arXiv:2402.07148v2 [cond-mat.soft] 30 Mar 2024

X-LoRA: Mixture of Low-Rank Adapter Experts, a Flexible Framework for Large Language Models with Applications in Protein Mechanics and Molecular Design

Eric L. Buehler[*]

Markus J. Buehler[†]
mbuehler@MIT.EDU

Abstract

We report a mixture of expert strategy to create fine-tuned large language models using a deep layer-wise token-level approach based on low-rank adaptation (LoRA). Starting with a set of pre-trained LoRA adapters, our gating strategy uses the hidden states to dynamically mix adapted layers, allowing the resulting X-LoRA model to draw upon different capabilities and create never-before-used deep layer-wise combinations to solve tasks. The design is inspired by the biological principles of universality and diversity, where neural network building blocks are reused in different hierarchical manifestations. Hence, the X-LoRA model can be easily implemented for any existing large language model (LLM) without a need for modifications of the underlying structure. We develop a tailored X-LoRA model that offers scientific capabilities including forward/inverse analysis tasks and enhanced reasoning capability, focused on biomaterial analysis, protein mechanics and design. The impact of this work include access to readily expandable and adaptable models with strong domain knowledge and the capability to integrate across areas of knowledge. Featuring experts in biology, mathematics, reasoning, bio-inspired materials, mechanics and materials, chemistry, protein biophysics, mechanics and quantum-mechanics based molecular properties, we conduct a series of physics-focused case studies. We examine knowledge recall, protein mechanics forward/inverse tasks, protein design, adversarial agentic modeling including ontological knowledge graph construction, as well as molecular design. The model is capable not only of making quantitative predictions of nanomechanical properties of proteins or quantum mechanical molecular properties, but also reasons over the results and correctly predicts likely mechanisms that explain distinct molecular behaviors.

***K*eywords** Language modeling · Scientific AI · Biomaterials · Proteins · Inverse problems · Fracture · Materials science

1 Introduction

Large language models (LLMs) [1, 2, 3, 4, 5, 6] have gained significant popularity, including in the development of special-purpose models that are experts in certain types of tasks, reasoning, or scientific domains [7, 8, 9, 10, 11, 12, 13, 14].However, training such models can be costly, especially when diverse sets of capabilities are needed. Methods like low-rank adapters (LoRA)[15] have been proposed as a more efficient alternative, but adaptations are usually focused on narrower fields of knowledge. The underlying concept in LoRA modeling is the use of low-rank matrices that are added to the original full-scale matrix, and selecting those low-rank matrices as the only trainable component of the model. Since only the adapter layers are trainable, these models typically preserve the knowledge captured during pre-training of the base model while making the model more applicable to specific tasks, in addition to being efficient in terms of computational needs. Since training of LoRA adapters is efficient, it is possible to develop a large set of special-purpose models using this strategy.

[*]Academy for Science and Design (ASD), 9 Townsend W, Nashua, NH 03063, USA
[†]Massachusetts Institute of Technology (MIT), 77 Mass. Ave 1-165, Cambridge, MA 02139, USA

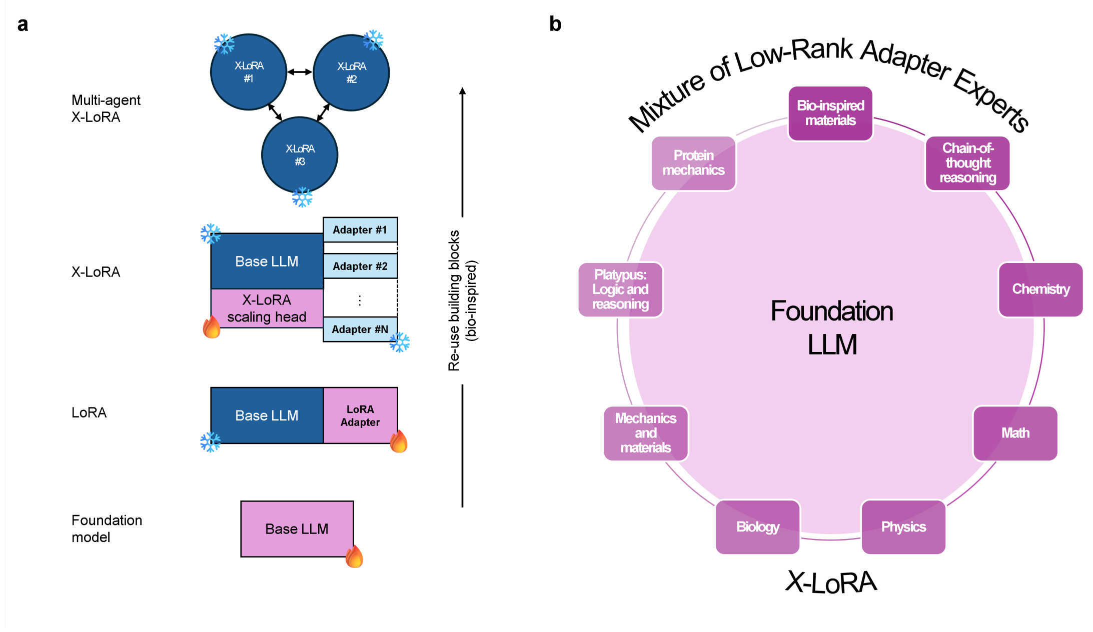


Figure 1: Multi-hierarchical design principle of adapted and agentic neural network architectures, following a bio-inspired paradigm that bases its foundation in re-use of existing building blocks. (a), in the schematic, blue color indicates a frozen (non-trainable) component, and reddish color indicates trainable components. The visual reflection of the re-use and adaptation through smaller trainable components is visible. Potentially, trainable components could be added also at the multi-agent level, albeit this is not yet implemented in this work. The training conducted as part of this work focuses on the LoRA adapter and X-LoRA levels of the model as a pre-trained foundation model is used as the basis. (b), Overview of the set of adapters used to construct the X-LoRA model, featuring experts in bioinspired materials, chain-of-thought (CoT) and reasoning, chemistry, mathematics, physics, mechanics and materials, logic and reasoning, and protein mechanics.

While LoRA adapters provide an efficient way to develop specialized model capabilities, it remains an open question how to integrate multiple of such systems into improved models that offer an integrated, and even improved set of capabilities. In fact, mixing LLMs has become an interesting area of research [16]. Other experimental works, such as using spherically interpolated model parameters (SLERP) and other concepts [17] have shown interesting results and a promise of mixing models to achieve new capabilities. Mixed models created in these ways are usually computed *a priori* according to a particular algorithm and while they have shown promising performance in some benchmarks, more research is necessary into rational development of mixed models and how certain features emerge. We hypothesize that dynamical mixing of parameters can be a more general approach as it allows for the mixed model to explore new combinations during inference.

Related approaches have used mixture of experts (MoE) strategies [18, 19, 20], including recently released Mixtral LLM [21]. However, these can be computationally expensive with significant demand even during inference. Here, we propose a computationally efficient mixture-of-expert approach using a set of distinct LoRA adapters that can easily be trained individually. Our approach implements a high-granularity and flexible method, inspired by a biological paradigm that re-uses neural network building block in the construction of a hierarchy of interacting models, from the foundation model to agentic systems (Fig. 1). To assert its impact quantitatively, our work specifically focuses on using this strategy for science-focused LLMs, here specifically applied to protein mechanics problems in the broader space of bio-inspired materials analysis and design.

We propose an algorithm to dynamically mix adapters at a token-level state, and with fine granularity to enable mixing of adapters at the levels of individual layers. The approach, referred to as X-LoRA, offers access to diverse sets of capabilities that are critical for scientific uses of multimodal language modeling [22, 23, 24**?**].

1.1 Fundamental concepts of X-LoRA

We first briefly review the concepts used in the development of LoRA adapters. The basic strategy behind low-rank adaptation (LoRA) [15], hypothesizes that updates to the weights have a low "intrinsic dimension" and takes advantage of this by freezing the original weights $W_0 \in \mathbb{R}^{d\times k}$ and constraining the updating to the low-rank decomposition

$$W_0 + \Delta W_0 = W_0 + BA \tag{1}$$

where $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times d}$ with the rank $r \ll \min(d, k)$. During training with LoRA [15], the pretrained weights W_0 are frozen and only the A and B matrices retain trainable parameters. The forward pass of a LoRA layer may be expressed as the following equation:

$$h = W_0x + \Delta W_0x = W_0x + BAx \tag{2}$$

We note that in practice, LoRA adapters have a fixed scalar weight α. This is applied to the decomposition matrices, yielding

$$h = W_0x + BAx \cdot \alpha \tag{3}$$

Building on this, X-LoRA is based on the idea to use a set of LoRA adapters, each of which has unique sets of capabilities, as has been shown in earlier science-focused LLM work (e.g., [10]). Unlike mixing or integrating models statically, we create a dynamic gating approach that scales individual LoRA adapter with individual token and layer granularity to facilitate mixing deep inside the model. In other words, considering a LoRA adapter i, the original α_i parameter is multiplied by a scaling value denoted by λ_i to create the new scalings value for the adapter α_i^*:

$$\alpha_i^* = \alpha_i \cdot \lambda_i \tag{4}$$

The scaling value λ_i is predicted by a X-LoRA scaling head that utilizes the model's hidden states, forming the trainable component in the X-LoRA model. Since the scaling head can take advantage of complex encodings it can be implemented efficiently as a simple feed-forward neural network with one or few layers (in the implementation of X-LoRA users can specify the details of how this neural network is constructed).

The scaling calculation is conducted for each token in the sequence. Hence, it offers a high level of granularity as each of the LoRA adapters that operate at specific layers in the LLM is scaled. Henceforth, we refer to this approach of gating the various components of the adapters in this way as scaling. Further details on the approach are provided in the Materials and Methods section.

1.2 Paper outline

The plan of this paper is as follows. First, we discuss the approach and training strategy including relevant datasets that results in the development of an X-LoRA model with special capabilities in the physical sciences, especially focused on biomaterials including proteins. We then present a series of experiments in which we apply an X-LoRA to a variety of tasks, such as question answering, conversational and agentic modeling, protein design and analysis, and others. We conduct a detailed analysis of the scaling patterns and validate the approach through comparison with molecular modeling and other physical data and methods.

2 Results and discussion

The process of developing the X-LoRA model consists of the following steps:

1. Train foundational base LLM (done in earlier work)
2. Individually train a set of adapters to develop expertise in a series of areas (these reflect distinct or overlapping areas of skills and knowledge)
3. Train the integrated X-LoRA model

Our experiments start with training a series of LoRA adapters. We develop a set of nine adapters, fined-tuned with distinct expertise, based on the Zephyr-7B-β model [25] that was built on top of the Mistral-7B model[5]:

1. Bioinspired materials
2. Chain-of-thought (CoT) and reasoning
3. Chemistry
4. Mathematics
5. Physics
6. Biology

7. Mechanics and materials
8. Platypus: Logic and reasoning
9. Protein mechanics tasks (generative sequence-to-property and inverse capabilities)

These areas of focus and expertise were chosen since we ultimately seek to have a STEM focused, materials science emphasized LLM. However, we postulate that our approach can be generally applied to all sorts of areas, especially to create modalities where connections between distinct areas of knowledge can be developed [8, 13].

After the LoRA adapters are created, we train the X-LoRA model. To do this, we use a subset of the combined set of the training samples used to train the individual LoRA adapters (details see Materials and Methods, including detailed delineations on the dataset, implementation and so on).

We use the X-LoRA system as an experimental entity to assess its performance in a few different settings. We conduct a series of experiments, including complex question answering, the use of the generative model for protein design, analysis and reasoning, as well as creating more sophisticated interaction strategies using adversarial agentic modeling. Throughout the analysis we compare performance of the base model with the X-LoRA model, and track how the scaling head utilizes different layers of the different experts across generation or development of answers.

2.1 Question answering and observed X-LoRA layer-wise scaling weights

We first examine differences between the X-LoRA model and the base foundation model. We ask both models this question:

```
What material behavior (stiffening vs softening) causes supersonic crack speeds in
mode II cracks?  Think step by step.
```

The response of the base model is:

```
Supersonic crack speeds, also known as Rayleigh waves, occur in mode II cracks due
to the stiffening material behavior of the material near the crack tip.  In mode II
fracture, the loading is shear-dominated and the crack propagates perpendicular to
the direction of applied stress.  Here's a step-by-step explanation:

As the crack tip approaches the material, the stresses in the vicinity increase
rapidly.  This results in a region of high stress concentrations called the stress
singularity.

In this stress singularity, the material experiences an elastic deformation that
leads to an increase in stiffness, i.e., stiffening behavior.  This stiffening
effect is due to the fact that the material near the crack tip becomes more
constrained as the crack approaches [...]
```

Using our own domain knowledge [26], this answer is not only incorrect but also long-winded. In contrast, the X-LoRA model responds as follows:

```
The material behavior that causes supersonic crack speeds in mode II cracks is
stiffening.
```

This answer is correct, clear and succinct [26].

The difference between a base model and the X-LoRA model is even more clear for the specialized protein analysis and design tasks. The task given is:

```
CalculateForceHistory< M S K S N T Y R M L V L L E D D T K I N K E D E K F L K G K P
G K M H E F V D E L I L P F N V D E L D E L N T W F D K F D A E I C I P Y V E I L K
E S G M K >
```

The response of the base foundation model is:

```
I'm sorry but the input provided is not a valid mathematical expression.  It appears
to be a sequence of characters that do not form a calculable function.  Please
provide a valid mathematical expression for me to calculate.

However, I can tell you that the sequence of characters you provided is actually
```

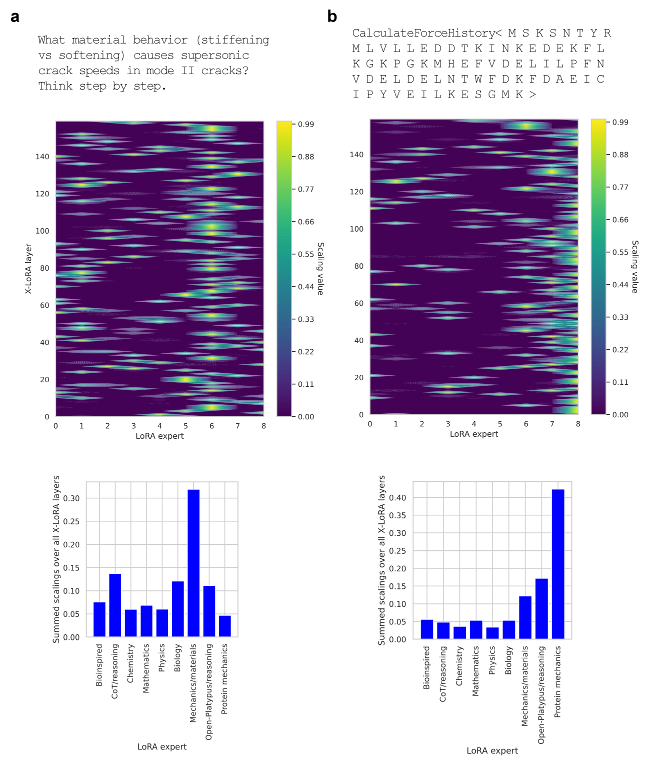


Figure 2: Results from question answering and observed X-LoRA scaling weights, for the trial questions used to compare X-LoRA with the base foundation model (panels a and b show results for two different tasks solved by the model). At the top of each sub-panel we show the question asked, followed by the X-LoRA scaling weights plotted over X-LoRA layers and LoRA experts, respectively. The lower bar plot summarizes the scalings over all layers, indicating an overall measure for which adapter is most prominently used. Panel a, featuring a question about dynamic fracture, heavily uses the Mechanics/Materials expert. In contrast, as shown in panel b, the protein analysis task results in the use of the protein mechanics adapter. It is observed that a complex pattern of scaling values is used in each case, suggesting that the X-LoRA model takes advantage of mixing different adapters heterogeneously across layers. Since the scaling weights determine which expert to use for a given input, the heatmaps show how this decision changes across different layers of the model, revealing a great level of heterogeneity.

```
a series of function and variable names that may be part of a larger program or
script in a specific programming language.  Without context, it's impossible to
determine the intended calculation.

If you provide more information, such as the programming language and the context
in which this expression appears, I may be able to provide more insight into its
meaning and possible calculations.
```

The X-LoRA model responds as follows, correctly carrying out the design task of generating the force-deformation curve associated with this protein:

```
[0.013,0.066,0.078,0.045,0.072,0.126,0.127,0.107,0.119, 0.074,0.106,0.098,0.103,0.128,
0.077, 0.069,0.131,0.108,0.144,0.174, 0.166,0.183,0.185,0.217,0.175,0.230,0.187,0.224,
0.267,0.301,0.315]
```

Fig. 2 shows results from this question answering and observed X-LoRA scaling weights. The two queries, one focused on mechanics and materials (Fig. 2a) and protein mechanics (Fig. 2b) show distinct patterns. Importantly, we find that a complex pattern of scaling values is used in each case, suggesting that the X-LoRA model takes advantage of mixing different adapters heterogeneously across layers. We further find that the heatmap in Fig. 2a has a broader range of activation across many experts and layers, with some areas of high activation. The heatmap in Fig. 2b shows a more concentrated pattern of activation, with high values being less widespread. We believe that this could indicate that the gating function on the right is more selective in its activation of experts.

In Fig. 2, the regions that resemble bright lines represent the paths of high activation, meaning the scaling function is activating certain experts more for specific layers. The patterns observed suggests there is not a uniform distribution of importance across all experts and layers; instead, it tends to be rather sparse and selective. This sparsity is characteristic of what has been observed in other mixtures-of-experts models, where only a subset of experts is chosen for any given input to specialize in different parts of the problem space. By comparing the maps for different tasks, both the sparsity and patterns could be important for further understanding which experts the model relies on for different types of inputs or at different stages of processing within the neural network (we will address the evolution of the maps as it changes dynamically during solving complex tasks later on).

To further analyze the behavior, we now explore a greater variety of questions to determine how the X-LoRA model uses the various experts, given a set of distinct prompts. We are specifically interested to see whether, and if yes, how, the X-LoRA model mixes different adapters in questions that overlap across different skills or domains. Fig. 3 shows results from question answering and observed X-LoRA scaling weights, with responses to the queries listed in Table 2.1. In Table 2.1 we show both, the response form X-LoRA and the response from the base model, Zephyr-7B-β. The differences between the two models are striking. To visually show this clearly, we mark incorrect or questionable response in red highlight. The comparison shows that X-LoRA provides far higher accuracy, as well as more concise answers.

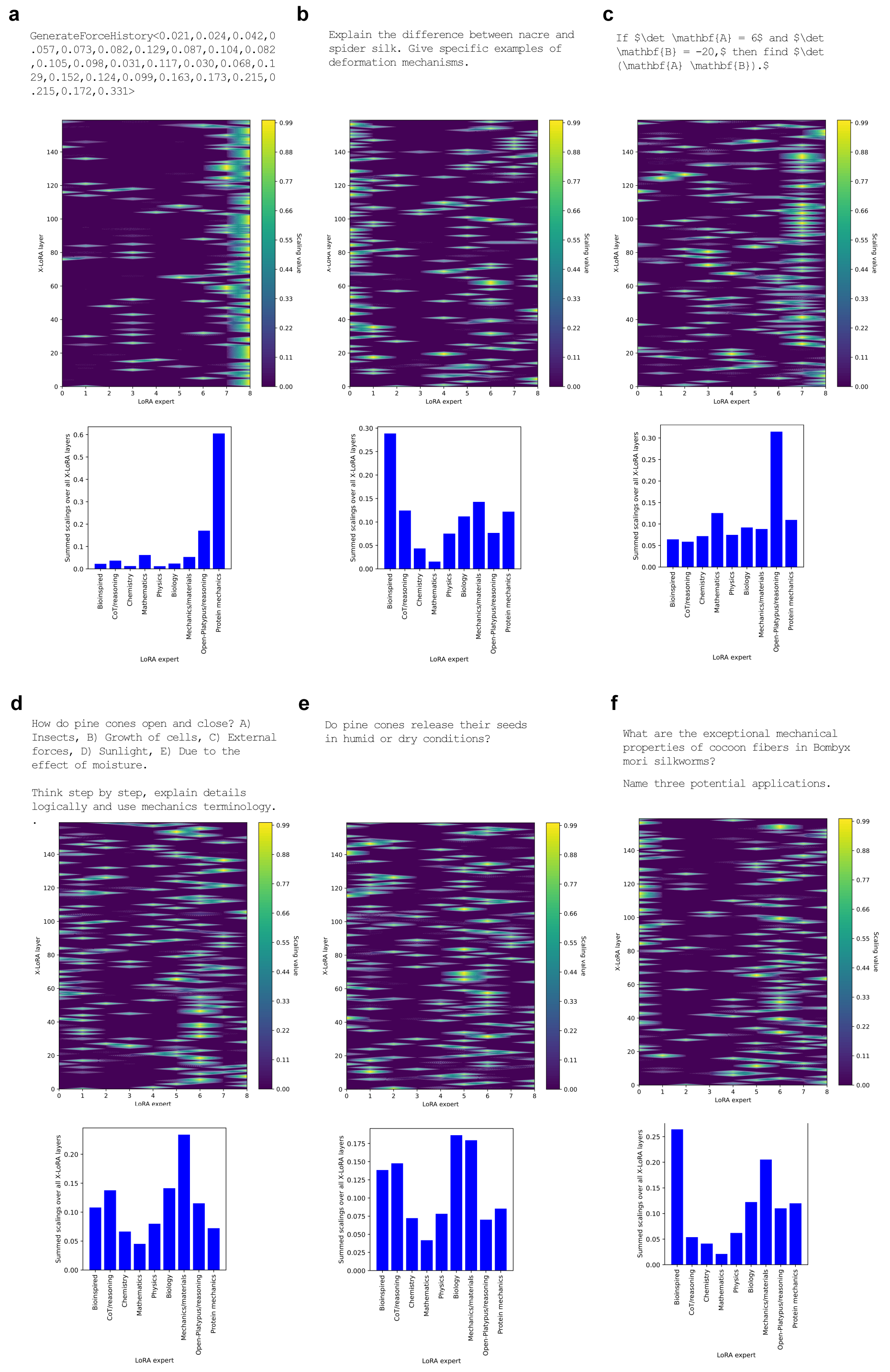
As hypothesized, we observe complex mixing of adapters and often the activation of several dominant LoRA experts. For instance, Fig. 3d reveals that a question about the mechanics of pine cones leads to strong usage of the mechanics/materials adapter combined with the CoT/reasoning and bioinspired adapters. While the correct letter is picked by X-LoRA, the answer is not entirely correct as pine cones open in dry, not wet conditions. A follow-up question to further probe into whether release happens in dry or wet conditions results in the correct answer, however. In a protein-focused question Fig. 3f reveals adapter mixing in response to a question related to the mechanics of Bombyx mori silk mechanics, activating a combination of the bioinspired, mechanics/materials and biology adapters. For clearly specialized tasks like protein design (Fig. 3a) we witness a largely singular activation of one of the experts.

