knowing the tool used to generate a publicly shared molecular model help with either biosecurity or scientific integrity? "I used AlphaFold3 to make this model" seems like a statement that nobody would need to try to hide.

We thank the reviewer for this comment. We'll note that the original and revised Section 5.1 (“Biosecurity perspective”) only hypothesize on the value of SynthIDBio-sequence to this space.

In terms of scientific integrity, we do believe knowing what tool folded a protein will become more important the more tools are in widespread scientific use. Irrespective of whether somebody has an incentive to hide whether a structure model was generated by AF3 or not, properly identifying the used tool is and will be important for research. Without that information it will become harder to distinguish whether actual validation through, e.g., crystallography happened or not. This will make proper attribution difficult, may create the wrong incentives, and will also impact future advances in AI for biology as it is expected that training such models on AI-generated predictions may cause problems.

Beyond these two aspects, we also see an interesting use case in IP protection that was also raised by R1. The weights of many protein folding and more and more protein design models are publicly available or made accessible online. While this does usually not include the corresponding training code or data, it has greatly benefited the scientific community [1]. The use of these open models is often subject to non-commercial licenses and approaches such as SynthIDBio-structure could allow proper monitoring and deterrence of license violations. While

this use case remains hypothetical, we believe that our work shows what is currently possible from the watermarking perspective.

[1] Váradi, M., & Velankar, S. (2022). The impact of AlphaFold Protein Structure Database on the fields of life sciences. *PROTEOMICS*, 23.

Fig. 1 improvements:

- *Fig. 1a: The watermark detector is more than just a threshold - it needs sampling data and the key that was used for generation, which needs to be transmitted between the generator and detector.*
- *Fig. 1b: Shouldn't the two watermark detectors be the same color?*
- *Fig. 1b seems to conflate together the model training and model application stages. If I understood correctly, a model user should not have access to the watermark detector module, else they could use it to undo the fine tuning.*

These are all great suggestions to improve Figure 1. We note that, for SynthIDBio-sequence, the detector does actually not need sampling data, it just needs the threshold, which has been calibrated before, and the secret key. We made this clear in the updated figure. Regarding SynthIDBio-structure, the watermark detector co-trained with AlphaFold3 is clearly the same detector used at test time. We also indicated in the caption that the detector (or the secret key for SynthIDBio-sequence) are not shared with users.

Fig. 2 improvements:

- *Fig. 2a,e would be much clearer if presented as violin plots.*
- *Fig. 2c: Is the third line meant to be distortionary 0.1, rather than non-distortionary 0.5? It looks like it's meant to show one of each case. If you're showing multiple distortionary 0.5, there is vertical space enough to show more than 2.*

We appreciate the detailed feedback and updated Figure 2A and E to be violin plots. Regarding Figure 2C, this is indeed supposed to be non-distortionary 0.5. This is because distortionary watermarking leads to more extreme g-values for individual amino acids. This means that the individual amino acids are either blue (0.5) or dark red (high values). Adding these to the same plot would have shifted the color scale for non-distortionary 0.5. We show examples for distortionary 0.1 in the supplementary material Figure S21 (previously S9).

Fig 4 and 28 improvements

- *The "Detectability and accuracy" graphs in Figs. 4a and S28a are extremely difficult to read and do not appear to match one another. Separate the two metrics into different panels, use consistent axes, plot with more readily distinguishable colors, and use line graphs instead of bar graphs.*

Thanks for the feedback to make these figures better. We want to emphasize that Figures 4A and S28A (now S43A) do *not* show the same results on the right y-axis. Figure 4A shows IDDT and TM score, while original Figure S28A (now S41A) shows RMSD. We did, however, find an error in our plotting, resulting in the TPR numbers for original Figure S28A being assigned incorrectly to models. Below (left), we include the original plot with correct assignments.

As per suggestions, in the revised paper, we split up the plot into a bar plot and a line plot. The reason for keeping a bar plot for TPR is that lines would largely overlap and with TPRs being between 0 and 99.9 (see below. left), this would thus be very difficult to see. We made colors more bright, but kept the pairing of lines to make clear which lines should be compared.

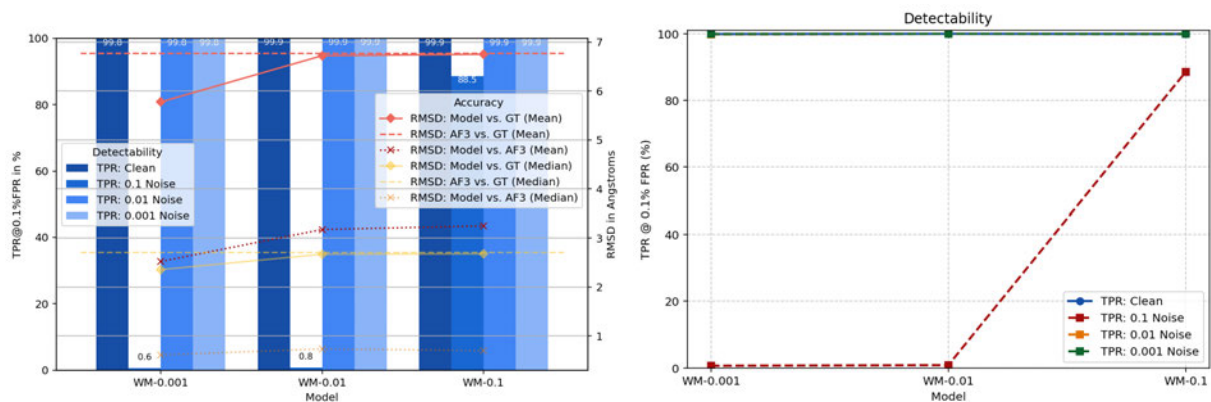


Figure: Left: The original plot from the original Figure S28A (now Figure S41A) with corrected numbers. Right: An example plot of plotting detectability using line plots, where it is very difficult to recognize exact numbers.

