# A Transformer Based Machine Learning of Molecular Grammar Inherent in Proteins Prone to Liquid Liquid Phase Separation

Abdul Wasim* and Jagannath Mondal†

*Tata Institute of Fundamental Research Hyderabad, Telangana 500046, India*

In this study, we delve into the intricate molecular language of Intrinsically Disordered Proteins (IDPs) using specialized transformer neural network-based language models, specifically GPT models, pre-trained on sequences exhibiting varying propensities for liquid-liquid phase separation (LLPS). Our approach encompasses the development of distinct models tailored for proteins highly predisposed to LLPS (LLPS+), those with moderate LLPS potential (LLPS-), and folded proteins resistant to LLPS (PDB*). Through the generation of 18,000 sequences, evenly distributed among the three model types, a majority of which demonstrate minimal similarity to proteins cataloged in the SwissProt database, we derive residue-level transition probability matrices. These matrices offer a probabilistic insight into the amino acid grammar unique to each dataset. Analysis of local sequence properties reveals the potential of sequences from LLPS+ GPT models to undergo phase separation. Subsequent validation through multi-chain simulations further substantiates the phase separating potential of the generated proteins and the generation of phase separating sequences from LLPS+ GPT. Additionally, we introduce and train a classifier capable of discerning whether a given amino acid sequence is prone to LLPS. This comprehensive investigation elucidates the molecular grammar of proteins, facilitating the integration of advanced computational methodologies with practical applications in generating protein sequences with desired phenotype.

## I. INTRODUCTION

Proteins play a crucial role in our survival by participating in all metabolic activities. Yet, for proteins to fulfill their functions, they must undergo specific three-dimensional folding, leading to the formation of secondary, tertiary, and quaternary structures. The higher-order structures are particularly vital for the optimal functioning of a majority of known proteins. The ongoing exploration of the intricate relationship between protein structure and function is an active area of research[1–3]. It's worth noting that not all functional proteins exhibit well-defined higher-order

*Electronic address: wabdul@tifrh.res.in

†Electronic address: jmondal@tifrh.res.in

structures.

Intrinsically disordered proteins (IDPs) represent a category of proteins lacking a well-defined secondary structure. They possess the ability to transition between various folded conformations based on the specific function they are required to perform. However, misfolding of such proteins are pathogenic. Upon misfolfing these can undergo fibrilization or liquid-liquid phase separation followed by liquid to solid phase transition. In both cases, they form solid aggregates that deposit as plaques resulting in incurable diseases such Alzheimer's, ALS and Parkinson's diseases which are caused by misfolding and aggregation of amyloid $\beta$-40, Prion Protein (PrP) and $\alpha$-synuclein respectively.

A hypothesis posits that IDPs, prior to aggregation, might experience spontaneous liquid-liquid phase separation (LLPS). Experimental observations indicate that IDPs can undergo LLPS in vitro when present in a solution above specific concentrations and under particular environmental conditions. Consequently, the LLPS of various proteins in diverse environments—whether physiological or otherwise—is currently under investigation through a combination of experimental[4–9], computational[10–18], and analytical[19–24] approaches.

To unravel the distinctive features of IDPs stemming from their amino acid sequences, extensive analysis of the primary sequences of numerous IDPs has been conducted[25, 26]. The development of models, such as the "sticker and spacer" model[27], has been instrumental in comprehending the LLPS-forming behavior of IDPs. Furthermore, studies have employed various metrics derived solely from the primary structures of different protein classes to accurately quantify and characterize the LLPS-forming properties of IDPs[26, 28]. Collectively, these investigations suggest the existence of different set of rules governing the amino acid arrangement of IDPs and folded proteins. Consequently, these rules, often referred to as the "molecular grammar" of these proteins, contribute to shaping their unique phenotypes. In this study we try to understand the grammar of IDPs with the use of machine learned Protein Language Models (PLMs). PLMs uses blocks of transformer neural networks to understand the context in a given protein sequence and hence can not only be used to understand the molecular grammar but also can be used to generated new IDPs and also traverse the protein sequence space.