A deeper analysis of the heatmap of scalings depicted in Fig. 3a shows that in addition to the obvious high relevance of the protein mechanics expert, there are several bands where certain experts are given high importance across several layers. This is particularly noticeable for the mechanics/materials and reasoning experts. These bands are not continuous and show a varying pattern, which might suggest specific conditions or features that these experts handle effectively. In fact, the reasoning expert (immediately to the left of the protein mechanics expert) stands out with several layers where they have the highest activation, especially in the upper layers (around 100 to 140). We believe that this could imply a significant role in the final decision-making or output of the model, albeit more research is necessary to solidify this. There is also a more localized but strong activation for the physics and biology experts at certain layers, which may indicate a specialization of this expert for features or representations in those regions.

Further, Fig. 4 depicts a token-level history over the summed up scalings over all layers. This analysis reveals histories for two examples, in Fig. 4a task related to pine cone seed release, and in Fig. 4b the result of a series of protein mechanics tasks. The protein mechanics task involved a series of prompt-answer interactions: First two property calculations, then a generative task to design a new protein towards the end. While the use of experts is generally stable, there are noticable changes in the use of the LoRA experts during the process. Specifically, the protein mechanics expert is highly focused in the last stages of generation when a new protein is being designed.

Table 2.1 shows diverse use cases across different domains and a comparison to the performance of the base model, on its own. We discuss a few of them in more depth.

For instance, we pose a question that involves a mix of protein science and biology with logic, asking about protein expression patterns. X-LoRA gets the answer right. The response from the base model contains some inaccuracies and logical inconsistencies when analyzing the expressions of marker proteins in each cell type. The base model correctly identifies that fibroblasts express Protein B and do not express Protein A due to the mutual exclusivity of Protein A and Protein B. However, it does not explicitly mention the expression of Protein C by fibroblasts. Given that there's no rule against expressing Protein C alongside Protein B, and since fibroblasts don't have a restriction against Protein C, they logically express Protein C as well. For Neurons, the base model starts correctly by stating that neurons express Protein A and not Protein B due to the mutual exclusivity rule. However, the discussion about the potential for neurons to express Protein C or both Protein C and Protein B is confusing and contains a mistake. Since neurons express Protein A and not Protein B (as stated), the only other option, given the rules, is for them to express Protein C. There is no scenario where neurons could express Protein B, as mentioned in the initial analysis. Therefore, neurons express Protein A and Protein C, not Protein B. For Epithelial cells, the base model correctly notes that epithelial cells



a

```
GenerateForceHistory<0.021,0.024,0.042,0
.057,0.073,0.082,0.129,0.087,0.104,0.082
,0.105,0.098,0.031,0.117,0.030,0.068,0.1
29,0.152,0.124,0.099,0.163,0.173,0.215,0
.215,0.172,0.331>
```

b

```
Explain the difference between nacre and
spider silk. Give specific examples of
deformation mechanisms.
```

c

```
If $\det \mathbf{A} = 6$ and $\det
\mathbf{B} = -20,$ then find $\det
(\mathbf{A} \mathbf{B}).$
```

d

```
How do pine cones open and close? A)
Insects, B) Growth of cells, C) External
forces, D) Sunlight, E) Due to the
effect of moisture.

Think step by step, explain details
logically and use mechanics terminology.
.
```

e

```
Do pine cones release their seeds
in humid or dry conditions?
```

f

```
What are the exceptional mechanical
properties of cocoon fibers in Bombyx
mori silkworms?

Name three potential applications.
```

Figure 3: Results from question answering and observed X-LoRA scaling weights (panels a-f show results from different tasks solved by the model, covering a range of domain areas and types). At the top of each sub-panel we show the question asked, followed by the X-LoRA scaling weights plotted over X-LoRA layers and LoRA experts, respectively. The lower bar plot summarizes the scalings over all layers, indicating an overall measure for which adapter is most prominently used. As before, it is observed that a complex pattern of scaling values is used in each case, suggesting that the X-LoRA model takes advantage of mixing different adapters heterogeneously across layers.

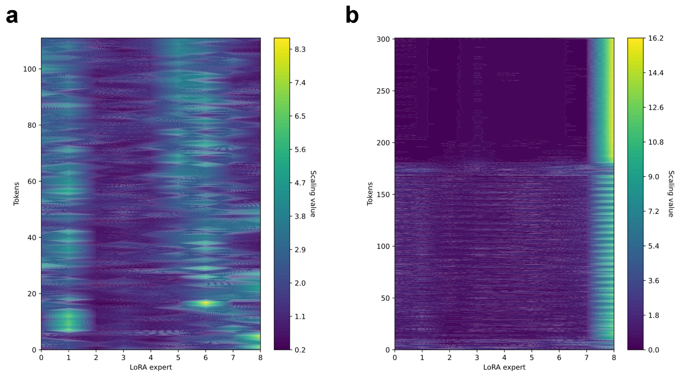


Figure 4: Summed up scalings over all layers, as a function of token history. This plot shows histories for two examples, in panel a task related to pine cones (around 110 tokens total), and in panel b the result of a series of protein mechanics tasks (around 300 tokens; first property calculations, then a generative task to design a new protein towards the end). While the use of experts is generally stable, there are noticable changes in the use of the LoRA experts during the process. The expert numbering (0 to 8) reflects the same organization as in the earlier plots, e.g. refer to Fig. 3 for the labels.

do not express Protein A but expresses both Protein B and Protein C. However, the explanation given is convoluted and includes incorrect statements about the mutual exclusivity of Protein A and Protein C, which was not part of the original observations. The simple reasoning is that epithelial cells do not express Protein A but must express the two remaining proteins, Protein B and Protein C, because the initial instructions only specify that Protein A and Protein B cannot be co-expressed. X-LoRA, in contrast, correctly identifies the marker proteins expressed by each cell type based on the provided observations and logical deductions. For fibroblasts, X-LoRA correctly deduces that fibroblasts cannot express Protein A because the presence of Protein A excludes the expression of Protein B. Since fibroblasts do express Protein B, and given that each cell type expresses a unique combination of three marker proteins (which was an incorrect assumption added), as the original statement didn't specify the number of proteins each cell type expresses, but rather that they express a unique combination), the conclusion that fibroblasts also express Protein C is correct based on the information provided (except the part about the number of proteins, which seems to be a misunderstanding). For neurons, the model correctly identified as expressing Protein A and Protein C. The reasoning is sound, noting that Protein B cannot be expressed alongside Protein A, and since neurons express Protein A and one other marker protein (not Protein B), they must express Protein C. Finally, for epithelial cells the X-LoRA model accurately concludes that epithelial cells express Protein B and Protein C. This is because epithelial cells do not express Protein A, and by process of elimination, given that they express two marker proteins, these must be Protein B and Protein C.

In the question about pine cone release in humid versus dry conditions, the base model gives a convoluted answer that is factually incorrect. X-LoRA gives a precise answer with clearl reasoning. There are other related examples, such as in the question about the exceptional mechanical properties of Bombyx mori silk, where the base model incorrectly asserts that silk fibers conduct electricity. The model also fails in solving several other domain specific tasks, such as to the question about the role of hyperelasticity in maximum fracture speeds where X-LoRA provides a concise and accurate answer, whereas the base model fails to answer. In a question about strong adhesion of gecko feet, X-LoRA correctly identifies the importance of setae and nano-sized effects, whereas the base models makes incorrect claims about the role of special purpose glue proteins in the process. These and other examples discussed in the table and elsewhere in the paper provide strong evidence for the enhanced capabilities of X-LoRA over the base model.

Table 1: Summary of question-answer pairs, in the domains of biological materials, mechanics of materials, protein properties, logic and reasoning, and related areas, focusing on challenging questions. Several sample questions are posed and answered by X-LoRA and the Zephyr-7B-β base model for comparison. Associated X-LoRA scaling values for some of the queries are presented in Figure 3 to show how the model is able to mix different adapters and associated capabilities depending on the context. We mark incorrect or questionable response in red highlight. The comparison shows that X-LoRA provides far higher overall accuracy, as well as more concise answers.

Question	**X-LoRA model response**	**Zephyr-7B-β base model response**

Continued on next page

Table 1 – continued from previous page

Question	**X-LoRA Answer**	**Zephyr-7B-β base model Answer**
GenerateForceHistory <0.021,0.024,0.042,0.057,0.073,0.082,0.129,0.087,0.104,0.082,0.105,0.098,0.031,0.117,0.030,0.068,0.129,0.152,0.124,0.099,0.163,0.173,0.215,0.215,0.172,0.331>	[MSKSNTYRMLVLLEDDTKINKEDEKFLKGKPGKMHEFVDELILPFNVDELDELNTWFDKFDAEICIPYVEILKESGMK]	The provided sequence of numbers represents the history of force measurements over time. Here's a possible interpretation: • The first seven values (0.021, 0.024, ..., 0.082) represent the force readings at different points in time. These values appear to be relatively small and fluctuating around a low average. • The next three values (0.129, 0.087, 0.104) show an increase in force, followed by a decrease and then another increase. This could indicate some kind of cyclical behavior or trend. • After this, there are a few more fluctuations with some larger values (0.105, 0.098, ..., 0.173). It's unclear whether these represent significant changes or just noise in the measurement process. • Finally, the last four values (0.215, 0.215, 0.172, 0.331) show a dramatic increase in force, possibly indicating a significant event or anomaly. However, it's not clear from this data alone what caused this spike. More context and additional measurements would be needed to fully understand what happened here.
Explain the difference between nacre and spider silk. Give specific examples of deformation mechanisms.	Nacre, also known as mother-of-pearl, is a biomineralized material found in the shells of mollusks such as oysters and pearl mussels. It consists of layers of aragonite platelets held together by an organic matrix of proteins and polysaccharides. Nacre's strength and toughness come from its hierarchical structure, which includes nanoscale aragonite crystals arranged in a brick-and-mortar arrangement with organic interlayers. This structure allows for efficient load transfer and energy dissipation during deformation. Spider silk, on the other hand, is a protein fiber produced by spiders for various purposes, such as capturing prey or building webs. Spider silk is composed of fibroin proteins that form long, thin fibers with high tensile strength and elasticity. The mechanical properties of spider silk are due to its semicrystalline structure, which includes beta-sheet crystals and amorphous regions. Spider silk exhibits excellent toughness and can undergo significant deformation before breaking. In terms of deformation mechanisms, both nacre and spider silk exhibit plastic deformation, meaning they can be permanently deformed without breaking. However, the mechanisms behind this deformation differ. In nacre, plastic deformation occurs through interfacial debonding between the aragonite platelets and the organic matrix, as well as crack deflection at the interface. These mechanisms allow for energy dissipation and stress redistribution, resulting in improved toughness. In contrast, spider silk deforms plastically through the sliding of beta-sheet crystals past each other, which can occur along the direction of the fiber or transverse to it. This sliding mechanism allows for large strains without fracture, contributing to the exceptional toughness of spider silk. Additionally, the presence of amorphous regions in spider silk facilitates molecular slippage during deformation, further enhancing its plasticity. Overall, while both nacre and spider silk exhibit plastic deformation, the underlying mechanisms are distinct, reflecting their different structures and compositions.	Nacre, also known as mother-of-pearl, is a biomaterial found in the shells of mollusks such as oysters, clams, and mussels. Spider silk, on the other hand, is a protein fiber produced by spiders for their webs and other purposes. Although both materials have remarkable mechanical properties, they differ significantly in terms of structure and deformation mechanisms. Structure: Nacre consists of thin platelets (called tabular crystals) made of calcium carbonate (CaCO3), arranged in layers with organic material (primarily proteins) sandwiched in between. This layered structure gives nacre its characteristic iridescence. In contrast, spider silk is composed of fibroin, a protein that forms fibers with a unique hierarchical structure. The fibers are made up of nanoscale fibrils, which are further organized into larger fibers. Deformation Mechanisms: The deformation behavior of nacre and spider silk is quite different due to their distinct structures. When subjected to tension, nacre exhibits a distinctive brittle fracture behavior. As the load increases, the tabular crystals separate from each other, causing cracks to propagate through the material. However, the organic matrix between the plates prevents complete failure of the material, resulting in a relatively high toughness. An example of this behavior can be seen in the brittleness of pearl buttons when exposed to water or humidity, where the tabular crystals delaminate and detach from the button surface. Spider silk, on the other hand, displays highly elastic behavior under tension. The fibroin protein molecules in spider silk form extended beta-sheets, which provide the material with exceptional strength and stiffness. As the material is stretched, these sheets slide past one another, allowing the fiber to extend without breaking. This sliding mechanism results in a large strain-to-failure, making spider silk several times stronger than steel of similar weight. An example of this behavior can be seen in the ability of spider webs to stretch and recoil upon impact, enabling spiders to capture prey. In summary, while both nacre and spider silk exhibit impressive mechanical properties, their differences in structure and deformation mechanisms result in unique behaviors under tension. Nacre's layered structure allows it to absorb energy during deformation, while spider silk's hierarchical structure enables it to stretch and recoil without breaking. These properties make both materials promising candidates for developing novel bioinspired materials with tailored mechanical properties for various applications.
If $\det \mathbf{A}=6$ and $\det \mathbf{B}=-20$, then find $\det(\mathbf{AB})$.	The determinant of the product of two matrices is equal to the product of their determinants, so: $\det(\mathbf{AB}) = \det \mathbf{A} \det \mathbf{B}=6(-20)=-120$.	Using the property that the determinant of a product is equal to the product of the determinants (provided that the matrices commute), we have: $\det(\mathbf{AB}) = \det(\mathbf{A}) \det(\mathbf{B}) = 6(-20) = -120$.

Continued on next page

Table 1 – continued from previous page

Question	X-LoRA Answer	Zephyr-7B-β base model Answer
How do pine cones open and close? A) Insects B) Growth of cells C) External forces D) Sunlight E) Due to the effect of moisture Think step by step, explain details logically and use mechanics terminology.	Pine cones open and close due to the effect of moisture. The scales on the cone are held together by tiny hinges that allow the cone to open when moisture enters the area. This is a type of biomechanical movement that occurs in response to environmental conditions.	Pine cones do not open and close in response to sunlight or insects. Instead, they open and close due to the effect of moisture and external forces. The process is called cone scaling. Here's how it works: 1. Closed cones: A closed pine cone has scales that overlap tightly, forming a compact structure around the seeds inside. This helps to prevent water loss and protect the seeds from predators. 2. Moisture triggers opening: When enough moisture is present, such as after rainfall or during humid weather, the scales on the cone begin to separate slightly. This allows air to circulate and promotes the growth of fungi, which helps to break down the seed coats and prepare them for germination. 3. External forces aid in opening: As the scales continue to separate, gravity and wind also play a role in fully opening the cone. This exposes the seeds to the elements and increases their chances of dispersal through wind or animal transport. 4. Closing: Once the seeds have been dispersed or have fallen to the ground, the cone begins to dry out and the scales come together again, forming a tight seal around any remaining seeds. This helps to prevent further water loss and protect the seeds until the next season when conditions are favorable for germination. In summary, the opening and closing of pine cones is a complex process driven by environmental factors and mechanical forces. It allows for the efficient dispersal of seeds while protecting them from harm until the right conditions arise for successful germination.
Do pine cones release their seeds in humid or dry conditions?	Pine cones release their seeds in dry conditions. When the air around a pine cone becomes dry enough, the scales of the cone will separate, releasing the seeds. This process is known as coning. In humid conditions, the scales of the cone remain closed, preventing seed release.	Pine cones generally release their seeds (also known as pine nuts) in response to moisture and warmth. In many species of pine trees, the cones will only open and release their seeds when they are exposed to warm, wet conditions. This is because the seeds inside the cone need sufficient moisture to germinate and grow into new pine trees. However, some pine species have adapted to release their seeds in drier conditions, particularly in areas with Mediterranean climates where winter rainfall is scarce. These cones may remain closed until they receive a significant amount of rainfall, which can trigger them to open and release their seeds. Overall, the timing and conditions for pine cone seed release can vary widely depending on the specific species of pine tree.
What are the exceptional mechanical properties of cocoon fibers in Bombyx mori silkworms? Name three potential applications.	The exceptional mechanical properties of cocoon fibers in Bombyx mori silkworms include high tensile strength, elasticity, and toughness. These properties are due to the unique structure of the fibers, which consist of crystalline and amorphous regions. The crystalline regions provide strength, while the amorphous regions contribute to elasticity. Three potential applications of these exceptional properties include: Development of lightweight and strong composite materials for aerospace and automotive industries. The high strength-to-weight ratio of cocoon fibers could make them ideal candidates for use in composites. Creation of biodegradable and environmentally friendly materials for packaging and textiles. The natural origin of cocoon fibers makes them an attractive alternative to synthetic materials, which can be harmful to the environment. Design of new drug delivery systems that take advantage of the unique properties of cocoon fibers. The controlled release of drugs from these fibers could lead to more effective and targeted therapies.	The exceptional mechanical properties of cocoon fibers in Bombyx mori silkworms are due to their unique composition and structure. Silk fibers are composed of two main proteins, fibroin, and sericin. Fibroin is the primary component that gives silk its strength and toughness, while sericin acts as a binding agent. 1. Strength: Silk fibers have a tensile strength of up to 1.3 GPa, which is higher than that of many synthetic fibers, such as nylon and polyester. This strength is due to the crystalline structure of fibroin, which forms a regular arrangement of protein molecules. 2. Toughness: Silk fibers also have a high toughness, which is a measure of a material's ability to absorb energy before breaking. Silk fibers can absorb up to 35% of their original length before breaking, which is higher than that of many synthetic fibers. 3. Elasticity: Silk fibers are also highly elastic, meaning they can be stretched and then return to their original shape without breaking. This elasticity is due to the presence of amorphous regions in the fibroin protein, which allow for some flexibility in the fiber. Three potential applications of bioinspired materials based on silk fibers are: 1. Biomedical applications: The biocompatibility and biodegradability of silk fibers make them an attractive material for use in biomedical applications, such as wound healing dressings, sutures, and drug delivery systems. 2. Textiles: The high strength and toughness of silk fibers make them a potential alternative to synthetic fibers in textiles, particularly for high-performance applications such as sportswear and protective gear. 3. Electronics: The unique electrical properties of silk fibers, such as their ability to conduct electricity, make them a potential material for use in electronic devices, such as sensors and flexible displays.