- Fig. S28a is missing the bars for 0.001 and 0.01 noise.

These are actually not missing, but they are indeed very close to 0. We corrected the plot by indicating the numbers on top of the bar. As shown above, we moved the numbers to above the bars.

- Figs 4a, S28a, and S28b would be much clearer if they were converted to line graphs with the number of AAs on the x-axis.

Thanks for the suggestion, you can find clearer line plots in the revised version, Figure 4A and Figure S41.

- Figs 4b and S28e should plot distributions with lines rather than semi-opaque areas so that it's easier to see the actual values.

Following your suggestion, we used Gaussian kernel density estimation (scipy.stats.gaussian_kde with a factor of 0.15) to redo these plots. All plots can be found in Figures S42-S45 in the appendix. Figure 4B has also been updated accordingly.

- The detectability rates shown in Figs. 4a (right) and S28b do not appear to be compatible with what is down in 4a (left) and S28a. Some of that may be due to the problems in the presentation of these graphs, however.

Yes, this is right. This is because in Figure 4A (right), the indicated average is a balanced average across the different length buckets. As the distribution of structures over these buckets is not uniform, this does not match the averages reported in Figure 4A (left). We removed these averages to avoid confusion while also moving to line plots as suggested above.

Response to reviewer 3

SynthIDBio as additional layer for biosecurity

Biosecurity is an increasingly discussed topic with many proposed interventions, yet none that are 100% foolproof; and that is okay. Each intervention has its strengths and weaknesses, but the cumulative effect of layering a set of flawed guardrails can lead to a resilient system. I think it is important to frame any biosecurity intervention in this context; so, rather than focusing primarily on where SynthIDBio works well, equal attention should be given to where it does not. In other words, discussion should provide more focused discussion on the specific elements of the biosecurity environment that SynthIDBio both does and does not support, and, further, how the method fits together with other proposed interventions. I think that Section 5.1 provides an excellent starting point for this type of reframing, where a potential synergism between watermarking and DNA synthesis screening is proposed. The paper would be strengthened if use cases for watermarking (like the one in the aforementioned section) were more richly worked through, with a focus on realism and feasibility (i.e., How realistic is the use case in question?) and robustness (i.e., How resilient would watermarking be in the proposed usage?).

We fully agree and added a more detailed discussion in Section 5.1 with a more concrete proposal. This largely mirrors your framing of biosecurity as multiple layers of guardrails of which watermarking could be one. The strengths and weaknesses of SynthIDBio that we outline in our response and the updated manuscript are also reflected in this proposal advocating against a punitive implementation. Please see our response “Relevance of watermarking for biosecurity” for a detailed discussion.

Adoption of SynthIDBio by open source tools

Most bio design tools (BDTs) are open source, including, of course, ProteinMPNN, which was the BDT primarily used for testing SynthIDBio-sequence. How do you implement watermarking in such an environment? A malicious actor would likely just use an open-source tool that did not use watermarking.

Similar to the above, a sentence or two on the generalizability of SynthIDBio should be added. As written, it appears to me that SynthIDBio is not as easy to use as simply wrapping an existing model implementation with a watermarking implementation—some level of modification to underlying code and, at least in the case of the structure version of the tool, model fine tuning will be required. Is this correct? If so, some discussion regarding the challenges of encouraging/promoting uptake should be added, as I would imagine many (if not most) tool developers will not be motivated to do the extra work.

Please see our response “Encouraging watermark adoption, especially in the context of open weight models” that addresses both concerns. We added this discussion to Section 5.3.

Cost of SynthIDBio-sequence

It is mentioned in the text that SynthIDBio introduces additional overhead to the generation pipeline, but this is not quantified anywhere. Ideally, quantification via a few metrics should be added, including GPU hours and monetary cost. I think that Figure 3B gets close to this, as it shows the decrease in pass rates when designs are filtered by g-value as well. Perhaps a companion figure could show the increase in cost or compute required to achieve the same rate of passing designs when using g-value filtering versus when not.

Please see our response “Cost of SynthIDBio-sequence” for R2. We added Figure S35 translating pass rate reductions to additional compute required. In A.5, we additionally give concrete numbers showing that the increased cost for SynthIDBio-sequence is small and does not scale with FPR stringency as the majority of cost comes from structural validation via AF3 which we do not need to run on significantly more candidates than without watermarking.

More details on resequencing attack

In Section 4.3, information should be provided on how the resequencing attack was performed. Additional discussion should be added on how easy (or not) this would be to do. My sense is that anyone capable of using BDTs would be able to launch such an attack.