The advent of Transformers[29] and the availability of robust computing hardware have ushered in the era of powerful large language models (LLMs). One notable example is the renowned ChatGPT, which has captivated both experts and non-experts with its capabilities. The remarkable accuracy achieved by current LLMs can be attributed to the utilization of transformer neural networks. Transformers excel in comprehending sequential data by leveraging token and positional

embeddings to discern the "context" within a sequence. With an ample amount of data and models of sufficient scale (comprising billions of parameters), transformers exhibit a proficiency in understanding language in a probabilistic manner. This proficiency grants them an implicit understanding of the inherent set of rules or grammar governing the language. Since the amino acid sequences of proteins adhere to a set of rules very specific to proteins and entirely different to natural languages, language models developed for protein sequences have been coined as Protein Language Models (PLMs) as they understand the grammar of the amino acid sequences rather than a language.

In this investigation, we employ decoder-only transformer-based neural networks to train GPT models on protein sequences, aiming to enable the models to grasp the intrinsic molecular grammar of these sequences. Utilizing pre-curated protein datasets—LLPS+, LLPS- and PDB*[25] we train individual GPT models on each dataset. Subsequently, we generate a substantial number of amino acid sequences with varying lengths using these models. Our initial analysis involves examining and comparing specific features of these generated sequences with corresponding features extracted from the original datasets.

Furthermore, we compute an amino acid transition probability matrix to gain insights into the molecular grammar of intrinsically disordered proteins (IDPs) and folded proteins from a probabilistic perspective. To validate our findings, we conduct both all-atom and coarse-grained simulations. For instance, we demonstrate that a short protein derived from the LLPS+ GPT model can undergo spontaneous liquid-liquid phase separation (LLPS). Additionally, through all-atom simulations, we illustrate that a sequence obtained from the PDB* GPT model maintains a folded structure.

Subsequently we use the pre-trained GPT models to build a balanced dataset for the purpose of training a ML based classifier for distinction between LLPS prone sequences and sequences that do not undergo LLPS. Using the dataset we train a extreme gradient boosted classifier that can distinguish whether a given sequence can undergo LLPS or not with high accuracy.

This research endeavors to decode the molecular grammar of proteins through the application of machine learning. It seeks to discern the nuanced distinctions in the grammar of intrinsically disordered proteins (IDPs) and folded proteins, representing a stride towards unraveling the underlying factors contributing to their respective phenotypes. Finally the developement of a classifier that can distinguish between proteins that can or cannot LLPS would help in preliminary screening of new protein sequences discovered in the future.

## II. METHODS

### A. Datasets

In this investigation, we utilized the already reported LLPS+, LLPS-, and PDB* datasets[25].

The LLPS+ dataset encompasses 137 amino acid sequences of intrinsically disordered proteins (IDPs) with critical/saturation concentrations either identical or lower than 100 µM. In contrast, the LLPS- dataset comprises 84 amino acid sequences of IDPs with critical/saturation concentrations exceeding 100 µM.

The PDB* dataset consists of 1562 folded proteins that do not spontaneously undergo liquid-liquid phase separation (LLPS) unless under extremely drastic conditions.

Protein sequences were denoted using the single letter codes for amino acids. Notably, all datasets include proteins with seleno-cysteine, denoted by the single-letter code U, and proteins with missing residues, represented by the letter X. To prepare the protein sequences of each dataset for training, a single text file was generated for each dataset. Within these files, protein sequences were separated by a "blank space" to indicate the end of a sequence.

We used a character level tokenization for training resulting in a vocabulary of the following characters: **ACDEFGHIKLMNPQRSTUVWXY<blank space>**.

### B. Tokenization and positional embeddding

For any PLM to work, we need to convert the characters to numbers that the machine can understand. Thus we use character-level tokenization ($t$) of the amino acid sequences to convert a protein's primary sequence to an array of numbers, more specifically integers. The character level tokenization has been summarized in Table-I

| **character** | A | C | D | E | F | G | H | I | K | L | M | N | P | Q | R | S | T | U | V | W | X | Y | <blank space> |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **token** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |

TABLE I: The numeric representation of the vocabulary

To comprehend the context or significance of the sequence order, Protein Language Models (PLMs) require information about the position of each token within a given sequence. Achieving this involves the use of positional embedding. The positional embedding value for any character at position $p$ in a sequence of length $l$ is determined by Eq-1[29].

$$p_e(p) = \begin{cases} \sin(\frac{p}{10000^{2i/d}}) \text{ if i is even} \\ \cos(\frac{p}{10000^{2i/d}}) \text{ if i is odd} \end{cases} \tag{1}$$

where $i$ is the positional encoding along the $i$-th dimension for a $d$ dimensional embedding of the tokens.