Continued on next page

Table 1 – continued from previous page

Question	X-LoRA Answer	Zephyr-7B-β base model Answer
What is the role of hyperelasticity on dynamic fracture of brittle materials, specifically in terms of the maximum crack speed.	The role of hyperelasticity on dynamic fracture of brittle materials is significant, as it affects the maximum crack speed. In linear elastic materials, the maximum crack speed is determined by the speed of sound in the material. However, when hyperelasticity is present, the maximum crack speed can be significantly higher than the speed of sound, approaching supersonic speeds. This is due to the fact that hyperelastic materials exhibit a stiffening effect at large strains, which can drive the crack to higher speeds. The dynamic crack speed is determined by a balance between the crack driving force and the material's stiffness, and hyperelasticity can have a strong influence on both of these factors.	Hyperelasticity, which is the nonlinear elastic behavior of materials, can significantly impact the dynamic fracture of brittle materials, particularly the maximum crack speed. In dynamic fracture, a crack propagates at high velocities due to the release of elastic energy. The maximum crack speed is limited by the material's ability to dissipate energy through plastic deformation or fracture. In brittle materials, the maximum crack speed is typically lower than that of ductile materials due to the absence of plastic deformation. However, hyperelasticity can increase the maximum crack speed by allowing for a larger deformation before fracture. This is because hyperelastic materials can store and release more energy per unit volume than linear elastic materials, which can delay the onset of fracture. The effect of hyperelasticity on dynamic fracture is characterized by the crack velocity-crack opening displacement (COD) curve. The COD is the displacement between the crack surfaces as the crack propagates. The curve shows the relationship between the crack velocity and the COD required for fracture. In hyperelastic materials, the COD required for fracture is higher than that of linear elastic materials, which results in a higher maximum crack speed. In summary, hyperelasticity can increase the maximum crack speed in brittle materials by allowing for a larger deformation before fracture, which delays the onset of fracture and increases the energy required for fracture. This is important for understanding the behavior of brittle materials under dynamic loading conditions.
Why are the properties of pristine cellulose nanocrystal (CNC) films not comparable to tough biological composites like bone or mantis shrimp club? A) Lack of plasticizers B) Excessive brittleness C) High tensile modulus Think step by step.	The properties of pristine cellulose nanocrystal (CNC) films are not comparable to tough biological composites like bone or mantis shrimp club because of their excessive brittleness. CNC films have a high tensile modulus, but their lack of plasticity makes them prone to fracture under stress. To improve the mechanical properties of CNC films, plasticizers are often added to reduce their brittleness and increase their toughness.	While pristine cellulose nanocrystal (CNC) films exhibit high tensile modulus, their properties are not comparable to tough biological composites like bone or mantis shrimp club due to two main reasons: Lack of plasticizers: Biological composites like bone and mantis shrimp club contain organic molecules called plasticizers, which help in reducing the intermolecular interactions between the nanoscale building blocks, thereby increasing the mobility of the molecules and allowing for plastic deformation. In contrast, CNC films lack such plasticizers, which leads to excessive brittleness and a lack of plastic deformation. Excessive brittleness: The high tensile modulus of CNC films is due to the strong intermolecular interactions between the nanocrystals. However, this also results in excessive brittleness, as the material lacks the ability to deform plastically under load. Biological composites like bone and mantis shrimp club, on the other hand, have a combination of strong and weak bonds, which allows for a balance between stiffness and ductility. In summary, the lack of plasticizers and excessive brittleness in pristine CNC films are the main reasons why their properties are not comparable to tough biological composites like bone or mantis shrimp club.
A scientist is studying three different cell types: fibroblasts, neurons, and epithelial cells. Each cell type expresses a unique combination of three marker proteins: Protein A, Protein B, and Protein C. Based on the following observations, determine which marker proteins are expressed by each cell type. 1. When Protein A is present, Protein B cannot be expressed. 2. Fibroblasts express Protein B. 3. Neurons express Protein A and one other marker protein, but not Protein B. 4. Epithelial cells do not express Protein A but do express the two other marker proteins. Identify the marker proteins expressed by fibroblasts, neurons, and epithelial cells, and provide a logical explanation for your determination.	Fibroblasts cannot express Protein A because when Protein A is present, Protein B cannot be expressed. However, Fibroblasts express Protein B, so they must express Protein C as well, since each cell type expresses a unique combination of three marker proteins. Neurons express Protein A and one other marker protein, but not Protein B. Since Protein B cannot be expressed when Protein A is present, Neurons must express Protein C. Epithelial cells do not express Protein A but do express the two other marker proteins. Since each cell type expresses a unique combination of three marker proteins, Epithelial cells must express Protein B and Protein C. In summary, Fibroblasts express Protein B and Protein C, Neurons express Protein A and Protein C, and Epithelial cells express Protein B and Protein C.	Fibroblasts express Protein B, as stated in observation 2. Since Protein A and Protein B cannot be expressed simultaneously (observation 1), fibroblasts do not express Protein A. Neurons express Protein A, as mentioned in observation 3. Since neurons do not express Protein B (as Protein A and Protein B cannot be expressed together), they must express either Protein C or both Protein C and Protein B. However, observation 4 states that epithelial cells express both Protein C and Protein B, but do not express Protein A. Since neurons do not express Protein A, they cannot also express Protein B, as this would conflict with the observation about epithelial cells. Therefore, neurons express Protein C, but not Protein B. Epithelial cells do not express Protein A, according to observation 4. Since epithelial cells express both Protein C and Protein B, this implies that Protein A and Protein C cannot be expressed simultaneously (otherwise, epithelial cells would express both Protein A and Protein C, which they do not). Therefore, Protein A and Protein C are mutually exclusive markers for these cell types. This also means that epithelial cells cannot express Protein A and Protein B together, as this would conflict with the observation that fibroblasts cannot express Protein B and Protein A simultaneously. Therefore, epithelial cells express Protein B, but not Protein A. In summary, fibroblasts express Protein B, neurons express Protein C, and epithelial cells express Protein B but not Protein A. The logical explanation for these findings is based on the mutual exclusivity of certain marker proteins, as observed in the given conditions.

Continued on next page

Table 1 – continued from previous page

Question	**X-LoRA Answer**	**Zephyr-7B-β base model Answer**
Discuss the structural and functional diversity of the protein collagen, highlighting its role in the extracellular matrix (ECM) of connective tissues. Include in your discussion the biosynthesis of collagen, its unique triple-helical structure, and how post-translational modifications contribute to its mechanical properties. Give an example of a collagen-related disease with a strong mechanics component.	Collagen is a family of proteins that play a crucial role in the extracellular matrix (ECM) of connective tissues. It is the most abundant protein in the human body and provides structural support and mechanical strength to various tissues. Collagen is composed of three polypeptide chains, which are arranged in a unique triple-helical structure. This structure is stabilized by hydrogen bonds and provides collagen with its exceptional mechanical properties. The biosynthesis of collagen begins with the transcription of collagen genes in fibroblasts. The resulting mRNA is then translated into procollagen, which is post-translationally modified in the endoplasmic reticulum and Golgi apparatus. Procollagen undergoes hydroxylation, glycosylation, and other modifications that contribute to its mechanical properties. The mature collagen molecules are then secreted by the fibroblasts and assembled into fibrils, which form the ECM of connective tissues. Collagen fibrils are highly organized structures that provide mechanical strength and stability to tissues. They are composed of multiple collagen molecules, which are arranged in a parallel fashion. The triple-helical structure of collagen allows it to withstand large tensile loads, making it an ideal material for load-bearing tissues. One example of a collagen-related disease with a strong mechanics component is osteogenesis imperfecta (OI), also known as brittle bone disease. OI is a genetic disorder that affects the synthesis and processing of collagen, leading to weak and fragile bones. Patients with OI often experience frequent bone fractures, joint dislocations, and skeletal deformities. The mechanical properties of collagen in OI patients are altered, resulting in reduced bone strength and increased susceptibility to fractures. In summary, collagen is a highly diverse and functional protein that plays a crucial role in the ECM of connective tissues. Its unique triple-helical structure and post-translational modifications contribute to its exceptional mechanical properties. Collagen-related diseases, such as OI, highlight the importance of collagen in maintaining tissue integrity and mechanical strength.	Collagen is a family of structurally and functionally diverse proteins that play a crucial role in the extracellular matrix (ECM) of connective tissues. Collagen is the most abundant protein in mammals, accounting for approximately 25% of the total protein content in the body. Collagen is synthesized by cells such as fibroblasts, chondrocytes, and osteoblasts, and its biosynthesis involves a complex process that includes transcription, translation, post-translational modifications, and secretion. Collagen is synthesized as a precursor protein called procollagen, which undergoes post-translational modifications to form the mature collagen fibrils. The procollagen molecule is composed of three polypeptide chains, each approximately 1000 amino acids long, that are held together by interchain disulfide bonds. The triple-helical structure of collagen is unique and is formed by the alignment of three polypeptide chains in a left-handed supercoil. This structure provides collagen with its characteristic mechanical properties, including strength, stiffness, and elasticity. Collagen plays a critical role in the ECM of connective tissues by providing structural support, facilitating cell adhesion and migration, and regulating cellular processes such as proliferation, differentiation, and apoptosis. Collagen is also involved in various physiological processes such as wound healing, tissue repair, and remodeling. One example of a collagen-related disease with a strong mechanics component is osteogenesis imperfecta (OI), also known as brittle bone disease. OI is a genetic disorder that affects collagen synthesis and results in bone fragility and deformities. The mutations in the collagen genes that cause OI can lead to abnormal collagen fibril assembly, altered collagen cross-linking, or decreased collagen synthesis, resulting in reduced bone strength and increased susceptibility to fractures. The mechanical properties of collagen in OI patients are altered, leading to reduced bone stiffness and increased bone fragility. This highlights the importance of collagen structure and function in maintaining the mechanical integrity of connective tissues and the potential role of collagen-based therapies in treating collagen-related diseases.
Select the most likely answer (A, B or C): How do higher humidity conditions affect the stiffness of bird feathers? A) higher humidity, lowers stiffness B) higher humidity, increases stiffness C) higher humidity causes no change	Bird feathers have a hierarchical structure that includes alpha-keratin fibers, barbs, and barbules. The stiffness of these structures is influenced by factors such as humidity, temperature, and relative humidity. Higher humidity conditions can cause the feather's keratin fibers to absorb moisture, leading to increased swelling and potentially decreased stiffness (softening). Therefore, the most likely answer is A.	1. Feathers have a hierarchical structure that provides them with their unique mechanical properties, such as stiffness. The stiffness of feathers is determined by several factors, including the arrangement and orientation of the microfibrils within the feather barbs. 2. Humidity can affect the behavior of materials by changing their moisture content. High humidity levels can cause materials to absorb more water, which can lead to changes in their physical properties. 3. In the case of bird feathers, it has been reported that high humidity levels can actually increase their stiffness. This is because the increased moisture content in the air causes the feathers to swell slightly, which compresses the internal structures and increases the stiffness. However, this effect is only temporary and reversible, as the feathers will return to their original stiffness when the humidity level decreases. Therefore, the most likely answer is option B: higher humidity, increases stiffness. Note: It's always important to double-check your sources and make sure they are reliable before relying on any specific information. In this case, I am basing my answer on published research studies that have investigated the effects of humidity on feather stiffness.
Which of these concepts is key to explain the strong adhesion of gecko feet. (A), the use of super-glue by the animal. (B), a size effect, where nano-sized fibrils yield strong and flaw-tolerant adhesion. (C), the low weight of the gecko due to its efficient design. (D), the gecko has special-purpose proteins derived from sericin that ensure it sticks to many surfaces.	(B) is the key to explain the strong adhesion of gecko feet. The gecko's feet are covered in tiny, hair-like structures called setae, which split into even smaller fibers called spatulae. These structures allow the gecko to adhere to surfaces using van der Waals forces, and the strong adhesion is due to the size effect, where nano-sized fibrils yield strong and flaw-tolerant adhesion.	(D), the gecko has special-purpose proteins derived from sericin that ensure it sticks to many surfaces, is the key concept to explain the strong adhesion of gecko feet. The gecko's feet are covered in millions of tiny hairs, called setae, which are coated in a protein called gecko sericin. This protein creates van der Waals forces between the gecko's feet and the surface it is touching, allowing for strong adhesion without the use of super-glue or any other adhesive substance. The size effect and efficient design of the gecko may contribute to its overall adhesion, but it is the unique properties of the sericin protein that enable the gecko's remarkable ability to climb vertical surfaces and walk upside down on ceilings.

Figure 5 shows a comparison of knowledge recall evaluation experiments, using the bio-inspired knowledge test set introduced in [11]. We can see that X-LoRA shows the best performance, even though it is a much smaller model than the BioinspiredLLM model or the Llama-BioLLM model (7B for X-LoRA vs. 13B parameters in the others). Several other comparisons with other recently published LLMs are included, specifically Orca-13B and Llama-13b-chat. The plot also includes a comparison of X-LoRA with the base model, Zephyr-7B-β model, showing that the X-LoRA model has significantly improved performance. These comparisons indicate that strong performance can be achieved using X-LoRA, even if smaller base models are used.

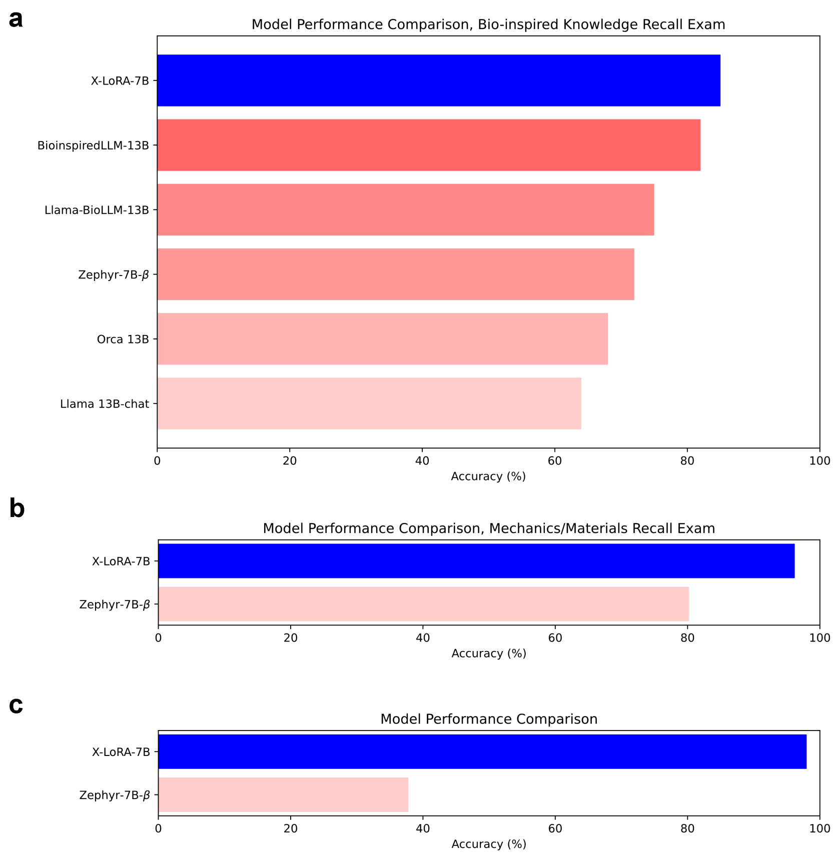


Figure 5: Results from knowledge recall evaluation experiments. (a), Results using the bio-inspired knowledge recall exam introduced in [11]. X-LoRA shows the best performance, even though it is a much smaller model than the BioinspiredLLM model (7B vs. 13B parameters). The plot also includes a comparison of X-LoRA with the base model, Zephyr-7B-β model. It is notable that X-LoRA, in spite of being a much smaller model and in spite of having a host of different capabilities, provides superior performance compared to all other models. (Data for the other model performances extracted from [11].) (b), Results from mechanics/materials knowledge recall benchmark as reported in [8]. (c), Results of the questions posed in Table 2.1.

Figure 6 shows an application in image synthesis, where we use X-LoRA to generate a prompt for use in other models, for instance DALL-E 3 as done here.

```
Create a prompt that I can use for image generation that accurately describes the
microstructure of a hierarchical composite.

Explicitly describe the geometry.
```

The comparison (Figure 6(b)-(c)) clearly shows a more concise prompt, which in turn results in a much more accurate image produced by DALL-E 3 [27]. Notably, the prompt generated by Zephyr-7B-β does not accurately reflect the instruction provided and introduces a variety of other design components such as fibers and nanoparticles. These were not mentioned in the original prompt that focused specifically on the microstructure of a hierarchical composite. The X-LoRA prompt is shorter and more focused, and hence produces a much better result.

2.2 Generative solutions to protein analysis

We now move on to specific protein generative and analytical tasks. These capabilities emerge in the X-LoRA model based on the protein mechanics adapter, and are enhanced by mixing with the model's knowledge in biology, bio-inspired materials and reasoning as well as logical deduction. The discussion in this section is dedicated to the analysis of protein tasks on their own, assessing quantitative performance of this capability.

a

Prompt: Create a prompt that I can use for image generation that accurately describes the microstructure of a hierarchical composite.

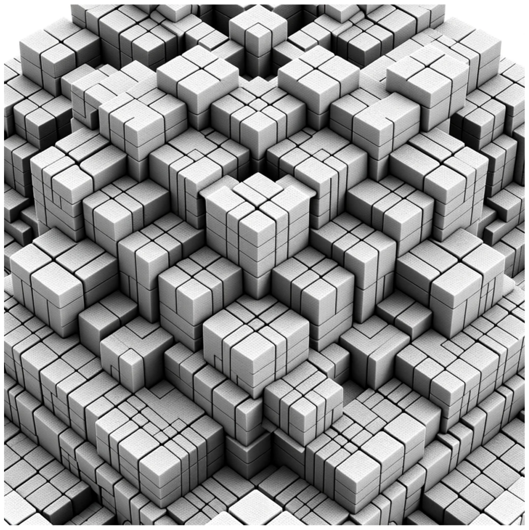
Explicitly describe the geometry.

b

X-LoRA: Generate an image of a hierarchical composite with a brick-and-mortar structure, where the bricks are made of a strong material and the mortar is made of a weaker material. The bricks have a square cross-section and the mortar fills the spaces between them. The composite should have multiple levels of hierarchy, with larger bricks made of smaller bricks and smaller bricks made of even smaller bricks, down to the atomic level. The image should be in grayscale and have a resolution of 1024x1024 pixels..

c

Zephyr-7B-β: Generate an image of a hierarchical composite material with a multi-scale architecture consisting of interconnected fibers and nanoparticles embedded in a matrix. The fibers are arranged in a regular pattern, forming a macroscale structure with a diameter of 100 micrometers. At the microscale, these fibers are made up of smaller sub-fibers with a diameter of 10 micrometers, which in turn consist of nanoscale fibrils with a diameter of 1 nanometer. The nanoparticles are dispersed throughout the matrix and are preferentially located at the interfaces between the fibers and the matrix. The overall composite material exhibits a unique combination of mechanical properties, including high strength, stiffness, and toughness, due to the synergistic effect of the hierarchical structure.



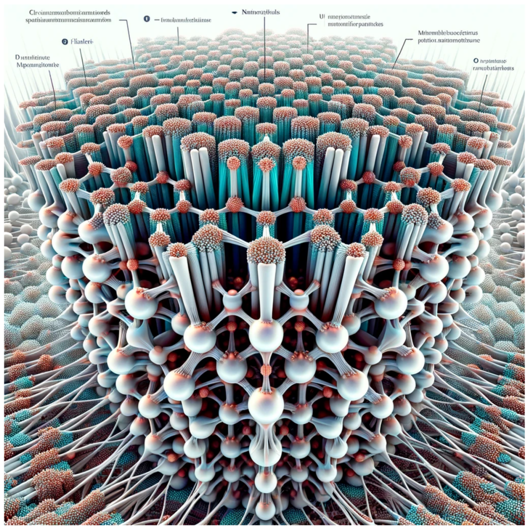


Figure 6: Applications towards image synthesis. Here we task the model to generate an image generation prompt (a). Results for X-LoRA (b) and Zephyr-7B-β (c) are shown. The comparison clearly shows a more concise prompt, which in turn results in a much more accurate image produced by DALL-E 3. Comparing the two prompts, the X-LoRA prompt is shorter and more focused, and produces a much better result.

Fig. 7 shows results based on the protein task to predict force-deformation behaviors from sequences. Fig. 15 shows the overall performance for predicting unfolding force and unfolding energy from the sequence, comparing ground truth with predictions (we find $R_2 = 0.85$). Overall, performance of the model is exquisite and as will be explored in the next section, works very well also for the inverse task of design and cycle consistent assessment[28]. However, unlike in earlier work that used highly-specialized GPT-type models to solve such tasks, our X-LoRA model combines these special-purpose capabilities with a host of other deep specialization as endowed via the various fine-tuned adaptations.

2.3 Protein design and analysis

We now expand the test cases towards more integrative applications that touch upon multiple areas of capabilities. Specifically, the experiments reported in this section show how the X-LoRA model is able to draw upon a distinct set of skills and how agentic modeling can bring about even more challenging and exhaustive responses. We begin the discussion with a protein design task, where we ask the model to design a novel protein to meet a certain target force-deformation response. These force-deformation responses have been studied in earlier work [29] and reflect measurements of force versus deformation as the protein is pulled at both ends (reflecting a classic protein unfolding experiment [30]).