You are correct in assuming anyone capable of using a BDT tool would be able to carry out the attack. As now specified in the methods, we performed the resequencing attack by running the non-watermarked ProteinMPNN (with default 0.1 temperature) on a watermarked binder in complex with its target. Supplementary Figure S39A shows the effect of this attack on a sample of ~38,000 distortionary 0.5 sequences, after the attack only 15 remain detectable at 0.1% FPR.

SynthIDBio-structure robustness to relaxation:

For Section 4.5, attacks using a relaxation protocol or some other energy minimizing protocol should be benchmarked. As of now, there is only brief mention in the discussion of how the authors expect such an attack to remove the watermark. Assuming the authors' hypothesis regarding these attacks is true, discussion over what this means within the broader context of biosecurity should be added.

Please see our response "SynthIDBio-structure robustness to relaxation".

Discussing related work in the introduction.

Related work on watermarking biological designs is introduced in the discussion ("Concurrent work also considers watermarking..."), when it would more appropriately be first referenced in the introduction. The second paragraph of the Introduction seems a reasonable place to put this.

We agree that relevant related work should be discussed earlier in the paper. We moved part of the discussion to the introduction and left a more detailed commentary in the discussion Section 5.4.

Figure 4B shows worst-case examples of the impact of watermarking on feature distribution.

On pg. 10, there is the statement "However, the examples shown in B are meant to illustrate the worst case." In what sense are these examples meant to illustrate the worst case? More explanation is needed.

We appreciate that this statement is misleading and rephrased it appropriately. In the main paper, we wanted to highlight features (i.e., the C-CA distance distribution in MET) that were particularly impacted by watermarking. In this sense, worst-case means to say that these shifts in feature distributions are particularly noticeable and more noticeable than for many other features. This is illustrated in Figure S42 where we show many other features, illustrating that *typically* watermarking has a significantly smaller impact for $s = 0.001$. Note that we also included update plots of feature distributions in the appendix emphasizing this point, specifically Figures S42 to S45. We clarified this in Sections 4.4 and B.3.

Support for rigid transformation robustness of SynthIDBio-structure.

At the top of pg. 11, there is the statement "It is also worth emphasizing that, by construction of our watermark detector, the watermark is robust to rigid transformations of the protein structure." A supplementary figure should be added to support this claim.

We believe that a supplementary figure is not necessary. This is because the detector is indeed robust to these transformations *by construction* as the inputs to the detector are only features invariant to rigid transformations. With that we mean the following: As detailed in Section 10.1, the detector's input are intra-residue distances (i.e., pairwise distance between all atoms in a specific residue) and intra-residue torsion angles. The latter are computed in a sliding window fashion, meaning we compute the torsion angle for each window of 4 consecutive atoms. The detector then does several per-residue 1d convolutions with ReLU non-linearity, computing per-residue features. These features are then aggregated by average pooling. The computed input features are clearly invariant to rigid transformations (meaning that rotations, translations and reflections do not change intra-residue atom-atom distances or torsion angles). This means the rest of the detector computation is entirely invariant to rigid transformations, as well. Moreover, by doing per-residue computation and average pooling per-residue features, the detector is also invariant to the ordering of the residues. We clarified this point in Sections 3, 4.5, and 10.1.

Response to reviewer 4

I co-reviewed this manuscript with one of the reviewers who provided the listed reports.

Please see our responses to R1 to R3 above.

Response to Reviews for Manuscript

“Function-preserving biological watermarking of AI-generated protein sequences and structures”

We thank the reviewers for their feedback. All reviewers agree that our rebuttal addressed the majority of the raised concerns. R1 highlights that proof-of-concept studies like ours are “undeniabl[y] valuable contributions to science”, R2 found the revised version significantly clearer and particularly highlighted our cost analysis, and R3 highlighted our revised proposal around biosecurity and that the paper is “suitable for publication”. In the following, we will address the remaining comments in detail. We note that with the addition of new supplementary figures, the figure numbering has shifted slightly. We have not modified the original reviewer comments in light of these numbering changes.

Response to R1

Rephrasing our work as proof of concept

The authors have substantially edited and appended the manuscript top address the concerns expressed by all reviewers, and have also shared code and (raw) data as requested by myself and other reviewers. I also appreciate the effort the authors have put in highlighting all edits and overall presenting a very clear and concise rebuttal.

The technical concerns raised by the reviewers have in principle been addressed. In some cases the revised manuscript clearly acknowledges weaknesses and shortcomings of the proposed procedure, which is commendable. However, this ultimately contributes to the impression that the procedure as presented here is not really useful (Except for protecting IP, which is not the focus of the manuscript). The authors themselves state in the manuscript:

“Ultimately, however, we consider this a proof-of-concept that function preserving watermarking of protein sequences and structures is possible.”

This statement is undeniably true, and proof-of-concept studies are undeniably valuable contributions to science. Whether this proof-of-concept is enough to justify publication in Nature is for the editor to decide, but I would recommend treating the manuscript as exactly this, a proof-of-concept, and not a ready-to-use method (as one might have expected from these authors).