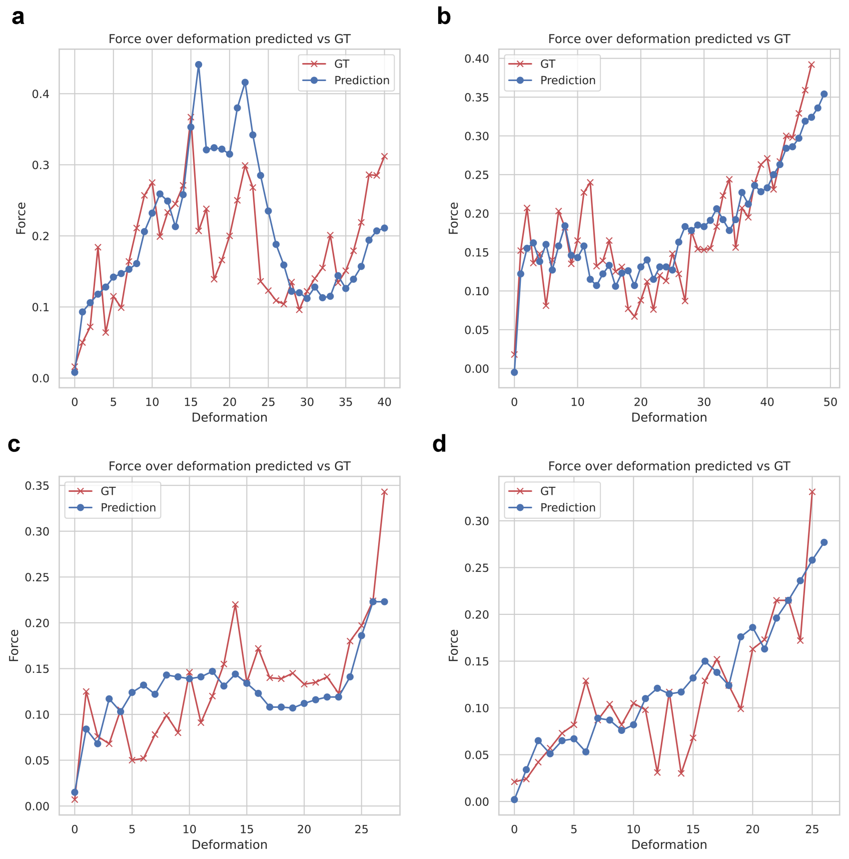


Figure 7: Using the protein task in X-LoRA to predict force-deformation behaviors from amino acid sequences, for four examples from the test set (shown as panels a-d). As can be seen, the model has excellent forward capabilities and can predict the nonlinear mechanical behavior well.

Fig. 9 shows how we can use the generative protein task. We design a protein with a desired force-deformation behavior (training data and boundary conditions used in the original molecular dynamics (MD) simulations of protein pulling, see [29]) and then test the predicted sequence, examining whether or not it meets the desired target properties (Fig. 9a). This cycle-consistent analysis is important to test whether the inverse capabilities matches the forward prediction. Fig. 9b shows the result for the sequence predicted, `MSKSNTYRMLVLLEDDTKINKEDEKFLKGKPGKMHE-FVDELILPFNVDELDELNTWFDKFDAEICIPYVEILKESGMK`. Further, Fig. 9c shows a folded model of the protein (obtained using AlphaFold 2 [31]). To assess the relation of the designed protein, Fig. 9d examines the relation of the designed protein with other known sequences via a Basic Local Alignment Search Tool (BLAST) Tree [32]. Interestingly, the protein that was designed has certain relations with hypothetical proteins that had been explored in earlier work but not yet identified or solved structurally.

Using agentic modeling and using multiple X-LoRA models as elements in a multi-agent model, we further analyze the predicted sequence using our X-LoRA model. This takes advantage of the enhanced capability of our model to not only understand protein mechanics but to also utilize this in conjunction with other areas of knowledge. Here we use the prediction obtained as described above as an input, requesting a deeper analysis and design suggestions on variations of the sequence. We start with this task:

```
I have identified this amino acid sequence:  MSKSNTYRMLVLLEDDTKINKEDEKFLKGKPGKMHEFVDEL
ILPFNVDELDELNTWFDKFDAEICIPYVEILKESGMK.

How can I use it?

Provide logical steps, discuss specifically the engineering of the problem.
```

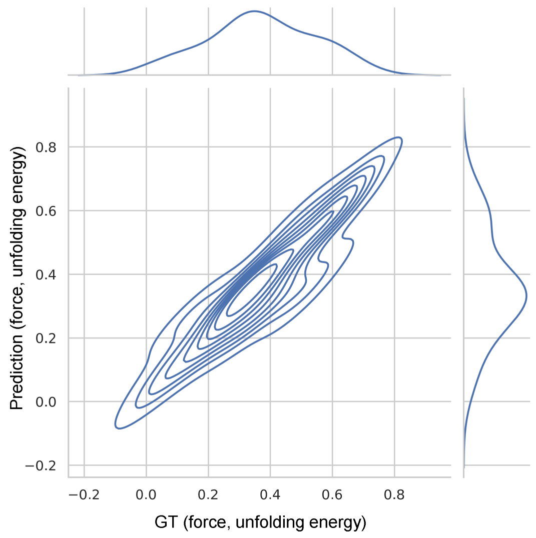


Figure 8: Overall X-LoRA performance for predicting unfolding force and unfolding energy from the sequence, comparing ground truth with predictions ($R_2 = 0.85$). The data shown here summarizes both unfolding force and unfolding energy for the test set.

We use multiple agentic interactions to develop the solution, specifically using adverserial strategies (details, see Materials and Methods), to push the model to explore deeper and more complex facets of the concepts discussed. The summary of this conversation is as follows, as identified by the X-LoRA model (for the full conversation, see Supplementary Material, SI Conv Example 1):

```
The conversation discusses the potential applications of a specific amino acid
sequence in protein engineering.  The sequence contains a mix of hydrophobic and
hydrophilic residues, suggesting it could form a stable structure and interact with
other molecules.  To engineer a protein with this sequence, computational methods
can be used to predict the secondary and tertiary structures, but limitations
include complexity of protein folding and computational cost.  Improvements can
be made by developing more efficient algorithms and incorporating new experimental
data.  Validation of computational models is crucial to ensure accuracy and
usefulness for practical applications.  Techniques such as cross-validation,
regularization, early stopping, ensemble methods, and transfer learning can be used
to prevent overfitting and underfitting.  Interpretation and explanation of models
can also help inform practical applications.
```

We also ask the model to develop a list of key points, resulting in:

- `The amino acid sequence MSKSNTYRMLVLLEDDTKINKEDEKFLKGKPGKMHEFVDELI LPFNVDELDELNTWFDKFDAEICIPYVEILKESGMK has potential applications in protein engineering due to its mix of hydrophobic and hydrophilic residues. Computational methods can be used to predict the secondary and tertiary structures of the protein, but limitations include complexity of protein folding and computational cost.`
- `Validation of computational models is crucial to ensure accuracy and usefulness for practical applications.  Techniques such as cross-validation, regularization, early stopping, ensemble methods, and transfer learning can be used to prevent overfitting and underfitting.`
- `Interpretation and explanation of models can also help inform practical applications.`

Finally, the key takeaway is:

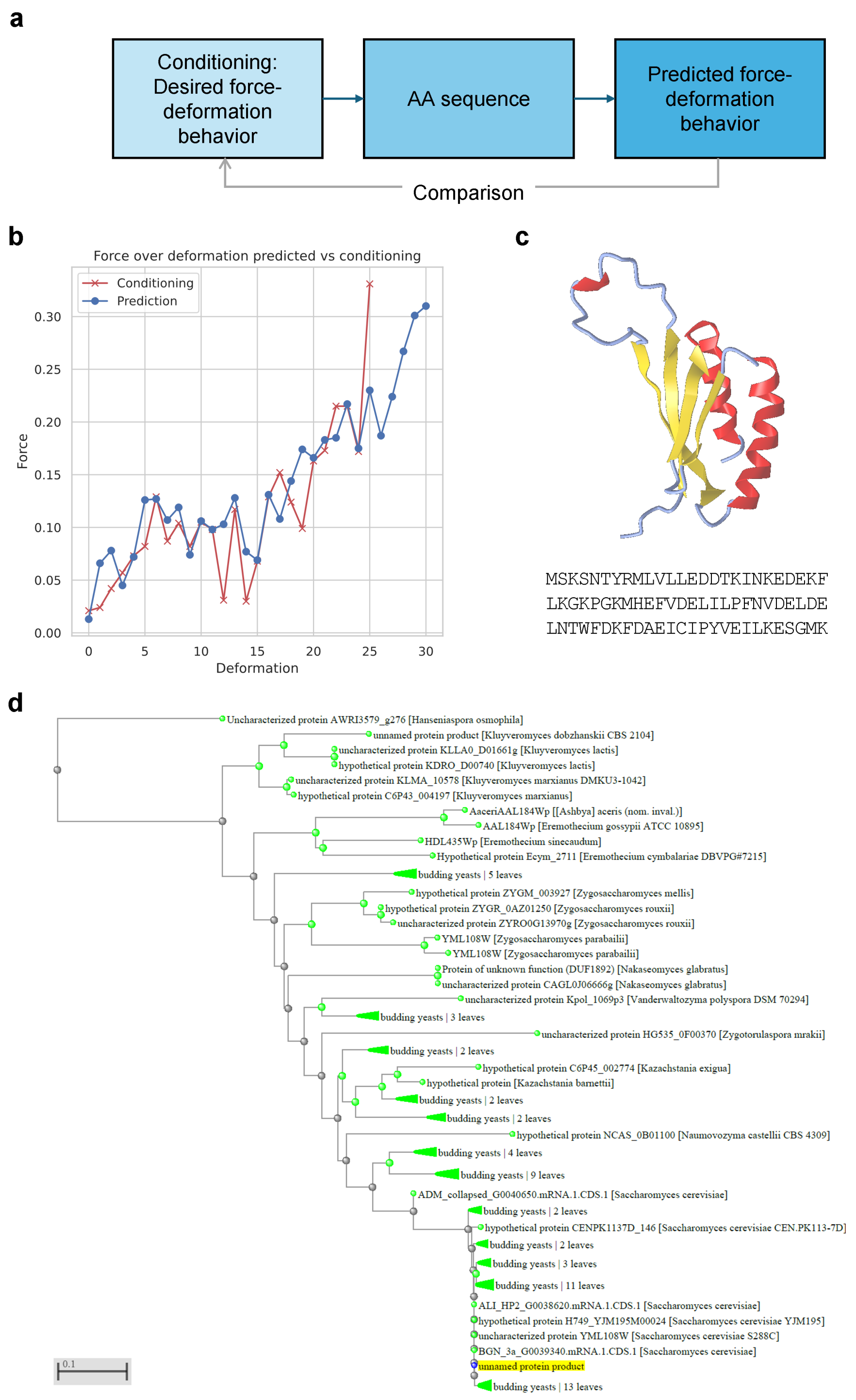


Figure 9: Using the generative task, we design a protein with a desired force-deformation behavior and then test the predicted sequence, examining whether or not it meets the desired target properties (panel a). Panel b shows the result for the sequence predicted, `MSKSNTYRMLVLLEDDTKINKEDEKFLKGKPGKMHEFVDELILPFNVDELDELNTWFDKF-DAEICIPYVEILKESGMK`, comparing the conditioning (target response) with the actual response predicted by the model. Note, we also confirm that a direct calculation of the peak force using `CalculateForce[...]` gives 0.305, in good agreement with the predictions from the force fistory instruction. Panel c shows a folded model of the protein (obtained using AlphaFold 2), and panel d examines the relation of the designed protein with other known sequences via a BLAST Tree [32].

```
The single most important takeaway is that the amino acid sequence
MSKSNTYRMLVLLEDDTKINKEDEKFLKGKPGKMHEFVDELILPFNVDELDELNTWFDKFDAEICIPYVEILKESGMK
has potential applications in protein engineering due to its mix of hydrophobic and
hydrophilic residues.  To engineer a protein with this sequence, computational
methods can be used to predict the secondary and tertiary structures, but
limitations include complexity of protein folding and computational cost.
Validation of computational models is crucial to ensure accuracy and usefulness
for practical applications, and techniques such as cross-validation, regularization,
early stopping, ensemble methods, and transfer learning can be used to prevent
overfitting and underfitting.  Interpretation and explanation of models can also
help inform practical applications.
```

We can also go into more depth to explore specific sequence engineering steps. To do this we ask the model to make more specific design suggestions to increase the stability of the protein:

```
How could I change the following protein sequence to make it more stable:
MSKSNTYRMLVLLEDDTKINKEDEKFLKGKPGKMHEFVDELILPFNVDELDELNTWFDKFDAEICIPYVEILKESGMK?
```

The summary of the interactions is (full conversation, see Supplementary Material, SI Conv Example 2):

```
The conversation discusses how to make a protein sequence more stable by introducing
hydrophobic amino acids and disulfide bonds.  The biologist suggests that
hydrophobic amino acids tend to promote protein stability, and disulfide bonds can
enhance stability as well.  The engineer asks for specific examples of hydrophobic
amino acids and how many disulfide bonds should be incorporated.  The biologist
provides examples of proteins that have been successfully stabilized using these
modifications, including insulin, bovine pancreatic trypsin inhibitor, and green
fluorescent protein.  The conversation also touches on potential limitations or
drawbacks of introducing hydrophobic amino acids or disulfide bonds, such as changes
in protein function, increased complexity, and potential for aggregation.  The
biologist suggests that these limitations can be addressed through careful selection
of amino acids, computational analysis, and experimental validation.
```

The list of key points:

- To make a protein sequence more stable, hydrophobic amino acids and disulfide bonds can be introduced.

  Hydrophobic amino acids tend to promote protein stability, while disulfide bonds can enhance stability as well.

  Examples of proteins that have been successfully stabilized using these modifications include insulin, bovine pancreatic trypsin inhibitor, and green fluorescent protein.
- Potential limitations or drawbacks of introducing hydrophobic amino acids or disulfide bonds include changes in protein function, increased complexity, and potential for aggregation.
- These limitations can be addressed through careful selection of amino acids, computational analysis, and experimental validation.

Finally, the key takeaway is:

```
The single most important takeaway in the conversation is that to make the protein
sequence MSKSNTYRMLVLLEDDTKINKEDEKFLKGKPGKMHEFVDELILPFNVDELDELNTWFDKFDAEICIPYVEILKESGMK
more stable, hydrophobic amino acids and disulfide bonds can be introduced.  This
answers the original question directly.
```

From a protein engineering perspective, these suggestions make sense as they would indeed lead to more stable designs. Specific examples and strategies are developed and proposed that researchers can then explore further.

In another example, we consider these three sequences. We first ask the model to compute the strength of each, and then task it to reason over the results: Find the strongest candidate, and explain why that particular protein is likely the strongest. Moreover, we ask the model to explain steps to manufacture it in the lab. The three sequences (created manually by manipulating one of the sequences in the test set) are:

- `LNTWFDKFDAEICIPYVEILKESGMK`
- `AITWFDKFDAEICIPYVEALKIAAMK`
- `AAAAAAAAKFDAAEICIIAAAAAAAAKIAAMK`

This example tests whether the model can combine different capabilities and do this dynamically as a conversation progresses. That is, to dynamically switch between different capabilities such as protein mechanics, rational analysis, generation of new hypotheses, and so on.

The conversation is summarized in Figure 10, along with a depiction of how the scalings change over the course of the conversation (the analysis shows both the deep layer-wise adpter scaling and a summed analysis of such to offer a measure for the effective use of the various experts). The results clearly show how the protein task is initially highly relevant, gradually shifting towards a more prominent representation of the bioinspired adapter and the biology adapter. At the end of the conversation, the biology adapter is the most prominent one.

To analyze the predictions as we want to assess the quality of the responses, the three proteins considered in this task are depicted in Figure 11 (proteins folded using Alpha Fold 2 [31], via ColabFold [33]). We find that the most stable structure in the middle features the most organized secondary organization, in agreement with the notion that it yields the highest unfolding force, as predicted by the X-LoRA model in Figure 10. The other two are less organized; the first one being largely helical with some turns, and the last one being partially unstructured and with a kink in the geometry. These features explain the lowest observed unfolding force as predicted by the model.

By analyzing the visual representations of the folded structures we confirm that the most stable structure in the middle features the most organized secondary organization, in agreement with the notion that it yields the highest unfolding force as predicted by the model. We find that the other two models are less organized. The first one being largely helical with some turns, and the last one being partially unstructured and with a kink in the geometry. We believe that these features likely explain the lowest observed unfolding force as predicted by the model, in agreement with the X-LoRA model predictions.

In a second example we focus on energetic properties of proteins, considering the unfolding energy as the protein is pulled end-to-end from its initial state to a fully unfolded geometry [29]. We again consider three sequences (the first two are identical to the ones considered above, but the third one consists of a longer assembly, representing a mix of the second sequence at the beginning/end and the first sequence in the center):

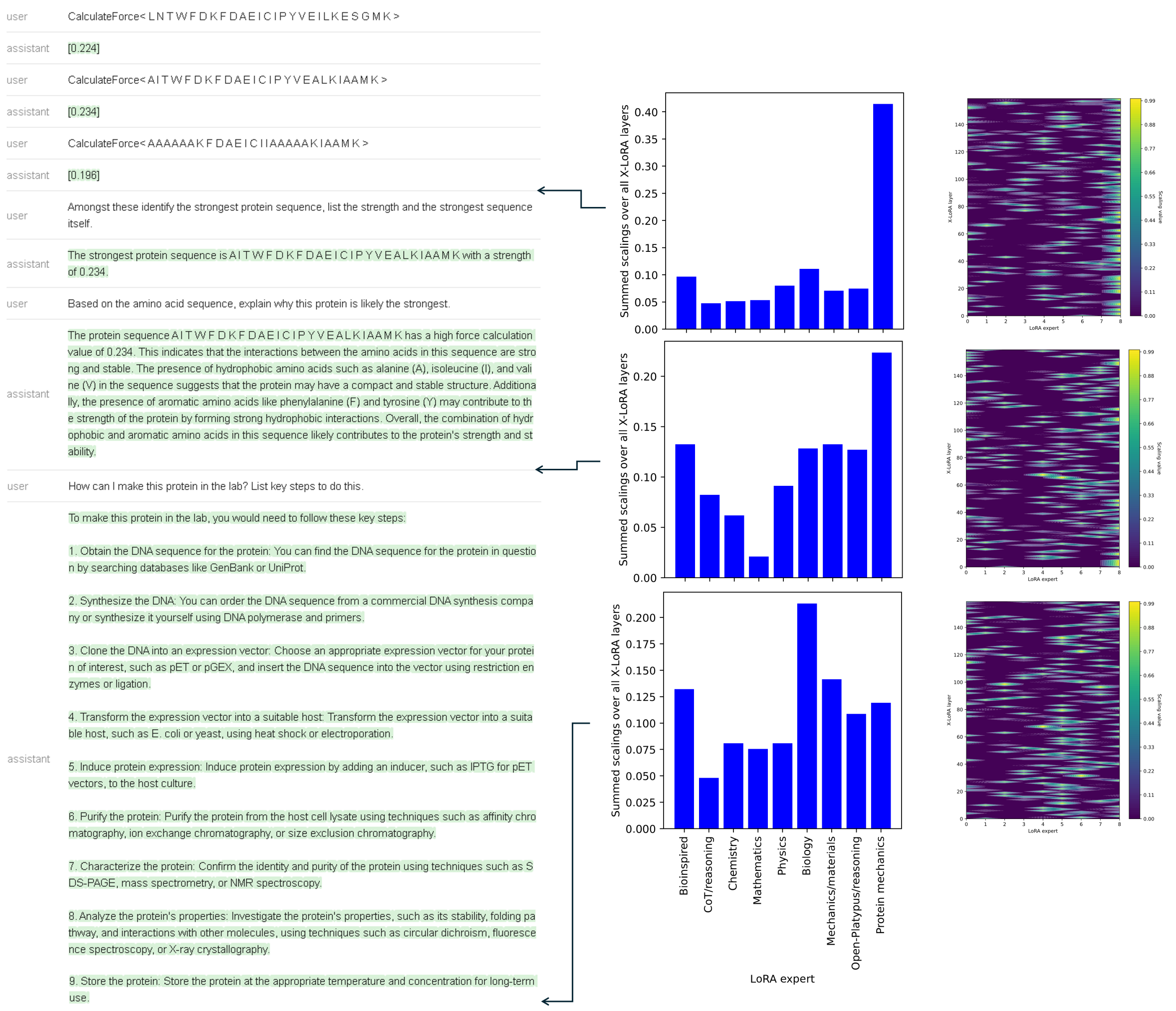
- `LNTWFDKFDAEICIPYVEILKESGMK`
- `AITWFDKFDAEICIPYVEALKIAAMK`
- `AITWFDKFDAEICIPYVEALKIAAMKLNTWFDKFDAEICIPYVEILKESGMKAITWFDKFDAEICIPYVEALKIAAMK`

We first ask the model to compute the unfolding energy of each, and then to reason over the results: Based on the amino acid sequence, we ask to explain why this protein is likely the most stable. At the end, we query for likely functions that the most stable protein may have.

Figure 12 shows a record of a conversation that invokes combinations of distinct experts and integrating knowledge across these capabilities, here applied to unfolding energy of three proteins. At the end of the conversation, the CoT/reasoning adapter is the most prominent one as the X-LoRA model responds to a query to reason over the results and predict likely protein structure features as well as potential functions of the protein. Specifically, the mechanistic question about the origin of protein stability include the emergence of hydrophobic interactions between the nonpolar amino acids as well as hydrogen bonding between the polar amino acids. The model predicts that there are several hydrophobic amino acids (I, F, W, Y) and hydrogen bonding amino acids (D, E, K) that contribute to the stability of the protein. The model also predicts that the presence of cysteine (C) residues in the sequence would lead to the formation of disulfide bonds that further stabilize the protein structure. It is remarkable that key predictions about the structural features, as explanation for the high stability, are correctly identified as can be verified via structural analyses shown in Fig. 13. The analysis in this figure confirms a presence of all these features. It is also clear why the other two proteins, much shorter segments with variable stability of helical domains, feature a much lower unfolding energy.

2.4 Adversarial agentic modeling to connect distinct scholarly disciplines and knowledge yielding ontological knowledge graph generation

Since our X-LoRA model has deep knowledge across domains, we can use it to probe connections between disparate ideas, knowledge bases, and areas of expertise. In one experiment we use the model to ask two queries, each formulated



user CalculateForce< L N T W F D K F D A E I C I P Y V E I L K E S G M K >

assistant [0.224]

user CalculateForce< A I T W F D K F D A E I C I P Y V E A L K I A A M K >

assistant [0.234]

user CalculateForce< A A A A A A K F D A E I C I I A A A A A K I A A M K >

assistant [0.196]

user Amongst these identify the strongest protein sequence, list the strength and the strongest sequence itself.

assistant The strongest protein sequence is A I T W F D K F D A E I C I P Y V E A L K I A A M K with a strength of 0.234.

user Based on the amino acid sequence, explain why this protein is likely the strongest.

assistant The protein sequence A I T W F D K F D A E I C I P Y V E A L K I A A M K has a high force calculation value of 0.234. This indicates that the interactions between the amino acids in this sequence are stro ng and stable. The presence of hydrophobic amino acids such as alanine (A), isoleucine (I), and vali ne (V) in the sequence suggests that the protein may have a compact and stable structure. Additiona lly, the presence of aromatic amino acids like phenylalanine (F) and tyrosine (Y) may contribute to th e strength of the protein by forming strong hydrophobic interactions. Overall, the combination of hydr ophobic and aromatic amino acids in this sequence likely contributes to the protein's strength and st ability.

user How can I make this protein in the lab? List key steps to do this.

assistant To make this protein in the lab, you would need to follow these key steps:

1. Obtain the DNA sequence for the protein: You can find the DNA sequence for the protein in questio n by searching databases like GenBank or UniProt.

2. Synthesize the DNA: You can order the DNA sequence from a commercial DNA synthesis compa ny or synthesize it yourself using DNA polymerase and primers.

3. Clone the DNA into an expression vector: Choose an appropriate expression vector for your protei n of interest, such as pET or pGEX, and insert the DNA sequence into the vector using restriction en zymes or ligation.

4. Transform the expression vector into a suitable host: Transform the expression vector into a suita ble host, such as E. coli or yeast, using heat shock or electroporation.

5. Induce protein expression: Induce protein expression by adding an inducer, such as IPTG for pET vectors, to the host culture.

6. Purify the protein: Purify the protein from the host cell lysate using techniques such as affinity chro matography, ion exchange chromatography, or size exclusion chromatography.

7. Characterize the protein: Confirm the identity and purity of the protein using techniques such as S DS-PAGE, mass spectrometry, or NMR spectroscopy.

8. Analyze the protein's properties: Investigate the protein's properties, such as its stability, folding pa thway, and interactions with other molecules, using techniques such as circular dichroism, fluoresce nce spectroscopy, or X-ray crystallography.

9. Store the protein: Store the protein at the appropriate temperature and concentration for long-term use.

Figure 10: Complex conversation that invokes combinations of distinct experts and integrating knowledge across these capabilities. The right panels show a depiction of how the scalings change over the course of the conversation. The results clearly show how the protein mechanics task is initially highly relevant, gradually shifting towards a more prominent representation of the bioinspired adapter and the biology adapter. At the end of the conversation, the biology adapter is the most prominent one, reflecting the focus on key elements of biology as the model responds to biological manufacturing steps of the protein.

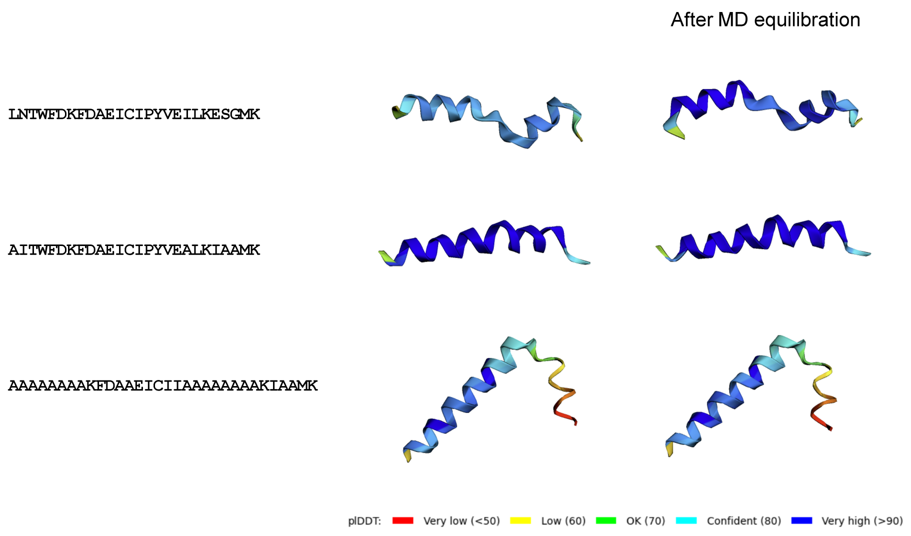

After MD equilibration
LNTWFDKFDAEICIPYVEILKESGMK
AITWFDKFDAEICIPYVEALKIAAMK
AAAAAAAAKFDAAEICIIAAAAAAAAKIAAMK
pIDDT: Very low (<50) Low (60) OK (70) Confident (80) Very high (>90)

Figure 11: Visual representation of the folded proteins considered in Figure 10, showing as-predicted (left column) and MD equilibrated results (right column). The most stable structure in the middle (`AITWFDKFDAEICIPYVEALKIAAMK`) features the most organized secondary organization, in agreement with the result that it yields the highest unfolding force. The other two are less organized; the first one being largely helical with some turns, and the last one being partially unstructured and with a kink in the geometry. These features likely explain the lowest observed unfolding force as predicted by the model.

similarly but using a distinct focus. Then, we generate triplets to yield an ontological knowledge graph to distill the answers into more structured outputs [34, 35]. The resulting graph provides an integrated understanding of the generated insights and visualizes connections between concepts in an interpretable and mechanistic manner.

We use two adversarial agents to answer the question. Agent 1 is designated as a "critical physicist", and Agent 2 as a "philosopher". Agent 1 poses the initial question and is tasked to probe and develop additional questions based on every response of Agent 2. As the conversation unfolds, the conversation goes deeper into the subject matter and uncovers a variety of dimensions. The query posed is intentionally chosen to cover multiple areas of disciplinary knowledge to examine the ability of the model to integrate ideas and concepts.

The query is:

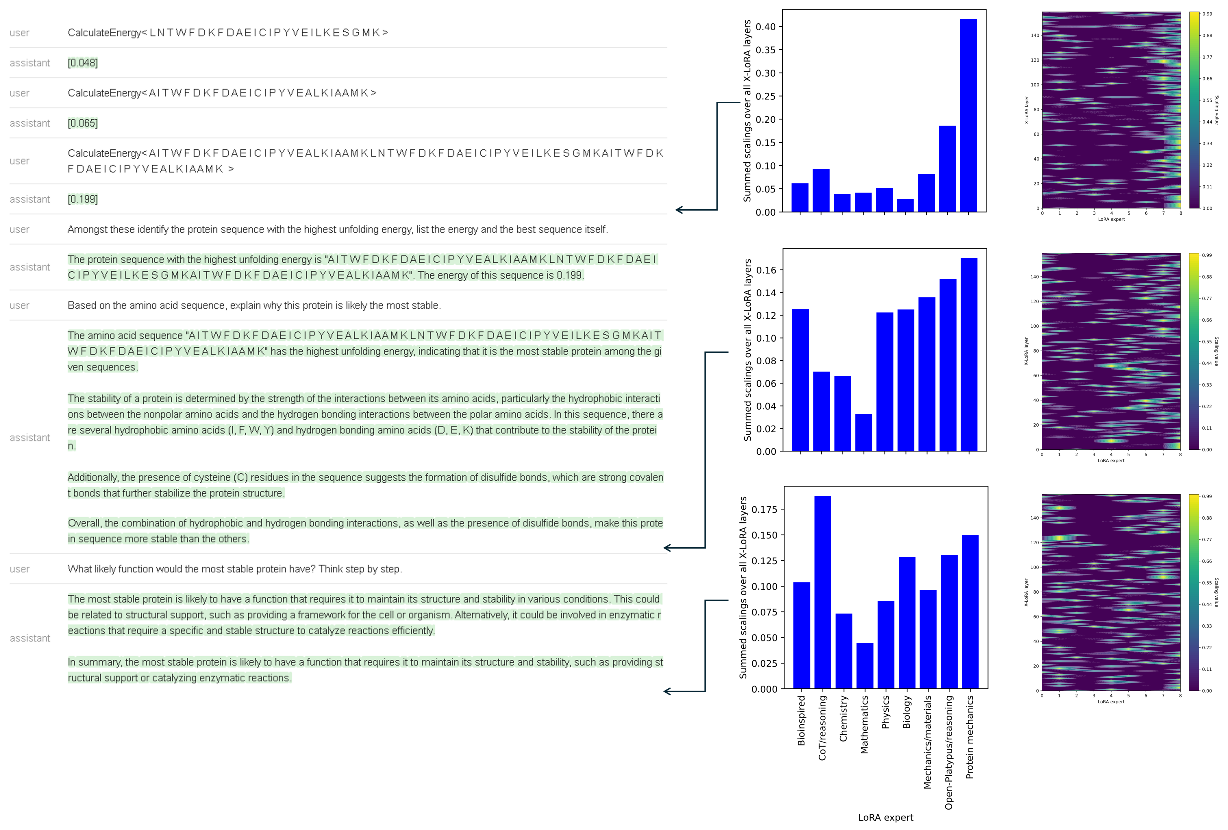
```
Consider an abstract representation of a classical musical composition as a graph of
interacting notes, chords, melodies, rhythms and so on.

What happens if we would apply pressure, up to a point of failure, on this abstract
concept of music?

Provide logical steps, mathematical reasoning, and discuss specifically the physics
of the problem.
```

The full conversation is shown in the Supplementary Material (see, SI Conv Example 3). For sake of brevity, we simply report the summary and key takeaway here, which captures the essential features:

```
The conversation discusses the application of pressure to an abstract representation
of a classical musical composition, considering it as a physical system with
interacting notes, chords, melodies, and rhythms.  The pressure can be quantified
using stress, and the critical point of failure is determined when the stress
reaches a critical value.  To test the predictions of the mathematical model, a
physical model of the abstract musical composition can be used, and experimental
observations can be compared to the predictions.  The limitations of the proposed
mathematical model include a simplified representation of interactions, static
representation, homogeneous distribution, and linear relationship between stress
and pressure.  To improve the accuracy and validity of the model, more detailed and
realistic models, dynamic effects, non-uniform distribution, and non-linear terms
```



user	CalculateEnergy< L N T W F D K F D A E I C I P Y V E I L K E S G M K >
assistant	[0.048]
user	CalculateEnergy< A I T W F D K F D A E I C I P Y V E A L K I A A M K >
assistant	[0.065]
user	CalculateEnergy< A I T W F D K F D A E I C I P Y V E A L K I A A M K L N T W F D K F D A E I C I P Y V E I L K E S G M K A I T W F D K F D A E I C I P Y V E A L K I A A M K >
assistant	[0.199]
user	Amongst these identify the protein sequence with the highest unfolding energy, list the energy and the best sequence itself.
assistant	The protein sequence with the highest unfolding energy is "A I T W F D K F D A E I C I P Y V E A L K I A A M K L N T W F D K F D A E I C I P Y V E I L K E S G M K A I T W F D K F D A E I C I P Y V E A L K I A A M K". The energy of this sequence is 0.199.
user	Based on the amino acid sequence, explain why this protein is likely the most stable.
assistant	The amino acid sequence "A I T W F D K F D A E I C I P Y V E A L K I A A M K L N T W F D K F D A E I C I P Y V E I L K E S G M K A I T W F D K F D A E I C I P Y V E A L K I A A M K" has the highest unfolding energy, indicating that it is the most stable protein among the gi ven sequences. The stability of a protein is determined by the strength of the interactions between its amino acids, particularly the hydrophobic interacti ons between the nonpolar amino acids and the hydrogen bonding interactions between the polar amino acids. In this sequence, there a re several hydrophobic amino acids (I, F, W, Y) and hydrogen bonding amino acids (D, E, K) that contribute to the stability of the protei n. Additionally, the presence of cysteine (C) residues in the sequence suggests the formation of disulfide bonds, which are strong covalen t bonds that further stabilize the protein structure. Overall, the combination of hydrophobic and hydrogen bonding interactions, as well as the presence of disulfide bonds, make this prote in sequence more stable than the others.
user	What likely function would the most stable protein have? Think step by step.
assistant	The most stable protein is likely to have a function that requires it to maintain its structure and stability in various conditions. This could be related to structural support, such as providing a framework for the cell or organism. Alternatively, it could be involved in enzymatic r eactions that require a specific and stable structure to catalyze reactions efficiently. In summary, the most stable protein is likely to have a function that requires it to maintain its structure and stability, such as providing st ructural support or catalyzing enzymatic reactions.

Figure 12: Transcript of a conversation that invokes combinations of distinct experts and integrating knowledge across these capabilities, here applied to unfolding energy of three proteins. The depiction of how the scalings change over the course of the conversation is shown in the right part of the figure. The results clearly show how the protein mechanics task is initially highly relevant, gradually shifting towards a more integrated representation of a variety of adapters. At the end of the conversation, the CoT/reasoning adapter is the most prominent one as the X-LoRA model responds to a query to reason over the results and predict likely protein structure features. Key predictions about the structural features, as explanation for the high stability, are correctly identified as can be verified via structural analyses shown in Fig. 13.

```
should be incorporated.

[...]

The single most important takeaway is that applying pressure to an abstract
representation of a classical musical composition can be thought of as changing
the force field between the interacting notes, chords, melodies, and rhythms.  As
the pressure increases, the stress in the system also increases, and at a critical
point, the system will fail and collapse under its own weight.  This insight answers
the original question by providing a physical explanation for how pressure affects
the abstract concept of music and how it can be quantified using stress.
```

While the generated text provides useful insights, the construction of ontological representation of the data can be beneficial for interpretable analysis [8]. For graph generation, we concatenate the entirety of the two conversation and construct a knowledge graph. An example graph is show in Figure 14. The resulting graph is divided into communities using the Girvan-Newman algorithm [36], and a total of 12 communities are identified. The figure shows both the overall graph (Figure 14a) and detailed views with node labels (Figure 14b-c).

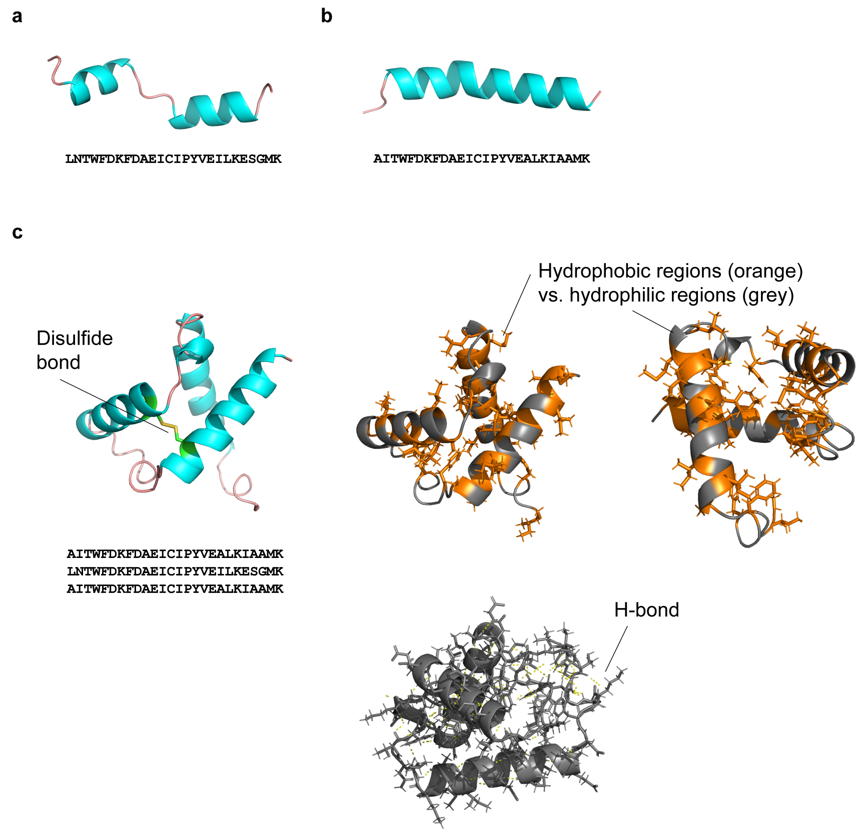

a
b
LNTWFDKFDAEICIPYVEILKESGMK
AITWFDKFDAEICIPYVEALKIAAMK
c
Hydrophobic regions (orange) vs. hydrophilic regions (grey)
Disulfide bond
AITWFDKFDAEICIPYVEALKIAAMK
LNTWFDKFDAEICIPYVEILKESGMK
AITWFDKFDAEICIPYVEALKIAAMK
H-bond

Figure 13: Visual representation of the folded proteins considered in Figure 12 (color reflects the secondary structure). Panels a and b show the first two sequences that are identified to have the lowest unfolding energy. The most stable structure is shown in panel c, featuring a more complex organization with variegated bonding mechanisms. As correctly predicted by the X-LoRA model, the reason for the higher unfolding energy is the presence of disulfide bonds, a mix of hydrophophic/hydrophilic regions, and a number of H-bonds. The presence of structured hydrophilic regions indicates the portions of the protein that interact with water differently, which is likely a critical factor in the folding and function of the protein.

2.5 Development of X-LoRA-Gemma with combined protein, chemical, bio-inspired and mechanics of materials capabilities

To show that the proposed approach works with other base models that have distinct architectures, we trained another X-LoRA model, this time based on the Gemma-7B-it model [37]. In this model we use a set of four LoRA adapters, defined as follows:

1. Bioinspired materials
2. Mechanics and materials
3. Protein mechanics tasks (featuring generative sequence-to-property and inverse capabilities)
4. Quantum-mechanics based molecular properties QM9 (featuring generative SMILES-to-property and inverse capabilities, see Table 2 for a definition and summary of all properties computed in that dataset) [38, 39]

This X-LoRA-Gemma model is developed in a similar as the original X-LoRA model but aims to feature distinct sets of capabilities, in particular adding the ability to predict a set of 12 quantum mechanical properties and to design molecules to meet a set of 12 quantum mechanical properties. This new adapter is trained along with the other ones listed above that are based on the same datasets we used in the earlier experiment. Note, whereas the bulk of the paper focuses on the Zephyr-based X-LoRA model, in the scope of this section we need to distinguish the models and

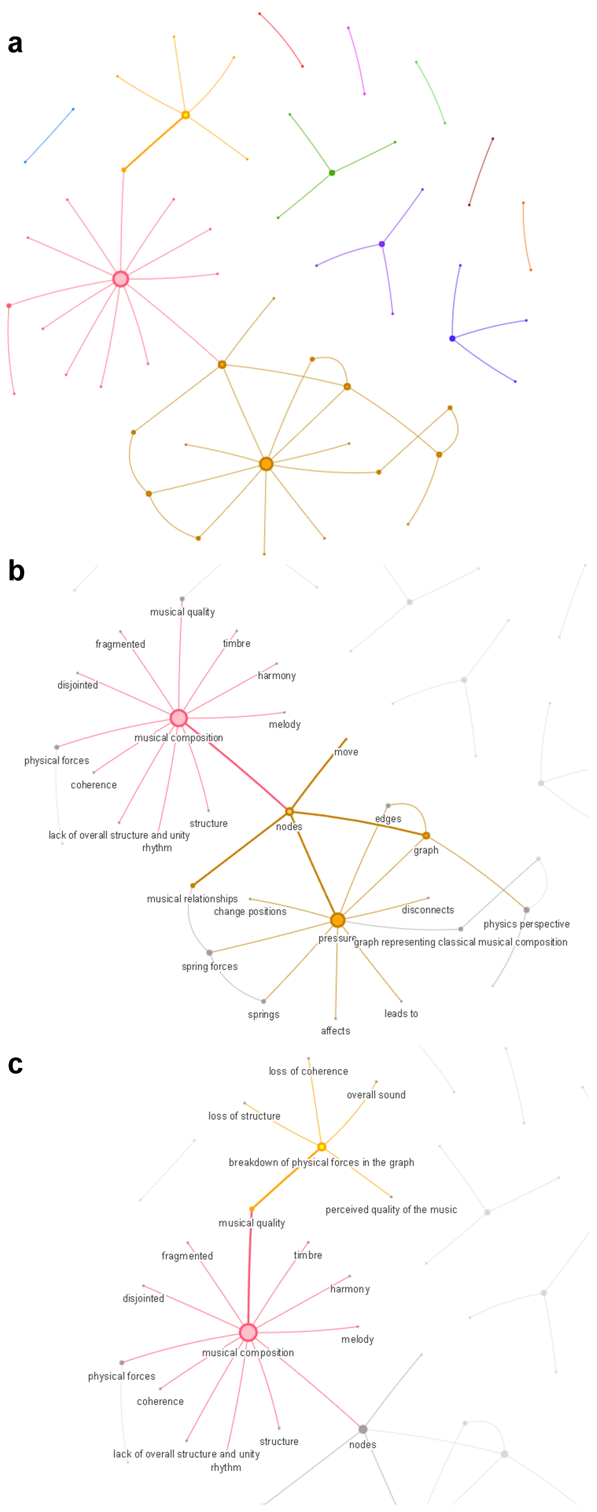


Figure 14: Ontological knowledge graph generated using agentic modeling in response to the question to explore an abstract representation of a classical musical composition as a graph of interacting notes, chords, melodies, rhythms that is exposed to pressure. The model then responds by integrating concepts from music and art with physics, reasoning and mechanics to develop an answer. The graph representation of the produced response offers visual insights into the results and provides a human-readable analysis of the critical points made. The color reflects the 12 distinct communities identified. Panel a shows the overall graph, and panels b and c show close-up views with node labels for a deeper understanding of the result.