We thank the reviewer for their positive assessment of our technical revisions and for their insightful perspective on the manuscript's framing. We agree that the primary contribution of this work is a technical proof-of-concept. As the reviewer notes, establishing that function-preserving watermarking is technically feasible for both protein sequences and structures is a necessary and impactful first step toward more robust provenance systems.

Following the reviewer's recommendation and the Editor's guidance, we have updated the manuscript to emphasize the proof-of-concept nature of this study. We have edited claims

appropriately and highlighted that our discussion in Section 5 is about potential future use cases.

Usefulness despite imperfect robustness

Other reviewers and I had been concerned that the original manuscript overestimated the robustness of watermarking against removal attempts in the form of resequencing (SynthIDBio-sequence) or relaxation (SynthIDBio-structure). The authors now acknowledge the lack of robustness in both cases, which is commendable. However, this rules out all potential use cases which would require watermarks to be irremovable (such as the cases discussed in section 5.2 of the manuscript).

Which leaves the question of how and where these watermarking procedures could be applied (except for protecting IP). The authors have amended the manuscript to address this question and expanded on it in the rebuttal, but I honestly still don't understand the underlying logic.

We appreciate the reviewer's candid feedback regarding the practical utility of SynthIDBio in light of imperfect robustness. In the following, we argue that we believe SynthIDBio to be useful for potential applications even without perfect robustness. Specifically, we believe that the watermark robustness needs to fit the potential applications. The level of robustness determines the severity of transformation required to change the protein significantly enough such that maintaining the original provenance information is not meaningful.

In the context of our “trusted tool” proposal for biosecurity (also see below), the level of robustness encodes which transformations will preserve the “trusted tool” label. In this setting, a very high level of robustness may be counterproductive. If the watermarked protein sequence changes to an extent that may induce a change in function, we would want the watermark to be removed. This ensures that the “trusted tool” verification signal remains valid only as long as the design matches the function it was intended for and vetted for by the original model's safety filters. Concretely, using another, non-vetted resequencer on top of the design should thus remove the watermarking signal since there is a reasonable chance that function changes (see Figure 3D). We have modified section 5.1 to acknowledge this indirect benefit of imperfect robustness. Moving away from the punitive implementation outlined in our original submission towards this “trusted tool” verification signal allows synthesis providers to lower operational costs for ambiguous orders, which in turn incentivizes good actors to adopt watermarked, trusted tools.



In other settings such as IP protection or scientific integrity, we believe that SynthIDBio is already robust enough to make a positive impact if operationalized properly. We can see this in the generative text and multimedia domain. For example, SynthID-text is currently used to watermark Google's flagship model Gemini, providing critical provenance information and veracity signals. However, as with SynthIDBio-sequence, robustness of text watermarking schemes are actively researched and robustness remains limited in the context of regeneration

attacks [1]. Moreover, we expect that future innovation in terms of more robust text watermarking methods may be transferrable to protein design based on our paper's proof of concept. A similar argumentation holds for SynthIDBio-structure, which is more robust in general but also lacks robustness to relaxation or regeneration style attacks. This is similar to existing work on image watermarking where re-generation attacks remain challenging [1]. Nevertheless, when deployed at scale, image watermarking holds promise to, e.g., address misinformation. SynthID-image, for example, is actively used by prominent news outlets to fact check and identify potential misinformation, for example in BBC's Verify series [2, 3, 4]. We believe that a similar effect, despite imperfect robustness, can benefit major protein databases, as well.

[1] <https://neurips.cc/virtual/2024/poster/96428>

[2]

<https://www.bbc.co.uk/news/bbcverify#:~:text=BBC%20Verify%20Live%20Friday%202027%20February:%20Misinformation.protests%20in%20Iran%20spreading%20to%20more%20universities>

[3] <https://www.bbc.co.uk/news/live/cx28nzdkwijt> (search for "SynthID")

[4] <https://www.bbc.co.uk/news/live/cvg3qzx512nt> (search for "SynthID")

Trusted model proposal for biosecurity

The authors state in the rebuttal:

"We believe that building good incentives for users to use "trusted", watermarked models and these models being state-of-the-art can go a long way in driving adoption."

What is a "trusted" model, and why should a DNA synthesis company prefer sequences obtained from a trusted model? If a malicious actor uses proteinMPNN to resequence toxins or pathogens or any "harmful" protein, why would it matter if a "trusted" version was used? How does identifying the specific resequencing tool help to distinguish between harmful and not harmful? I apologize if I am missing something obvious here, but I cannot think of a single concrete example where this identification of the resequencing tool would help to decide what should or should not be synthesized. Section 5.1 and the corresponding parts of the rebuttal sound as if the authors had a specific scenario in mind - maybe it would help to describe this in detail?

We thank the reviewer for highlighting that this concept was underdescribed. We envision "trusted" tools to refer specifically to pre- or post-deployment mitigations and guardrails that limit the ability to use these tools maliciously. We've added a description of the "trusted" model to section 5.1 to clarify these points:

"Here, watermarking could serve as a verification signal that a "trusted" tool which a third-party verifies contains sufficient mitigations and guardrails to prevent nefarious use was used to generate an order (e.g., a resequencer deployed via an application programming interface (API) that blocks input structure parent sequences which align to a database of harmful proteins). Establishing processes for this verification is outside the scope of this paper."