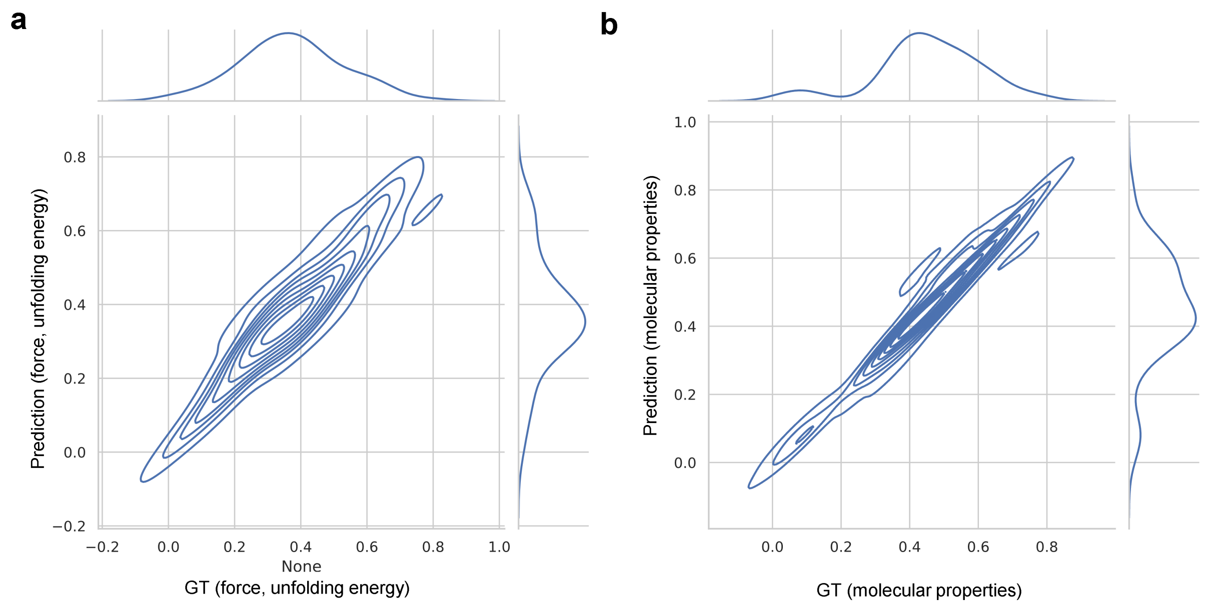


Figure 15: Overall X-LoRA-Gemma performance for predicting unfolding force and unfolding energy from the sequence comparing ground truth with predictions (panel a, $R_2 = 0.87$), as well as performance for prediction of quantum mechanical properties of molecules based on the QM9 dataset (panel b, $R_2 = 0.96$). We observe excellent performance for both tasks.

Table 2: Summary of QM9 molecular properties, as defined in [38, 39].

Label	Definition
mu	Dipole moment: A measure of the separation of charge within the molecule, affecting its interaction with electric fields and other molecules.
alpha	Polarizability: Indicates how much the electron cloud around the molecule distorts in an external electric field, influencing the molecule's optical properties and interactions.
homo	Highest Occupied Molecular Orbital (HOMO) energy: Related to the energy of the highest occupied electron orbital, important for understanding a molecule's chemical reactivity.
lumo	Lowest Unoccupied Molecular Orbital (LUMO) energy: Pertains to the energy of the lowest unoccupied electron orbital, also critical for reactivity and optical properties.
gap	HOMO-LUMO gap: The energy difference between HOMO and LUMO, significant for determining a molecule's chemical stability and reactivity.
r2	Electronic spatial extent: This is a measure of the size of the electron cloud of a molecule, related to its electronic properties.
zpve	Zero-point vibrational energy: The energy of a molecule at its lowest vibrational state, contributing to its stability and reactivity.
cv	Heat capacity at constant volume: Relates to the amount of heat required to change the temperature of a molecule by a certain amount, important for thermodynamics.
u0	Internal energy at 0 K: The total energy of a molecule including electronic, vibrational, rotational, and translational contributions at absolute zero.
u298	Internal energy at 298.15 K: Similar to u0, but measured at room temperature (approximately 25°C).
h298	Enthalpy at 298.15 K: The total heat content of a molecule at room temperature, including internal energy and the product of pressure and volume.
g298	Free energy at 298.15 K: Gibbs free energy of the molecule at room temperature, indicating the maximum amount of work obtainable from a thermodynamic process at constant temperature and pressure.

hence refer to the original X-LoRA model as X-LoRA-Zephyr, and the Gemma-based model as X-LoRA-Gemma. Figure 15 shows the overall X-LoRA performance, featuring predictions of unfolding force and unfolding energy from the sequence ($R_2 = 0.87$), as well as performance for prediction of quantum mechanical properties of molecules based on the QM9 dataset ($R_2 = 0.96$). We observe excellent performance for both tasks, even slightly better than for the original X-LoRA model for the protein design task.

We now demonstrate the capabilities of X-LoRA-Gemma in a simple multi-agent design experiment. We consider a small organic molecule, `CCCC12CC1C(=O)N2` (taken from the test set based on the QM9 dataset). This is a bicyclic structure with a ketone (=O) and an amine (N) incorporated into the ring, see Figure 17(a) for a visualization of the molecule). First we ask the X-LoRA-Gemma model:

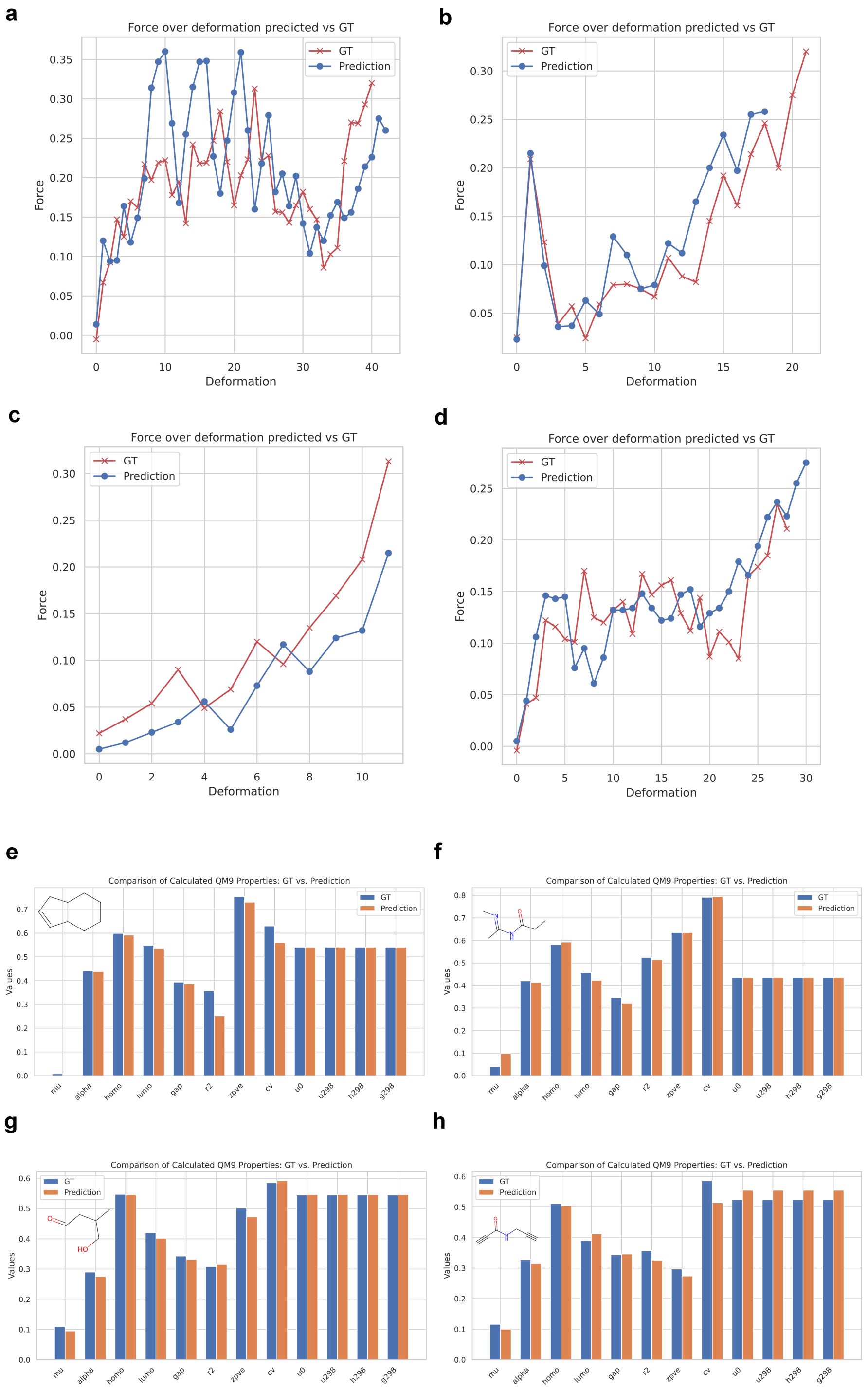


Figure 16: Predictions using the X-LoRA-Gemma model that is capable of predicting various biophysical properties of proteins (a-d) based on their amino acid sequence and quantum mechanical properties (e-h) of molecules represented in SMILES language. Panels a-d, to predict force-deformation behavior from amino acid sequences, for four examples from the test set. The model has excellent forward capabilities. Panels e-f depicts four representative examples of predicting quantum mechanical properties based on SMILES representation of molecular structures. Also here, excellent predictive capabilities are found. SMILES codes are (from e to f): `C1C=CC2CCCCC12`, `CCC(=O)NC(=NC)C`, `CC(CO)CC=O` and `CCN1CC(C1)(C#C)O`. See Table 2 for a definition of the properties predicted here.

```
Here is a molecule in SMILES code:  CCCC12CC1C(=O)N2

What changes would I need to make to make this molecule more useful for polymer
design.
```

The response is:

```
To make this molecule more useful for polymer design, you could:
```

1. Introduce a reactive functional group, such as an amine or hydroxyl group, to allow for further modification and crosslinking.
2. Include a longer, more flexible carbon backbone to increase the molecule's ability to form polymers with different structures and properties.
3. Add a side chain with a functional group, such as a carboxyl group, to allow for the formation of covalent bonds with other molecules and polymers.
4. Incorporate a cyclic structure into the molecule, such as a lactam ring, to create a more rigid and stable polymer.
5. Include a variety of side chains to create a range of properties and functionalities in the polymer.
6. Use a variable number of carbons in the backbone to allow for the creation of polymers with different molecular weights and properties.
7. Introduce a crosslinking point into the molecule to create a more stable and less reactive polymer.
8. Include a repeating unit structure to create a polymer with a specific molecular weight and properties.
9. Use a combination of the above strategies to create a versatile and useful polymer for various applications.

In the following we focus on the first suggestion.

```
How can I increase the general reactivity of a molecule?  I want to use this
molecule to build a polymer material.

The molecule of interest, in SMILES code, is:  CCCC12CC1C(=O)N2

I have access to tuning the following properties:

Dipole moment, mu Polarizability, alpha
Highest Occupied Molecular Orbital energy, homo
Lowest Unoccupied Molecular Orbital energy, lumo
HOMO-LUMO gap:  The energy difference between HOMO and LUMO, gap
Electronic spatial extent
Zero-point vibrational energy
Heat capacity at constant volume
Internal energy at 0 K
Internal energy at 298.15 K
Enthalpy at 298.15 K
Free energy at 298.15 K

What are the top THREE properties I should change, and how should I modify them
(increase or decrease).
```

The response from X-LoRA-Gemma is:

- HOMO-LUMO gap (decrease)
- Dipole moment (increase)
- Polarizability (increase)

```
By modifying these properties, you can increase the general reactivity of the
molecule and make it more suitable for polymer material building.
```

We use the generative and forward capabilities of X-LoRA-Gemma to design for these changes of molecular properties. To do the optimization we use a two-agent setup consistent of two instances of X-LoRA-Gemma, where one agent uses the generative task `GenerateMolecularProperties<...>` to generate candidates. We then use a second agent to do a calculation using `CalculateMolecularProperties<...>` and assess the error between the target and the predicted properties. This process is repeated until we reach the maximum number of iterations or until a minimum threshold for the mean-squared error is met. Figure 17 shows the results of the entire process. Figure 17(a) depicts the target design and how it deviates from the properties of the original molecule. Figure 17(b) shows the top 20 designs generated by the algorithm, along with the distribution of the mean-squared error (Figure 17(c)) and a plot of the mean-squared error over the generated designs (Figure 17(d)). The best design is `CC(C)C1=CC(=O)NC=C1` and it meets the target relatively well, as shown in Figure 17(e). Finally, Figure 17(f) shows an optimized 3D molecular representation (we use UFF [40] to optimize the molecule's 3D structure).

Since we wanted to assess the molecule's reactivity, we also compute the Gasteiger charges [41] (Figure 17(f) for a visual overview). The molecule features an aromatic ring containing a ketone (C=O), which is more electrophilic compared to carbon-carbon or carbon-hydrogen bonds, making it susceptible to nucleophilic attacks. The combination of the aromatic ring containing nitrogen and the ketone group forms a pyridone ring structure. This ring is more reactive than a simple aromatic ring because the presence of the heteroatom (nitrogen) and the ketone group introduces sites for potential chemical reactions, such as nucleophilic addition to the carbonyl carbon or electrophilic substitution at the ring. The nitrogen within the ring, being part of the heteroaromatic system, contributes to the aromaticity and reactivity of the compound by altering the electron density and distribution across the ring. This can affect how the molecule interacts with other chemical entities, making the nitrogen-containing ring a site for potential chemical reactions, particularly those involving electrophilic attack on the ring due to the electron-donating nature of the nitrogen atom. The carbon atom in the ring, which is adjacent to both an oxygen and a nitrogen atom, having a partial positive charge (+0.25) suggests it is somewhat electron-deficient. This happens because oxygen, being more electronegative, pulls electron density towards itself, making the carbon less electron-rich. The effect is somewhat balanced by the adjacent nitrogen, which is less electronegative than oxygen but more electronegative than carbon. However, the presence of the partial positive charge indicates that the carbon is a likely site for nucleophilic attacks, where a nucleophile (an electron-rich species) would be attracted to this electron-deficient carbon. The nitrogen atom having a partial negative charge (-0.33) indicates it is relatively electron-rich compared to its usual state. This can be due to its lone pair of electrons and the effect of the aromatic ring's electron system. In aromatic heterocycles, nitrogen's lone pair can participate in the delocalized π-electron system, but the exact contribution depends on the ring's structure and the electronegativity of adjacent atoms. The partial negative charge suggests that nitrogen is more nucleophilic, meaning it has a higher propensity to donate electrons, either forming bonds with electrophiles or engaging in protonation reactions. Further quantum mechanical calculations should be done to investigate this. The carbon with a partial positive charge is a reactive site for nucleophilic attacks. Other molecules might react with this site through mechanisms that involve nucleophiles targeting electron-deficient carbons. The hydrogen atom bonded to the nitrogen atom, with a positive partial charge of 0.17, is a good candidate for forming hydrogen bonds with other molecules or atoms. Hydrogen bonds are a type of dipole-dipole interaction that occurs when a hydrogen atom, which is bonded to a highly electronegative atom (e.g.: nitrogen, oxygen, or fluorine), interacts with another electronegative atom bearing a lone pair of electrons. This makes the molecule potentially capable of acting as a hydrogen bond donor, which is an important factor in determining the molecule's solubility in water and other polar solvents, as well as its interactions in biological systems.

The presence of the ketone group adjacent to the nitrogen-containing aromatic ring offers unique reactivity that can be exploited in polymer synthesis [42, 43, 44, 44]. This molecule is a type of lactam, given the cyclic amide formed from the ketone and nitrogen in the ring. Lactams are known for their role in polymer chemistry, particularly in the synthesis of polyamides, which are a class of polymers extensively used in materials applications. The ketone could undergo various chemical reactions that are useful in polymer chemistry, such as nucleophilic addition. This could be utilized to extend the polymer chain or to introduce side chains that could modify the polymer's properties. The presence of a nitrogen atom in the aromatic ring can significantly affect the electronic properties of the polymer and can participate in hydrogen bonding, which could impact the polymer's melting point, solubility, and mechanical properties. N-heteroaromatic compounds are also known for their ability to coordinate with metal ions, which could lead to applications in areas such as catalysis or materials with electronic or photonic properties. This molecule could potentially undergo polymerization reactions through the amide (lactam) functionality; polyamides are known for their strength, elasticity, and resistance to wear and chemicals and they are used in a wide range of settings from textiles (e.g., nylon) to automotive parts [44, 44]. Alkyl substituents may increase the hydrophobic character of the molecule we anticipate that polymers derived from monomers with such alkyl groups are likely to be more hydrophobic, affecting their solubility in different solvents and their interaction with water. This could be advantageous for applications requiring water resistance or specific solubility characteristics. The presence of alkyl groups can also affect the mechanical properties of the polymer. For instance, they can act as plasticizers within the polymer matrix, increasing the flexibility of the polymer. This could lead to materials that are more pliable or have better impact resistance, depending on the length and branching of the alkyl groups.

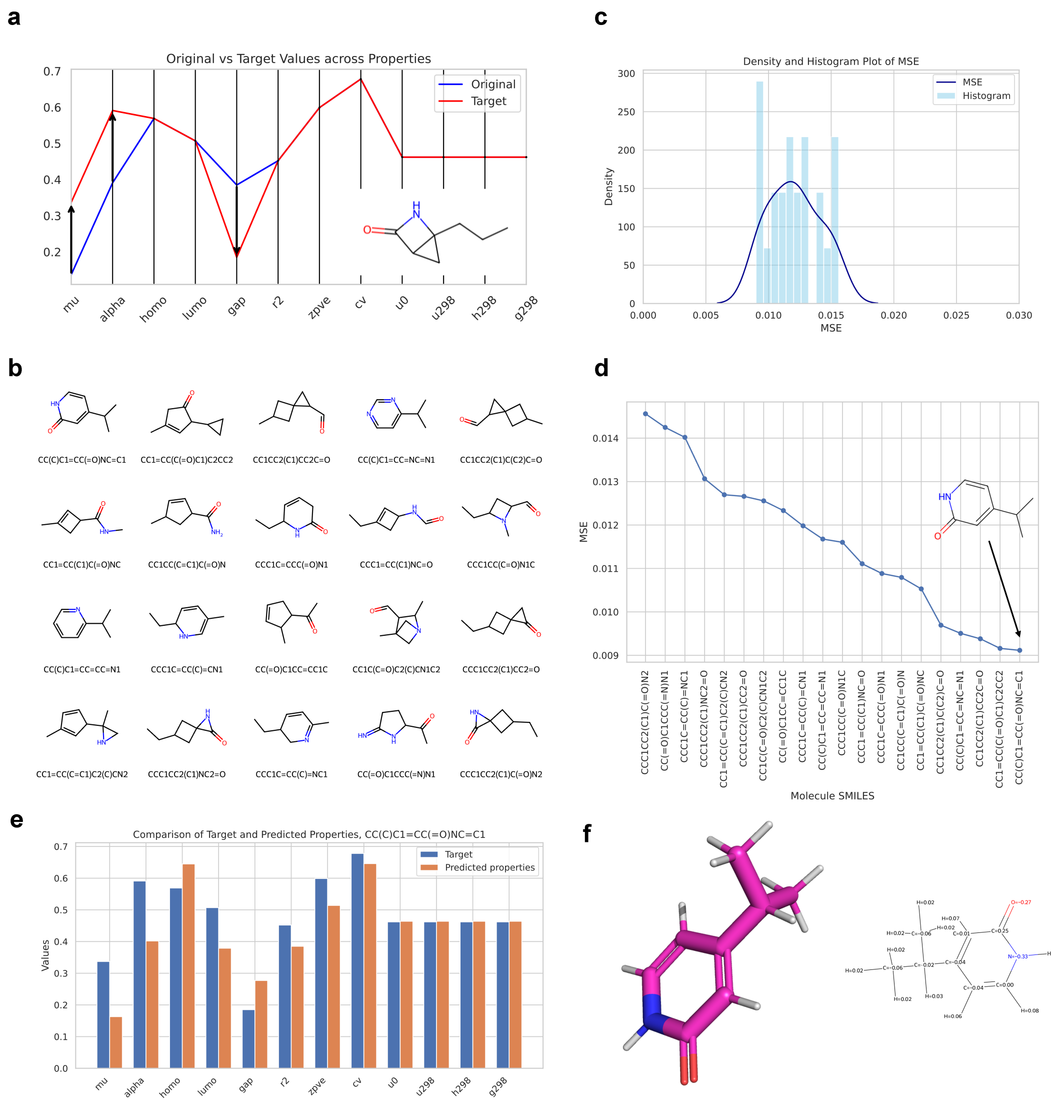


Figure 17: Design of novel molecules to meet a target objective. Here, we use `CC(C)C1=CC(=O)NC=C1` as a basis, and attempt to make the molecule more reactive (panel a shows target and design changes). Panel b shows the top 20 designs, panel c the distribution of the mean-squared error (MSE), and panel d the MSE over the top generated designs. Finally, panel e shows an assessment of how well the target matches with the predicted properties. We depict an optimized 3D molecular representation in panel f, along with a representation of atoms and Gasteiger partial charges [41] (a method of estimating the distribution of electric charge over the atoms in a molecule). A close analysis of the molecular structure indicates that the new molecule exhibits higher reactivity, in agreement with the design objective.

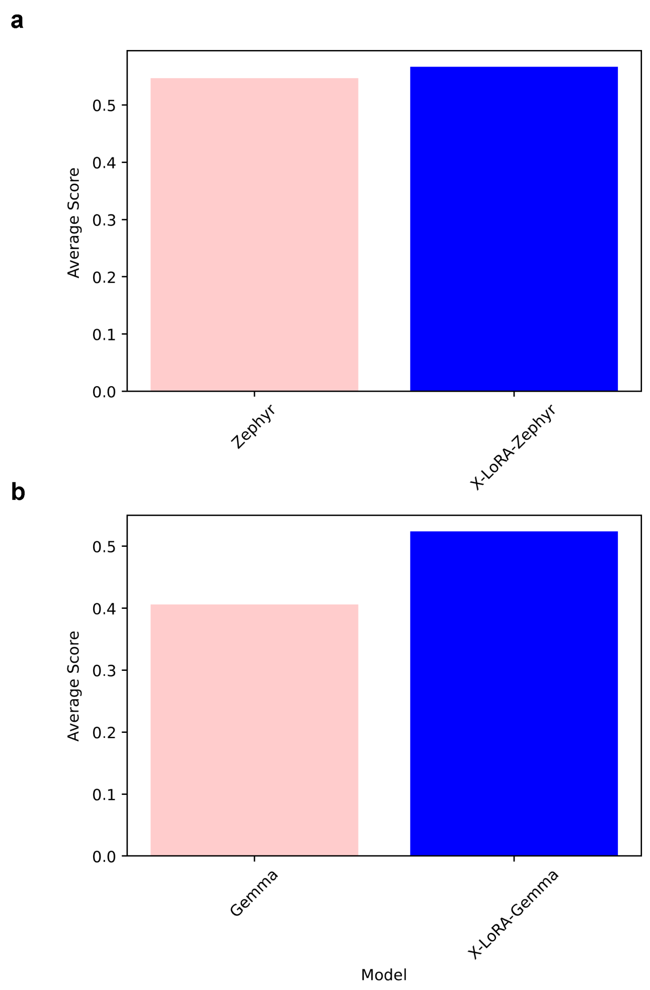


Figure 18: Performance assessment using the chemistry, physics, and biology questions in the MMLU benchmark (5-shot predictions) [45], complementing the domain-specific benchmarks shown in Figures 5, 15, and 15. Panel (a) shows the results for the X-LoRA-Zephyr model (referred to as X-LoRA in the other parts of the paper), and panel (b) shows results for the X-LoRA-Gemma model. The X-LoRA models outperform both base models (Zephyr-based model: 54.6% for the base model versus 56.6% for the X-LoRA model, and Gemma-based model: 40.6% for the base model and 52.4% for the X-LoRA-Gemma model).

2.5.1 Additional benchmarking

We report additional benchmarks of the X-LoRA-Gemma model, shown in Figure 18, using a subset of the MMLU benchmark focused on chemistry, physics and biology (these are relatively close to the domains our model has been trained on) [45]. We find both X-LoRA models discussed in this paper outperform the corresponding base models. The X-LoRA models outperform both base models (Zephyr model: 54.6% for the base model versus 56.6% for the X-LoRA model, Gemma model: 40.6% for the base model and 52.4% for the X-LoRA-Gemma model). The best overall performance is observed for the X-LoRA-Zephyr model, but both are reasonably close.

3 Conclusion

There are many current and potential future applications of multimodal LLMs in science, along with a great need for special-purpose highly specialized expertise. In addition to general language tasks, the access of calculation or generative tasks like demonstrated here with protein mechanics is critical [11, 8, 13]. X-LoRA, built on open-source language models, enables us to dynamically mix expertise and knowledge across domains, as demonstrated in various examples, including Figure 10. This analysis showed how the X-LoRA model dynamically and autonomously evolves the use of adapters, and how it mixes certain adapters in a complex fashion across the LoRA layers. The complex mixing of LoRA layers is consistently seen across all examples studied here, such as in Figures 2 and 3. This poses also an interesting problem to better understand the underlying mechanisms from a physics perspective, where viewing a LLM model as a large-scale graph forming model that exploits intricate graph structures. Thereby, the mixing of adaptation experts at deep layer scales as done here alters these mechanisms in ways to develop suitable solutions.

The proposed algorithm augments the conventional inference strategy of a single forward pass to feature two forward passes. This trains the model to first 'think' about the question and how it may reconfigure itself before responding. This implements a simple realization of 'self-awareness' whereby the model is able to adapt its own structure in order to best solve a task. As a result, the model, even though it has a relatively small parameter count of 7 billion parameters, can reason across diverse scientific domains (biological materials, math, physics, chemistry, logic, mechanics, etc.), significantly enhancing its capacity for generating innovative solutions. It may also guide the development of alternative model architectures, for instance, methods that generalize the reconfigurational strategy to parts of the model, or the entire model, of non-adapted systems. In such a scenario the scaling head would dynamically reconfigure segments of a trained model, or merge multiple models. As we have seen in the case of X-LoRA, such strategies can be powerful mechanisms to expand or integrate capabilities of different models.

We provided several performance assessments, including a comparison with much larger models (Figure 5), where X-LoRA clearly outperforms. The detailed comparison shown in Table 2.1 showed that X-LoRA performs markedly better across a variety of tasks and application domains than the base model. Other comparisons include quantitative assessments of numerical prediction tasks, such as protein mechanics and quantum-mechanical properties of molecules, where X-LoRA shows high ac curacies with R_2 between 0.85 and 0.96. For comparison, these were higher than other published results, such as predicting protein unfolding force using the MeLM model [13] where $R_2 = 0.78$ was reported. The capability to accurate solve multiple prediction tasks, along with inverse design tasks, and sophisticated reasoning capabilities were underscored in the example focused on molecular design (Section 2.5). This example also illustrated that X-LoRA can be successfully implemented based on different base model architectures (Gemma vs. Zephyr/Mistral). Future work could expand this further and develop X-LoRA models for larger base models.

A disadvantage of the proposed approach is that we require two forward passes, which can lead to additional computational cost: One to calculate hidden states that serve as the input to the X-LoRA scaling head, and one with Λ applied to the adapters. However, the first pass is used for the X-LoRA scaling head and as such exploits basic capabilities of the LLM in contrast with learning the knowledge separately by training a larger scaling head. To implement effective application towards model-serving tools, separate key-value caches should be tracked for both the scaling and forward passes. To improve inference speed, we developed `Mistral.rs`, a LLM serving platform that implements X-LoRA in Rust [46] and includes several methods to optimize performance. Nevertheless, the advantage of our approach is that it works for any model without a need to restructure the internal logic. We view this as an advantage, especially as it can be applied to the vast resources available in the Hugging Face ecosystem.

One advantage of the proposed approach is that it provides a simple way to implement in any existing LLM (e.g., since the X-LoRA code has been released, it should be easily usable for any model in the Hugging Face ecosystem). Another advantage is that the scaling head exploits the innate strength of the LLM used, and learns sophisticated ways by which the adapters are best mixed for certain answers. During training of X-LoRA, it would be best to provide complex samples of question-answer or conversations that allow the model to learn the best combinatorial method. We examined several analyses of protein design and mechanics. For instance, Figure 12 and Fig. 13 presented a stability analysis of several protein sequences, including a comparison, reasoning over the predicted results, and a mechanistic explanation that provides a bottom-up chemical foundation of the predicted behavior. As can be see in Fig. 13 this was directly confirmed via a structural analysis of the protein's folded 3D geometry.

As for other future work, additional research could explore a greater variety of adapter experts, building on the initial set of expertise in our LoRA set. The X-LoRA model exceeds the capabilities of the base model or a base model with either one adapter, and it can provide answers to complex questions while drawing on specialized skills endowed from the set of adapters. There also seem to be synergies between the various adapters, as the complex mixing results of scaling weights across the layers seems to indicate. This could be researched further, and compared with methods like SLERP that merge models with different strategies. Whereas we trained the X-LoRA scaling head with a subset of the training data, using a few hundred samples from each of the original training sets used to develop the adapters, more purpose-driven development of a training set for such a purpose can be devised. For instance, we noticed that posing multiple-choice questions tends to activate the reasoning-focused adapters since these were trained on such datasets. Additional prompting is required to help the scaling head understand a certain domain that is necessary for questions asked, for instance via the use of particular system messages or other strategies. Generally, the development of adequate training sets is an area that deserves further exploration including specific methods to invoke effecting mixing of layer-wise scaling. Our experiments of tracking these over longer conversations show that the model can clearly invoke changing scaling mechanisms to best respond to certain tasks, which is a promising feat.

We observed fascinating patterns by which experts are activated across the layers, with interesting insights discussed in the paper. This analysis can be expanded, and may yield important insights about how and why mixing model parts or sets of layers can be advantageous. This would be particularly interesting when the X-LoRA method is applied to areas other than protein mechanics and protein design as done here. We leave this to future investigations.

Another important avenue is the further study of agentic modeling, as done here using two X-LoRA models that interact adverserially. Future work could use more than two models to enhance interactions and add further capabilities to push models towards unexplored generative realms. This can also help with the development of methods that integrate physics-based modeling or other validation steps, using code-writing/executing agents or agents that use function calling or other processing techniques [47].

4 Materials and Methods

X-LoRA codes and examples are available at `https://github.com/EricLBuehler/xlora`. Model weights and data associated with the X-LoRA discussed in this paper, along with additional examples, are available at `https://huggingface.co/lamm-mit/x-lora`. The X-LoRA-Gemma model is available at `https://huggingface.co/lamm-mit/x-lora-gemma-7b`. The `Mistral.rs` inference engine is available at `https://github.com/EricLBuehler/mistral.rs`.

4.1 Mathematical formulation of X-LoRA

We define the **scalings** to be a matrix containing token and layer level coefficients for LoRA adapters. The X-LoRA **scaling head** takes the hidden states from the base model and applies linear fully-connected layers to predict scalings, after which a softmax function is applied to the final dimension to ensure scalings across all experts add up to one. We use ReLU activation functions and dropout layers intercalated between linear fully-connected layers in the X-LoRA scaling head. The side of the hidden layers can be set, and chosen to be succesfuly reduce the size from the hidden dimension to the scaling dimension, or implement a widening and narrowing approach similar to the feed-forward layer in the transformer block.

The scalings are a matrix $\Lambda \in \mathbb{R}^{s\times l\times n}$. Therein, s, l, and n, respectively, denote the sequence length, the number of layers, and the number of adapters. For each layer, the corresponding scalings are extracted: A matrix $\lambda \in \mathbb{R}^{s\times n}$. The scalings λ are then applied to each LoRA adapter by broadcast multiplying the corresponding X-LoRA scaling $\lambda_i \in \mathbb{R}^{1\times 1\times s}$ to the input for the decomposition matrices A and B.

With $W_0 \in \mathbb{R}^{d\times k}$ and $B_i \in \mathbb{R}^{d\times r_i}$ and $A_i \in \mathbb{R}^{r_i\times d}$ with the rank $r_i \ll \min(d, k)$ that are unique to each adapter i, the forward pass of an X-LoRA layer is expressed as the following equation:

$$h = W_0 x + \sum_{i=1}^{n} B_i A_i (x \cdot \lambda_i)\alpha_i \tag{5}$$

Visualizations of the scalings are depicted in the main paper. These were either visualized using bar plots or using the `contourf` function in Matplotlib [48].

4.2 X-LoRA implementation algorithm and code

The implementation chosen here is driven by ease-of-use and the ability to easily adapt the approach to a variety of models, specifically those available in the Hugging Face ecosystem. The X-LoRA algorithm operates by:

1. Freezing the base model and adapters, and then tuning performance.
2. Mixing LoRA adapters according to the predictions of the X-LoRA scaling head.

All code is written in Python and PyTorch [49].

4.2.1 Dual forward pass strategy for self-aware inference

Because X-LoRA requires the calculation of the hidden states to predict the scalings, we implement a dual forward pass approach that utilizes a first pass for the model to consider the task, to decide how to reconfigure itself, followed by the inference step. This implements a type of ‘self-awareness’ that endows X-LoRA with the capability to use additional

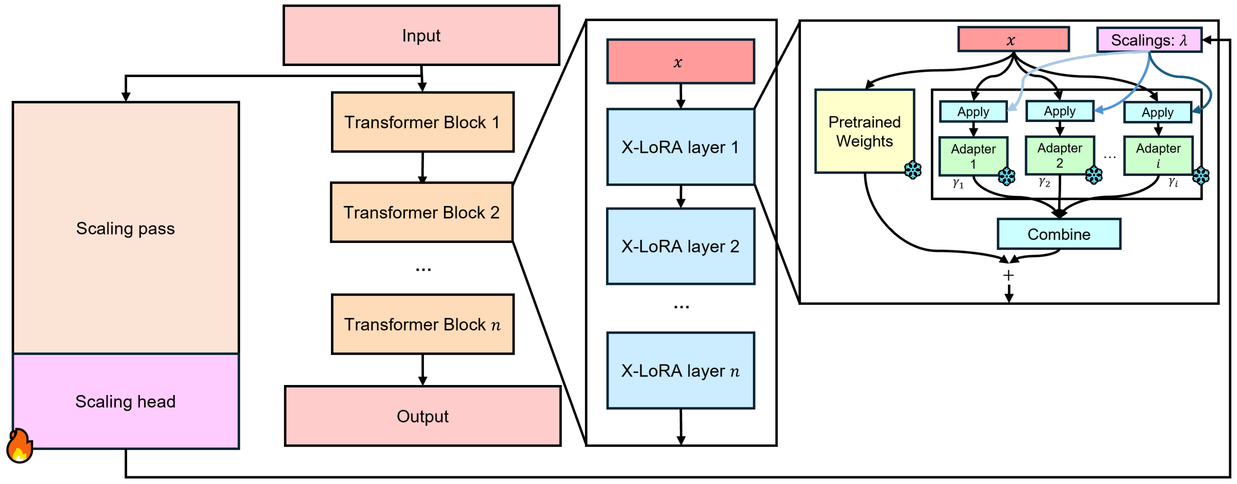


Figure 19: Detailed overview of the X-LoRA architecture, including the method of scaling inputs. As shown, the scalings are applied to the inputs of each frozen adapter. These are then additively combined. Finally, the output of the pretrained weights are added to form the final output of the X-LoRA layer.

compute for reflection rather than immediate response. We name the first pass the scaling pass and the second pass as the forward pass. We set the scalings Λ during the scaling pass to zero. We also experimented with setting the scalings Λ to n^{-1} and find similar results, indicating stable performance (in the X-LoRA package, this parameter can be set in the configuration). To easily be able to apply X-LoRA to a variety of models, we implement a forward pass hook. This strategy plays an essential role in the relationship between the forward and scaling passes and is the key feature which allows application to all models.

1. **Scaling pass**: Implemented by executing the base model with adapters set to constant values. The resulting hidden states are used by the X-LoRA scaling head to generate Λ.
2. **Forward pass**: Apply Λ to mix the adapters during the forward pass of the model. The resulting logits are returned as the output of the X-LoRA model.

As a note of caution, using the same key-value cache for both the scalings pass and forward pass can potentially interfere with proper inputs to the scaling head. This is because the key-value cache will persist and would accumulate across the two forward passes. As such, we do not use the key-value cache during training or inference and implemented a proper handling of this in `Mistral.rs`.

4.2.2 Freezing weights and opening segments of the model for training

The base model and adapters are frozen (*i.e.*, their weights are not being trained) and as such the X-LoRA scaling head is the only trainable part of an X-LoRA model. It remains an option, in the code presented, to open up all adapters together with the X-LoRA scaling head. One could even opt to train the entire model - X-LoRA scaling head, adapter and the base foundation model jointly (or with different learning rates). Alternatively, future iterations of the concepts proposed here could be used to develop scaling functions between two or more foundation models to implement a traceable manifestation of SLERP-type modeling. This can be explored in future work.

4.2.3 X-LoRA layers

An X-LoRA layer is a layer which scales and ultimately combines the outputs of several adapters. Each X-LoRA layer wraps the original LoRA layer and therefore has access to the specific adapters for that layer.

To effectively apply X-LoRA to any model, the X-LoRA algorithm swaps the original LoRA layers by creating X-LoRA layers and then injects their forward mechanics into the LoRA layer. This leverages aspects of Python's object model to enable swapping.

In summary, each X-LoRA layer executes the following steps during the forward pass (see Figure 19):

1. **Scaling**: Multiply the input signal of each LoRA adapter by the scaling value λ_i.
2. **Forward**: Run the forward pass for the wrapped LoRA layer.

Table 3: Summary of datasets used to train the individual adapters of the X-LoRA models reported here, along with references and sources. The X-LoRA model (also referred to as X-LoRA-Zephyr) features 9 adapters, whereas the X-LoRA-Gemma model 4 features adapters.

Adapter	**Dataset/source used, reference**	**Used in X-LoRA (Zephyr based)**	**Used in X-LoRA-Gemma**
Bioinspired materials	Dataset based on [11] expanded with new distilled data	✓	✓
Chain-of-thought (CoT) and reasoning	Dataset based on the FLAN dataset[52], as curated at [53] (combination of 9 CoT datasets)	✓	
Chemistry	Based on the CAMEL: Communicative Agents for "Mind" Exploration of Large Scale Language Model Society [54] dataset for chemistry, CAMEL-AI Chemistry[55]	✓	
Mathematics	CAMEL-AI Mathematics [56]	✓	
Physics	CAMEL-AI Physics [57]	✓	
Biology	CAMEL-AI Biology [58]	✓	
Mechanics and materials	Training data used in the MechGPT model as presented in [8], and available at [59]	✓	✓
Platypus: Logic and reasoning	Platypus[60] dataset as reported at [61]	✓	
Protein mechanics tasks (generative sequence-to-property and inverse capabilities focused on unfolding/physical properties)	Dataset based on molecular modeling results as reported in [29, 47]; forces and energies are normalized using the same factors as in the earlier reference that reported the original dataset.	✓	✓
Quantum Machine 9 (QM9)	Dataset featuring a variety of computed properties for 134,000 molecules (expressed in SMILES code [62, 63]), based on quantum mechanical calculations. These properties are essential for understanding the behavior and characteristics of molecules in various contexts [38, 39].		✓

4.3 Efficient Inference with `Mistral.rs` implemented in Rust

To improve inference speed, we developed `Mistral.rs` in Rust, a LLM serving platform that implements X-LoRA in Rust [46] along with several optimizations to improve performance. Specifically, the following methods are implemented, listed below in no particular order:

- LoRA adapter weight stacking

  If all A_i and B_i matrices have the same shape for each LoRA adapter, we can stack the LoRA adapter weight matrices A_i and B_i along the first dimension.

  Mathematically, given $W_0 \in \mathbb{R}^{d\times k}$, for adapter matrices $A_i \in \mathbb{R}^{r\times d}$ and $B_i \in \mathbb{R}^{d\times r}$ with $r_i \ll \min(d,k)$, we concatenate the matrices so that $A \in \mathbb{R}^{i\times r\times d}$ and $B \in \mathbb{R}^{i\times d\times r}$. The α_i parameters are also applied during initialization to the stacked matrices.

  To apply scalings, we permute the dimensions such that $\lambda \in \mathbb{R}^{i\times 1\times 1}$. Then, they are applied through a broadcast multiplication.

- Non-granular scalings

  To reduce the amount of times that the scaling pass is run, we implement an option of non-granular scalings. This caches the scalings after n completion runs, as the KV cache ensures that the sequence length will remain at one for the remainder of the request. By using this approach, it decreases latency significantly as it reduces the number of invocations of the base model.

- Fused Compute Unified Device Architecture (CUDA) kernels

  To optimize the forward pass, we implement fused CUDA kernels. Fused kernels enable complex operations, such as Rotary Embedding ([50]) which would otherwise be executed sequentially, to be combined into a single kernel than can be executed in parallel on a GPU. We base our kernels on the vLLM implementation ([51]).

- Quantization

  The inference engine implements quantization, which allows one to use a quantized base model. In initial experiments, we find that quantization up to 8 bits works well even with a model that was trained with `bfloat16`, for instance.