Response to R2

Thank you to the authors for the work that they have done clarifying figures and adding new information to fill in important gaps. The new version of the manuscript is significantly clearer and more convincing in a number of aspects, and a number of my concerns regarding the first version have been well addressed. Among other things, the cost charts are excellent: they give clear intuitions of tradeoffs and bottlenecks, and address all of the concerns that I previously had in that area.

We thank the reviewer for these encouraging comments. We are pleased to hear that the revisions and the additional information provided, including the detailed cost analysis, have addressed many of the raised concerns.

Biosecurity value of sequence watermarking

The authors have proposed to flip the value here to focus on watermarks as indicators of production by a trusted tool. Since such detection already requires secret key exchange, and since users of a trusted tool are incentivized to provide evidence that they should be trusted, why does sequence watermarking have any advantage over standard PKI approaches like certificate-based signing of a digest value in accompanying metadata?

We believe there are several advantages of watermarking compared to (encrypted) metadata. As discussed in Section 5, metadata is trivial to remove and encryption also requires coordination. While public-key encryption makes this easier, future advancements in watermarking could enable similar strategies (as discussed in Section 5.3). Even if the tool user is incentivized to provide encrypted metadata, metadata being easy to remove remains a problem. We can observe this being a problem in AI-generated media. For example, while many companies adopted the C2PA standard (a metadata-based standard for provenance, especially in the context of AI generated media), OpenAI states that C2PA is not reliable and commonly removed by many social media sites [1]. We observe the same with our work on SynthID-image [2] which is already used for fact-checking where C2PA is not available. We anticipate a similar situation in AI for bio. Even if a tool user wants to include metadata, it is difficult to guarantee that this metadata is preserved in all the processing steps and tools that may come before or after submitting to a database/synthesis provider. This would require all other tools used in the process to respect the metadata protocol. A watermark is not as easily removed. Tracking edits in metadata is also more involved, while watermarking can naturally be applied to parts of a sequence/structure (just like it is applied to parts of edited images [3]).

We note that Section 5.1 includes acknowledgement of the value of cryptographic signatures: “It is important to note that this practice could also be enabled by tool-specific metadata paired with cryptographic signatures.”

[1] <https://help.openai.com/en/articles/8912793-c2pa-in-chatgpt-images>

[2] <https://arxiv.org/abs/2510.09263>

[3] <https://blog.google/feed/synthid-reimagine-magic-editor/>

Biosecurity value of structure watermarking (and value for databases)

The authors claim that watermarking will help to identify whether crystallography was actually carried out (versus just in silico work). I still do not understand how this would be true.

Crystallography results would not have a watermark. Non-trusted tools produce no watermark and relatively small random perturbations destroy a watermark, so how are either of those cases distinguishable from real crystallography?

We first note that we do not claim direct biosecurity utility for our structural watermarking method, as suggested by the reviewer's title for their raised question. This is also reflected in our manuscript's focus on SynthIDBio-sequence for that domain, specifically.

Regarding the reviewer's questions on distinguishing AI-generated models from physical crystallography data within scientific databases, the SynthIDBio-structure watermark will allow identification and verification of AI generated structures. First, we note that SynthIDBio-structure has been shown to be robust to random perturbations, as shown in Figure 4a and the appendix. This confirms detectability of the watermark and differentiation from real crystallography and non-watermarking structure prediction tools. Thus, uploads of watermarked structures can be categorized accordingly, potentially saving time or allowing clearer categorization within the database. While unmarked structures could come both from real experiments or "untrusted" tools, we believe that is a strong incentive for major tool developers to integrate SynthIDBio to ensure proper attribution. With growing adoption, this additional database signal will get clearer and more reliable.

Entropy / sequence diversity

The new entropy and diversity results are quite interesting and contain some surprising and counterintuitive results. I very much appreciate the authors' exploration of this space.

That said, I find the results in Figs S10 and S11 concerning. The key point I see is that distortionary 0.1 results are almost entirely non-conserved with respect to the original sequence and highly conserved with respect to one another. This, in turn, I believe means that the entropy results in Figs S8 and S9 are telling us very different things about the non-watermark and distortionary 0.1 distributions. For non-watermark, we are seeing exploration of protein space around the original sequences, while for distortionary 0.1, we are seeing exploration of protein space around a very different consensus sequence. This view is also consistent with the example sequences shown in Fig. S21.

This leads me to wonder the degree to which ProteinMPNN is even remaining a significant aspect of the design, versus the watermarking effectively replacing ProteinMPNN with a different diversity generation technique in which the primary determinant of sequence variation is the randomly chosen encryption key. What would happen if ProteinMPNN was replaced with a different tool, such as EvoDiff, RFDiffusion, but the watermarking key was kept the same? Would the difference between watermarked outputs of different tools be less than the difference between watermarked and non-watermarked outputs from the same tool?

A milder form of the same concern applies for non-distortionary 0.5.

We thank the reviewer for their thoughtful engagement with our entropy and diversity results. We appreciate the opportunity to clarify this mechanism, and we have conducted additional analyses (detailed in our updated manuscript and new/updated Figures S10-S13) to explicitly address the diversity concerns.

Clarifying the Extent of Conservation:

The reviewer is correct that the data displayed in Figures S10 and S21 would give the impression that distortionary watermarked sequences are highly conserved with respect to one another. However, this view is inconsistent with Figures S6-S9, which are based on population-level values rather than individual sequences. In order to further probe this question, we have completely revised section A.4.2 with new analyses focused on the rates of per-position conservation:

- On a per-position basis, watermarked sequences display significantly lower rates of conservation to the original parent sequence and, importantly, to their within group consensus sequence compared to the non-watermarked sequences (Figures S10, S11)
- Regarding the previous PD-L1 Backbone 2 conservation plots (originally S10, S11), we have replaced these with Figures S12 and S13 which show the minimum and maximum conservation profile with respect to the consensus sequence and parent sequence. The primary motivation for replacing these plots was a recognition that a single backbone was unlikely to be representative of the population dynamics. In these new plots, it is demonstrated that both non-watermarked and watermarked generation settings are capable of producing sequences with very high or very low conservation levels relative to a parent sequence (Figure S12) or the consensus sequence (Figure S13). We note that the values, which couched in the reality of displaying extrema, are consistent with non-watermarked sequences having slightly higher rates of conservation than watermarked sequences.

Our interpretation of Figures S6-13 is that watermarked sequences are exploring a genuinely broader sequence space than non-watermarked sequences rather than a similarly narrow space anchored on a different consensus sequence.

The Role of the Generative Model vs. The Watermark Key:

Regarding the reviewer's concern that the encryption key might be acting as a replacement diversity generation technique: it is vital to note that our watermarking algorithm does not generate sequence tokens *from first principles*. Instead, they act as a bias applied directly to the logit distributions outputted by ProteinMPNN. ProteinMPNN's learned physical and structural priors dictate which residues are strictly required (high logit values) and which regions are permissive (flatter logit distributions). The watermarking key biases the sampling exclusively within these permissive, biologically viable token spaces. Because the algorithm only introduces variability where ProteinMPNN allows it, the structural and functional viability of the protein is preserved (as proven in their designs meeting strict structural similarity constraints with the input backbone structure, which is what ProteinMPNN is trained to do).

Application to Other Models:

To answer the reviewer's question regarding if this is applicable to other models: if ProteinMPNN were replaced with an alternative resequencing model and the exact same watermark key was used, the outputs would almost certainly be entirely different from the ProteinMPNN outputs. Because the key relies on the underlying model's specific logits to guide the biased sampling, the final sequence is inextricably linked to the base model's distinct architectural priors and sequence distributions. Note that RFDiffusion uses ProteinMPNN – ProteinMPNN is the de-facto standard in state-of-the-art protein design pipelines, and we show RFDiffusion transferability in Section A.11. It is also important to note that this protein sequence watermarking technique is applicable to any autoregressive machine learning model, there is nothing specific about the watermarking that makes it only applicable to ProteinMPNN. EvoDiff is not an autoregressive model and uses discrete diffusion, adapting watermarking methods such as SynthID-text to discrete diffusion is possible but an active area of research, see [1].

[1] <https://openreview.net/forum?id=3aBWTYGcaT>

Sequence watermarking false positives, false negatives, and natural sequences

The results in Figs S2 and S3 showing false positive rates against natural sequences are extremely helpful, and I thank the authors for including them. Order of magnitude lower false positive rates is very good news as a starting point.

Unfortunately, however, as the authors also note (correctly, I believe) a significant component of this lower FPR is due to the longer sequences. The authors also correctly note that adding adjacent non-watermarked material would readily spoof the system, and that for this reason it would be important to compare not whole sequences but windows.

What would happen to Figs S2 and S3 if the test were performed as an "apples to apples" comparison using a scanning window? Taking a window of length 60, matching the length of some of the engineered proteins worked with by the authors, and a 50% tiling (a bare minimum requirement for screening), a typical 400 amino acid natural protein would be tested on 13 different windows, and return a false positive if any of those 13 had a high g-value. Combine that with the fact that shorter windows will produce a broader g-value distribution, and the FPR may end up as bad or worse than the target.

I think that this is an important thing to check, if the authors wish to know what the FPR/TPR tradeoff is likely to actually be in the scenarios they envision.

We agree that a naive application of the full-length sequence calibrated thresholds would likely underestimate the real-world FPR rate for shorter sequences. In order to empirically measure this and determine a closer to real-world FPR/TPR tradeoff, we implemented the scanning window analysis as suggested (window length of 60 with a 30-residue overlap between adjacent windows) and quantified the resulting FPR shifts.

Our analysis yielded the following key insights:

- Naive thresholds underestimate the FPR: As shown in Table S5 and Figure S17, applying baseline UniRef50 full-sequence thresholds directly to windowed sequences does empirically increase the FPR, at least partially due to the broader variance of shorter sequences as noted by the reviewer.
- Our paper's thresholds remain robust: Crucially, when compared against the primary thresholds established in our paper, we observe little to no change in the empirical FPRs (Table 4, Figure 16). This robustness exists because our paper's thresholds were already calibrated using large datasets of short synthetic sequences from design models.
- Applying our pre-established thresholds to the g-value distribution obtained from length-60 UniRef50 windows results in negligible drops in the target and empirical FPRs. As updating calibration thresholds is computationally trivial (as demonstrated in our cost trade-off analysis), our method can seamlessly adapt to more screening-relevant FPR thresholds without sacrificing performance.