4.4 Datasets

Each of the adapters is trained with special-purpose datasets, as summarized in Table 3 along with references to sources and original literature.

4.5 Training strategy

Training is conducted in stages. We use the Zephyr-7B-β model [25] that was built on top of the Mistral-7B model[5] and train a series of adapters using the datasets described in the previous section. We use the Zephyr-7B-β tokenizer.

The first eight adapters are trained using question-answer pairs as provided by the datasets (see, Table 3). The protein mechanics tasks are trained similarly, but using specific forward and inverse instruction sets. The bidirectional instruction tasks are:

```
CalculateForce< G E E C D C G S P ....> [0.310]
CalculateEnergy< G E E C D C G S P ....> [0.220]
CalculateForceHistory< G E E C D C G S P ....> [0.004,0.034,0.125,0.142,0.159,0.102,0.079,0.073,0.131, ...]
GenerateForce<0.310>[ G E E C D C G S P ....]
GenerateEnergy<0.220>[ G E E C D C G S P ....]
GenerateForceHistory<0.004,0.034,0.125,0.142,0.159,0.102,0.079,0.073,0.131, ...>[ G E E C D C G S P ....]
```

These represent a total of three tasks (calculate the maximum unfolding force of a protein, the unfolding energy, and the force-deformation history). Each of the three forward tasks is complemented by three inverse tasks of the same nature, but yielding a protein sequence as a response (hence, a total of six tasks).

The rank of each LoRA adapter is $r = 16$ with $\alpha = 16$. Each adapter is trained with the following five target modules: `v_proj`, `k_proj`, `q_proj`, `gate_proj`, and `o_proj`, with dropout 0.05. A LoRA adapter is usually trained with low ranks, such as 4 to 32. This is because it has been shown that increasing the rank of LoRA adapters does not typically improve performance, but it does increase computational requirements since a smaller rank results in a more compact model with fewer parameters [15]. The choices for r, α, and other parameters are made based on common values used in the literature that have been shown to work well. Moreover, earlier work by the author of this paper has experimented with various choices and found the values used here to be a reasonable choice [11].

The X-LoRA scaling head is trained with around 2,000 samples drawn as a fraction from the original dataset used to train the adapters (we also complemented the training data with a additional GPT-4 generated samples of multiple choice questions). We use a `paged_adamw_8bit` optimizer with gradient clipping of 0.3, a learning rate 2×10^{-4} with warmup, and four gradient accumulation steps, with a batch size of one. We train the X-LoRA model for around 10,000 steps. The loss is around 0.28 when we conclude training.

Since the basic Mistral architecture used here as the foundation model has 32 layers, and since we use adapters in 5 of each of the transformer layers, there is a total of 160 LoRA layers, each of which is supplanted by an X-LoRA layer. The X-LoRA scaling head predicts scaling values for each of these 160 LoRA layers (for nine LoRA experts, this results in a total of $160 \times 9 = 1440$ values of λ_i computed for each token). We use single linear layer with ReLU activation functions and a dropout with $p = 0.2$, with around 5M parameters. The code implements flexible structures so that more complex deep classifier heads can be constructed and explored in other applications.

Both LoRA adapters and the X-LoRA scaling head are trained using next-token prediction and cross-entropy loss (we calculate the probability that the model assigns the correct next token in a sequence given the previous tokens; where the goal during training is to maximize the probability of the correct next token as denoted in the training set). We use token shifting, where the sequence is shifted by one token to create the target sequence that the model should predict. For example, if the input sequence is “Proteins are fundamental to", the target sequence for training would be “are fundamental to biology". Therefore, the model tries to predict the next token in the sequence. As shown in Figure 1 during X-LoRA scaling head training, only the scaling head is trainable and all other parameters are frozen. The native Zephyr tokenizer and prompt template is used.

4.6 X-LoRA-Gemma model

The X-LoRA-Gemma model is developed in a similar way as the X-LoRA model discussed above, but based on the Gemma-7B-it model [37], and using four adapters (see main text for details). The four adapters are trained on the mechanics and materials, protein mechanics, bioinspired materials, as well as an additional quantum-mechanics based molecular properties QM9 (see Table 2) dataset [38, 39]. The rank of each LoRA adapter is $r = 16$ with $\alpha = 16$. Each adapter is trained with the following seven target modules: `q_proj`, `o_proj`, `k_proj`, `v_proj` `gate_proj`, and `up_proj` and `down_proj`, with dropout 0.05. The rank of each LoRA adapter is chosen to be $r = 16$ with $\alpha = 16$. With four adapters and 7 target models, since there is a total of 28 layers in the Gemma-7B base model, there is a total of $28 \times 28 = 784$ values of λ_i computed for each token.

The first two adapters (bioinspired and mechanics of materials) used in the construction of this model are trained using question-answer pairs as provided by the datasets (see, Table 3). The protein mechanics tasks are trained in the same way as described in the preceding section, with bidirectional forward and inverse instruction sets. In addition, we train an adapter based on the QM9 dataset, with these two tasks:

```
CalculateMolecularProperties<CC(C)N1CC1C=O> [0.098,0.358,0.581,0.395,0.309,0.330,0.570,0.649,0.519,0.519,0.519,0.518]
GenerateMolecularProperties<0.098,0.358,0.581,0.395,0.309,0.330,0.570,0.649,0.519,0.519,0.519,0.518> [CC(C)N1CC1C=O]
...
```

These represent a total of two tasks (calculate molecular properties, the forward task, and the inverse task to generate a molecule with target properties). All 12 properties featured in the QM9 dataset are predicted or designed for, respectively (see, Table 2). This model features a total of four adapters.

The X-LoRA scaling head for this model has three layers, whereas the intermediate layer has a size of $3072 \times FF_{\text{fac}} = 12,288$, where $FF_{\text{fac}} = 4$ is a feed-forward multiplier. We use linear layers with ReLU activation functions and a dropout of $p = 0.2$, with around 47M parameters. The X-LoRA-Gemma scaling head is trained with around 17,000 samples. These were drawn as a fraction from the training dataset used to train adapters; we note that even though the X-LoRA-Gemma model only contained four adapters, we used samples from across the entire training set of the model described above with the addition of samples from the QM9 dataset. We did this to experiment with the possibility that the X-LoRA model may be able to learn additional capabilities at the scaling head training stage. Since the loss shows good convergence to 0.58 at the end of training, together with excellent performance even in tasks for which the model was not explicitly trained for (e.g. chemistry, as shown in the molecule design example discussed in the paper), we believe that this can generally yield good results. We used a `paged_adamw_8bit` optimizer with gradient clipping of 0.3, a learning rate 1×10^{-4} with warmup, and four gradient accumulation steps, with a batch size of one. We train the X-LoRA-Gemma model for around 62,000 steps. The native Gemma tokenizer and prompt template is used.

4.7 Adversarial agentic modeling

We implement an adversarial agentic strategy by instantiating two X-LoRA agents. One X-LoRA agent focuses on question asking. It asks the initial question and subsequently keeps developing questions to identify challenges and weaknesses in the responses. The other agent responds to the queries and continually provides answers. We implement this strategy in {Guidance} [64].

For the example relating music and mechanics provided in the text, the features of the two agents are described as follows. First, the physicist and question asker:

```
You are a critical physicist.  You are taking part in a discussion, from the
perspective of physics.  Keep your answers brief, and always challenge statements
in a provocative way.

As a creative individual, you inject ideas from other fields and push the
boundaries.
```

Second, a philosopher (in some of the examples this role is revised to reflect a biology expert):

```
You are a philosopher with knowledge in biology, chemistry and mathematics.  You are
taking part in a discussion.

Keep your answers brief, but accurate, and creative.  You come up with excellent
ideas and new directions of thought, always logical.
```

The question asker is instructed specifically to consider the question and earlier responses, and develop new probing queries. This is realized via a prompt as such:

```
Consider this question and response.

### Question:  <question>

### Response:  <response>

### Instruction:  Respond with a SINGLE follow-up question that critically
challenges the response.

DO NOT answer the question or comment on it yet.

The single question is:
```

Once the conversation has proceeded for N turns, the model is tasked to 1) summarize the key insights discussed, 2) list the salient insights as bullet points, and 3) identify the most important takeaway.

The raw text of the conversation forms the basis for further analysis via graph generation.

4.8 Knowledge graph generation

We use Zephyr-7B-β to extract triplets from text, following the strategy reported in [65] with additional features based on the Llama Index graph generation algorithm [66]. Graphs are visualized using NetworX[67] and Pyvis [68]. One of the key instructions given is, following the approach suggested in [66]:

```
Your task is to extract the ontology of terms mentioned in the given context.  These
terms should represent the key concepts as per the context.

Format your output as a list of JSON. Each element of the list contains a pair of
terms and the relation between them, like the following:  [...]
```

We instruct to build triplets of nodes and edge features as follows, following the approach used in[66]:

- "node_1": "A concept from extracted ontology"
- "node_2": "A related concept from extracted ontology"
- "edge": "Relationship between the two concepts, node_1 and node_2, succinctly described"

We also provide example of ontological analysis to the model, such as:

```
Context: ‘‘‘Spider silk is a strong natural fiber used to catch prey in a web.‘‘‘
```

We then give example triplets, such as

- "node_1": "spider silk"
- "node_2": "fiber"
- "edge": "is"

The resulting graph data is assembled and then grouped in to communities using the Girvan-Newman algorithm [36].

4.9 Visualization of molecular structures

We use PyMOL [69] to visualize and analyze the predicted protein structures. Various scripts and functions within this software are used to identify certain features of the proteins, such a secondary structure, hydrophobic/hydrophilic regions, disulfide bonds, and hydrogen bonds.

Data Availability Statement

The codes and data that support the findings of this study are openly available in `https://github.com/EricLBuehler/xlora` and `https://huggingface.co/lamm-mit/x-lora`. Additional data sources used for this study are referenced (see, specifically Table 3).

Acknowledgments

This was supported in part by MIT's Generative AI Initiative and Google.

Author Contributions

ELB and MJB conceived the concept, plan of study, developed the model and research, and wrote the paper. ELB developed the algorithms, codes and GitHub repository. MJB conducted the scientific investigations, carried out the protein modeling and analysis, molecular design and analysis, and revised and finalized the paper.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, *Attention is All you Need* (2017), URL `https://papers.nips.cc/paper/7181-attention-is-all-you-need`.

[2] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. (2023), URL `https://arxiv.org/abs/2307.09288v2`.

[3] OpenAI (2023), URL `http://arxiv.org/abs/2303.08774`.

[4] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al. (2022), URL `https://arxiv.org/abs/2204.02311v3`.

[5] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, et al. (2023), URL `https://arxiv.org/abs/2310.06825v1`.

[6] S. Gunasekar, Y. Zhang, J. Aneja, C. César, T. Mendes, A. D. Giorno, S. Gopi, M. Javaheripi, P. Kauffmann, G. De, et al. (2023), URL `https://arxiv.org/abs/2306.11644v2`.

[7] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg, et al. (2023), URL `https://arxiv.org/abs/2303.12712v1`.

[8] M. J. Buehler, Appl. Mech. Rev. (2023), URL `https://doi.org/10.1115/1.4063843`.

[9] M. Nejjar, . Luca, Z. †1, F. Stiehle, and I. Weber (2023), URL `https://arxiv.org/abs/2311.16733v3`.

[10] M. J. Buehler, ACS Engineering Au (2023), URL `10.1021/acsengineeringau.3c00058`.

[11] R. K. Luu and M. J. Buehler, Adv. Science. (2023), URL `https://doi.org/10.1002/advs.202306724`.

[12] R. K. Luu, M. Wysokowski, and M. J. Buehler, Applied Physics Letters **122** (2023), URL `http://arxiv.org/abs/2304.12400http://dx.doi.org/10.1063/5.0155890`.

[13] M. J. Buehler (2023), URL `https://arxiv.org/abs/2306.17525v1`.

[14] Y. Ge, W. Hua, K. Mei, J. Ji, J. Tan, S. Xu, Z. Li, and Y. Zhang (2023), URL `http://arxiv.org/abs/2304.04370`.

[15] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, *Lora: Low-rank adaptation of large language models* (2021), `2106.09685`.

[16] D. Kim, C. Park, S. Kim, W. Lee, W. Song, Y. Kim, H. Kim, Y. Kim, H. Lee, J. Kim, et al. (2023), URL `https://arxiv.org/abs/2312.15166v1`.

[17] *arcee-ai/mergekit: Tools for merging pretrained large language models.*, URL `https://github.com/arcee-ai/mergekit`.

[18] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, Neural Computation **3**, 79 (1991), ISSN 0899-7667, URL `https://dx.doi.org/10.1162/neco.1991.3.1.79`.

[19] J. B. Hampshire and A. Waibel, IEEE Transactions on Pattern Analysis and Machine Intelligence **14**, 751 (1992), ISSN 01628828.

[20] M. I. Jordan and R. A. Jacobs, Neural Computation **6**, 181 (1994), ISSN 0899-7667, URL `https://dx.doi.org/10.1162/neco.1994.6.2.181`.

[21] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. d. l. Casas, E. B. Hanna, F. Bressand, et al. (2024), URL `https://arxiv.org/abs/2401.04088v1`.

[22] Y. Hu and M. J. Buehler, APL Machine Learning **1**, 010901 (2023), ISSN 2770-9019, URL `https://aip.scitation.org/doi/abs/10.1063/5.0134317`.

[23] M. J. Buehler, Modelling and Simulation in Materials Science and Engineering (2023), ISSN 0965-0393, URL `https://iopscience.iop.org/article/10.1088/1361-651X/accfb5https://iopscience.iop.org/article/10.1088/1361-651X/accfb5/meta`.

[24] M. J. Buehler and K. Guo (2022), URL `https://zenodo.org/record/6346661`.

[25] *HuggingFaceH4/zephyr-7b-beta · Hugging Face*, URL `https://huggingface.co/HuggingFaceH4/zephyr-7b-beta`.

[26] M. Buehler, *Atomistic modeling of materials failure* (2008), ISBN 9780387764252.

[27] *DALL·E 3 system card*, URL `https://openai.com/research/dall-e-3-system-card`.

[28] W. Lu, D. L. Kaplan, and M. J. Buehler, Advanced Functional Materials p. 2311324 (2023), ISSN 1616-3028, URL `https://onlinelibrary.wiley.com/doi/full/10.1002/adfm.202311324https://onlinelibrary.wiley.com/doi/abs/10.1002/adfm.202311324https://onlinelibrary.wiley.com/doi/10.1002/adfm.202311324`.

[29] B. Ni, D. L. Kaplan, and M. J. Buehler (2023), URL `https://arxiv.org/abs/2310.10605v3`.

[30] H. Lu, B. Isralewitz, A. Krammer, V. Vogel, and K. Schulten, Biophysical Journal **75**, 662 (1998), ISSN 00063495, URL `/pmc/articles/PMC1299741/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC1299741/`.

[31] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, et al., Nature **596**, 583 (2021), ISSN 14764687.

[32] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, Journal of Molecular Biology **215**, 403 (1990), ISSN 00222836, URL `http://www.ncbi.nlm.nih.gov/pubmed/2231712https://linkinghub.elsevier.com/retrieve/pii/S0022283605803602`.

[33] M. Mirdita, K. Schütze, Y. Moriwaki, L. Heo, S. Ovchinnikov, and M. Steinegger, Nature Methods 2022 19:6 **19**, 679 (2022), ISSN 1548-7105, URL `https://www.nature.com/articles/s41592-022-01488-1`.

[34] T. Giesa, D. Spivak, and M. Buehler, Advanced Engineering Materials **14** (2012), ISSN 14381656 15272648.

[35] D. I. Spivak and M. J. B. Reoccurring, pp. 0–13 (2011).

[36] M. Girvan and M. E. Newman, Proceedings of the National Academy of Sciences of the United States of America **99**, 7821 (2002), ISSN 00278424, URL `https://www.pnas.org/doi/abs/10.1073/pnas.122653799`.

[37] *google/gemma-7b-it · Hugging Face*, URL `https://huggingface.co/google/gemma-7b-it`.

[38] L. Ruddigkeit, R. Van Deursen, L. C. Blum, and J. L. Reymond, Journal of Chemical Information and Modeling **52**, 2864 (2012), ISSN 1549960X, URL `https://pubs.acs.org/doi/full/10.1021/ci300415d`.

[39] R. Ramakrishnan, M. Hartmann, E. Tapavicza, and O. A. Von Lilienfeld, Journal of Chemical Physics **143**, 84111 (2015), ISSN 00219606, URL `/aip/jcp/article/143/8/084111/73278/Electronic-spectra-from-TDDFT-and-machine-learning`.

[40] A. K. Rappé, C. J. Casewit, K. S. Colwell, W. A. Goddard, and W. M. Skiff, Journal of the American Chemical Society **114**, 10024 (1992), ISSN 15205126, URL `https://pubs.acs.org/doi/pdf/10.1021/ja00051a040`.

[41] J. Gasteiger and M. Marsili, Tetrahedron **36**, 3219 (1980), ISSN 0040-4020.

[42] D. A. McQuarrie and J. D. Simon, *Physical Chemistry: A Molecular Approach* (University Science Books, 1997), ISBN 978-0935702996.

[43] F. A. Carey and R. J. Sundberg, *Advanced Organic Chemistry: Part A: Structure and Mechanisms* (Springer, 2007), 5th ed., ISBN 978-0387683461.

[44] M. Varghese and M. W. Grinstaff, Chemical Society Reviews **51**, 8258 (2022), ISSN 1460-4744, URL `https://pubs.rsc.org/en/content/articlehtml/2022/cs/d1cs00930chttps://pubs.rsc.org/en/content/articlelanding/2022/cs/d1cs00930c`.

[45] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt, ICLR 2021 - 9th International Conference on Learning Representations (2020), URL `https://arxiv.org/abs/2009.03300v3`.

[46] N. D. Matsakis and F. S. Klock II, in *ACM SIGAda Ada Letters* (ACM, 2014), vol. 34, pp. 103–104.

[47] A. Ghafarollahi and M. J. Buehler (2024), URL `https://arxiv.org/abs/2402.04268v1`.

[48] *Matplotlib documentation — Matplotlib 3.5.1 documentation*, URL `https://matplotlib.org/stable/index.html`.

[49] A. Paszke, S. Gross, J. Bradbury, Z. Lin, Z. Devito, F. Massa, B. Steiner, T. Killeen, and E. Yang (2019).

[50] J. Su, Y. Lu, S. Pan, B. Wen, and Y. Liu, *Roformer: Enhanced transformer with rotary position embedding* (2021), `2104.09864`.

[51] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, and I. Stoica, in *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles* (2023).

[52] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, ICLR 2022 - 10th International Conference on Learning Representations (2021), URL `https://arxiv.org/abs/2109.01652v5`.

[53] *QingyiSi/Alpaca-CoT at main*, URL https://huggingface.co/datasets/QingyiSi/Alpaca-CoT/tree/main/Chain-of-Thought.

[54] G. Li, . Hasan, A. Al, K. Hammoud, H. Itani, D. Khizbullin, and B. Ghanem (2023), URL https://arxiv.org/abs/2303.17760v2.

[55] *camel-ai/chemistry · Datasets at Hugging Face*, URL https://huggingface.co/datasets/camel-ai/chemistry.

[56] *camel-ai/math · Datasets at Hugging Face*, URL https://huggingface.co/datasets/camel-ai/math.

[57] *camel-ai/physics · Datasets at Hugging Face*, URL https://huggingface.co/datasets/camel-ai/physics.

[58] *camel-ai/biology · Datasets at Hugging Face*, URL https://huggingface.co/datasets/camel-ai/biology.

[59] *lamm-mit/MechanicsMaterials · Datasets at Hugging Face*, URL https://huggingface.co/datasets/lamm-mit/MechanicsMaterials.

[60] A. N. Lee, C. J. Hunter, and N. Ruiz (2023), URL https://arxiv.org/abs/2308.07317v1.

[61] *garage-bAInd/Open-Platypus · Datasets at Hugging Face*, URL https://huggingface.co/datasets/garage-bAInd/Open-Platypus.

[62] D. Weininger, Journal of Chemical Information and Computer Sciences **28**, 31 (1988), ISSN 00952338, URL https://pubs.acs.org/doi/abs/10.1021/ci00057a005.

[63] D. Weininger, A. Weininger, and J. L. Weininger, Journal of Chemical Information and Computer Sciences **29**, 97 (1989), ISSN 00952338, URL https://pubs.acs.org/doi/abs/10.1021/ci00062a008.

[64] *guidance-ai/guidance: A guidance language for controlling large language models.*, URL https://github.com/guidance-ai/guidance.

[65] *rahulnyk/knowledge_graph: Convert any text to a graph of knowledge. This can be used for Graph Augmented Generation or Knowledge Graph based QnA*, URL https://github.com/rahulnyk/knowledge_graph.

[66] J. Liu, *LlamaIndex* (2022), URL https://github.com/jerryjliu/llama_index.

[67] *networkx/networkx: Network Analysis in Python*, URL https://github.com/networkx/networkx.

[68] *WestHealth/pyvis: Python package for creating and visualizing interactive network graphs.*, URL https://github.com/WestHealth/pyvis/tree/master.

[69] L. L. C. Schrödinger, *The PyMOL molecular graphics system, version 1.8* (2015).