To address this thoroughly in the manuscript, we have added a dedicated section to the supplementary text detailing this scanning window evaluation (section A.7), accompanied by Figures S16 and S17 and Tables S4 and S5, to provide full transparency on threshold recalibration and variance.

Impact of structure watermarking on distributions

The authors claim that the impact of structure watermarking on accuracy is not high because Figs 4a and S41b show that the RMSD values versus ground truth are similar for both watermarked and non-watermarked structures. Figs 4b and S42-S45, however, show that in many cases these distributions are highly bimodal, and in some cases cases the watermarking strongly distorts or eliminates a mode relative to both ground truth and non-watermarked. Is systematic elimination of a mode of distribution really negligible in impact on accuracy, even if bulk statistics are not much affected? Perhaps these are critical points in the protein. It would be interesting to inspect a few of the affected atoms and see if there is a high impact on local regions of the structure, even if the overall structure is relatively intact. The fact that the authors acknowledge that WM-0.1 has a high impact on distributions but that the bulk metrics show barely any difference between WM values indicates that those bulk metrics are not telling the full story on accuracy.

While aggregate statistics show no impact, we agree that there is a visible impact in some metrics. However, we believe that this averages out across metrics and we present additional evidence for this below. Beyond this, we also share selected protein structures as .pdb files, including SynthIDBio-structure $s = 0.001$ and AF3 predictions as well as ground truth structures.

Regarding systematic loss of a mode, when inspecting the $s = 0.001$ results (Figure S46) this is not a watermarking issue, but rather an AF3 issue that is inherited. We can see in all instances (for $s = 0.001$) where a mode is lost with regards to ground truth (GT), it is lost in both AF3 and watermarking. With respect to mode loss compared with AF3, this was only an artifact seen in

stronger watermarking settings such as $s = 0.01$ (Figure S47) and not observed in the recommended watermarking strength of $s = 0.001$.

For example, Figure S46 (top left) shows that watermarking increases the CA-N distance in MET, moving the distribution further away from the ground truth. Figure S49 (top left and top middle) shows that there are equal examples where watermarking moves the distribution closer to the ground truth for C-CA in ALA and C-CB in ALA, albeit in one case losing the bimodality (top middle). Attempting to quantify this, we computed the Wasserstein distances of these histograms. For example, for atom-atom distance distributions, we find that in 72.5% of the (amino-acid, atom, atom) triples the Wasserstein distances between SynthIDBio-structure vs. GT and AF3 vs. GT are not statistically significantly different. For torsion angles, this number is 96.9% and for bond angles it is 47.2%. Moreover, considering the average distances across all these features, they are not significantly different between SynthIDBio-structure and AF3. This means that in a bootstrap analysis, averages were within 2 standard deviations. This supports that, in expectation across these features, the distributions do not change in a statistically significant, predictable way, even though in individual cases significant shifts (both for the worse or the better) are possible.

In order to share a representative sample of targets, we considered Figure S50 which shows the distribution of IDDT between SynthIDBio-structure $s = 0.001$ predictions and AF3 predictions. We selected one target from each of these buckets: [95, 99), [90, 95), [85, 90), [80, 85). For each bucket we picked the target with the highest IDDT and a target with low similarity to the training data (<40% template similarity for a chain in an interface between two chains). We report the corresponding IDDT's in the table below and shared the structures with the editor. We also include visualizations in Figures S52 to S60.

IDDT Ours vs. AF3 Bucket	% of structures in bucket	Target	IDDT Ours vs. AF3	IDDT Ours vs. GT	IDDT AF3 vs. GT	IDDT AF3 vs. GT STD across samples
99 - 95	45.84%	7PPA	99.00	88.85	88.82	0.7923
99 - 95		7X7A	96.30	71.52	71.10	
95 - 90	5.17%	7P25	95.00	89.61	91.26	0.9918
95 - 90		7XSP	94.53	71.45	70.58	
90 - 85	0.86%	7SJN	89.92	77.45	77.27	1.3045
90 - 85		7Y81	88.13	67.70	68.06	
85 - 80	0.33%	7UZP	84.77	67.55	70.34	2.2781
85 - 80		7OIX	84.63	68.35	69.07	

We additionally included these targets which are featured in the AF3 paper: 7PZB and 7AU2. We added a detailed analysis of the table above into Section B.5.

Differentiability of structure watermarks

The authors appear to have misunderstood my question about different instances of structure watermarking, as they have compared their different My question is about what happens if we want to distinguish more than one "trusted tool" by watermark. Let's say that there are five different organizations that each run a "trusted AF3" tool, each of which has been independently fine-tuned with $s=0.001$. What does the confusion matrix look like for correctly interpreting a watermark as having been applied to Organization A vs Organization B vs Organization C, etc.?

We thank the reviewer for following up on this. Based on our experience with SynthID-image which has been deployed at large scale in a modality where there are multiple open-source image watermarking models available, there are two routes to make this possible. First, it is likely that there will be multiple different watermarking approaches available, potentially developed separately. In this case, fine-tuning the detector to ensure differentiability is typically sufficient. Since the detector is small and fine-tuning the watermark (in our case, the fine-tuned AF3 model) is not required, this is typically quick and light on resources. Second, we can move towards multi-bit watermarking where the watermark signal not only encodes a binary signal (watermarked or not), but actually a message consisting of multiple bits. This often makes sense when doing post-hoc watermarking (i.e., where the mechanism to embed the watermark is entirely separate from the folding model) since the same watermark can then be used to add watermarks to multiple tools, differentiating them through different bit codes. As we targeted an open-source setting, we did not pursue a multi-bit approach though. We acknowledged this in Section 5.5.

Minor points

- *For structure, some places say "s" and some way "WM". I think they are the same, but I found it confusing.*
- *The s value for Fig. 4b is not given.*
- *Figs 3a,b, S35, S37, S38, S40, S41a: Please do not use nearly-identical shades of the same color to distinguish different conditions. There are lots of good color-blind-friendly options that will make these figures much easier to read.*
- *Figs 3b, S40b: Put a gap between the bar clusters. They are running into one another right now.*
- *Fig. S41a: The bars are out of order. Clean is equivalent to 0, so the others should be in ascending order: 0.001, 0.01, 0.1, not the current non-monotonic order.*
- *Fig. S41f is very interesting! I'd like to understand better what it's telling us, but the graphs and text say there is noise=0.1 added and the captions say there is no noise added. Which is it?*
- *Fig. S41f: Order the WM values in the caption - right now they are out of order.*

We appreciate the detailed review and feedback to improve our figures. We addressed all the raised concerns.

Regarding Figure S41f, we want to clarify that these figures are indeed evaluated with Gaussian noise of standard deviation 0.1 as highlighted in the plot titles. Essentially, these results show that we cannot get good detectability at this noise level without also training at that noise level with $s = 0.1$. However, even when doing this, detectability reduces considerably for shorter sequences.

Response to R3

Allowing developers to use SynthIDBio-structure

We thank the authors for taking the time to address our concerns. The proposal for motivating adoption seems reasonable as a method adapting methods developed with language models to protein biology. We appreciate the more nuanced discussion regarding where SynthIDBio fits into the broader biosecurity ecosystem. Given the changes made, we feel that the work is suitable for publication. Before that, however, we want to suggest an important update. This change should not preclude publication, but addressing the following would increase the impact of the work:

The “trusted model” system that is proposed is a good idea, but it of course relies on DNA synthesis providers all agreeing to adopt a policy of preferential treatment of “trusted” orders. This should not be too difficult to achieve, so long as use of watermarking techniques does not significantly burden their customers. What would be most burdensome for customers would be a lack of options: Customers will want to be able to select from a wide array of “trusted” models. There can be wide availability of trusted models only if developers are encouraged to include SynthIDBio or similar as part of their models/generation procedures. This should also not be too difficult to get model developers to do, so long as it is easy to implement SynthIDBio as part of their projects. Thus, the impact of this work is maximized if it is SynthIDBio is made as easy as possible for developers to incorporate. Given this, the SynthIDBio code base should include the infrastructure needed to apply SynthIDBio to tools beyond those studied here. For SynthIDBio-sequence, this would ideally take the form of a class decorator, a wrapper function, a base class, etc.--something that can be dropped in to anyone else’s project to apply watermarking to their model with next-to-no additional effort. This would admittedly be more difficult to do for SynthIDBio-structure, owing to the requirement for retraining, but programmatic infrastructure could still be provided to streamline its application to others’ models (though we would imagine that more would be needed than, e.g., a simple decorator).

We appreciate the positive evaluation of the proposed "trusted tool" framework. We agree that maximizing scientific impact and biosecurity utility requires a low barrier to entry for model developers.

Regarding SynthIDBio-sequence: For the large number of existing protein design pipelines that already utilize ProteinMPNN, we have designed our open-source release to act as a direct, drop-in replacement. Our code adapts the original, most widely used PyTorch implementation of ProteinMPNN, carefully preserving its core API and functionality. Consequently, developers can swap our watermarked version into existing state-of-the-art pipelines with virtually no structural

code changes. We have updated our repository documentation to provide a clear example of this drop-in replacement.

Second, to address the reviewer's important point regarding the application of watermarking to entirely different tools (e.g., other autoregressive protein sequence models), we have provided the programmatic infrastructure requested. Because our method leverages the open-source SynthID-text library—which operates directly on the output logits of a model—it is inherently modular. Taking the reviewer's advice, we have added a generic wrapper function/class decorator example to our repository. This wrapper demonstrates how developers can import the watermarking logic and apply it to their own custom model's generation loop with, as the reviewer aptly stated, "next-to-no additional effort."

Regarding SynthIDBio-structure: We acknowledge that applying watermarking to structure models is more involved due to the requirement for fine-tuning. We have done this for one of the most popular and state-of-the-art models, AF3, and will release the fine-tuned model as previously shared with the reviewers. To further facilitate adoption for other models, many of which are based on diffusion components, we described losses and fine-tuning mechanisms in detail in Appendix B.1, in the same notation as the AF3 supplementary material. We amended this description by adding details on our detector architecture.