Therefore, the final numerical representation of a character in a given sequence is obtained by adding the token embedding and the positional embedding. Through this process, sequences of characters are transformed into machine-understandable numerical values, while retaining crucial information about the sequential order of the characters.

### C. Model details

We employ a decoder-only GPT model with an architecture resembling Figure-1. Specifically designed for amino acid sequences, this model serves as an advanced tool for sequence generation, eliminating the need for an encoder component. Trained on a specific type of protein sequences, the model excels at predicting subsequent amino acids given an initial sequence or amino acid, ensuring that the generated sequence reflects the properties of the proteins used during training.

In essence, the model undergoes a comprehensive training phase to grasp the intricate language of amino acids, analogous to mastering the grammatical structure of a language. When presented with a partial protein sequence of length $l$ as input, the model leverages its acquired knowledge to analyze the context of surrounding amino acids. This analysis is facilitated by multiple layers ($N$) of transformer neural networks, each utilizing a "multi-headed self-attention" mechanism to comprehend the contextual nuances.

Our current implementation of the decoder-only GPT model features 8 layers of multi-headed self-attention, with each layer containing 8 heads. The chosen sequence length for training is $l = 128$. This meticulous configuration optimizes the model's performance across diverse protein sequences.

The core process involves generating a probability distribution for potential amino acids at the next position in the sequence based on learned context. The model then selects the most probable amino acid, iteratively extending the sequence. Noteworthy is the model's adaptability, enabling it to provide diverse sequence predictions and offer multiple potential outcomes instead of adhering to a rigid pattern.

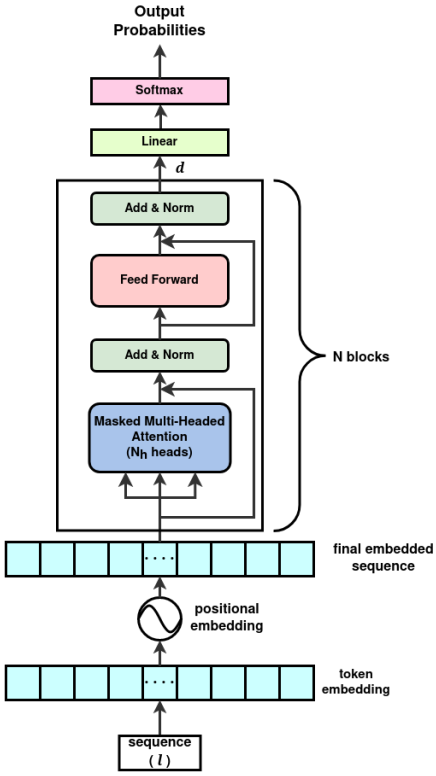


FIG. 1: A diagram representing the decoder only GPT model we have used.

The term "pre-trained" emphasizes the model's wealth of knowledge acquired during its initial training on a dataset as vast as possible. This foundational knowledge allows the model to discern general protein sequence patterns, facilitating the generation of meaningful and contextually relevant sequences even with partial input. The model converts partial input into an abstract vector representation with a length of $d$, subsequently transforming it into probabilities from which the next amino acid is predicted. For our study, we set $d = 256$.

### D. Training details

The training process for each model in our study involved a substantial training duration of $10^6$ epochs, ensuring a thorough exploration of the model's learning capabilities. We adopted a batch size of 64 to efficiently process and update the model parameters. The choice of the cross-entropy loss function, a standard measure for evaluating the dissimilarity between predicted and actual sequences, guided the training process. Additionally, we employed the AdamW optimizer with a specified learning rate of 0.0001. AdamW, renowned for its effective weight decay mechanism, contributed to the optimization of model parameters during training.

To assess the model's performance and generalization, we meticulously monitored both training and validation losses. The evolving trends in these losses over the epochs provide crucial insights into the model's learning dynamics and its ability to adapt to new data. Detailed visualizations of the training and validation losses are presented in Figure-S1, offering a comprehensive understanding of the model's progression. The chosen metrics and optimizer configuration aim to strike a balance between model convergence and preventing overfitting, ensuring that the trained models generalize well to new, unseen data. Overall, this rigorous training regimen, coupled with systematic loss monitoring, establishes a robust foundation for the subsequent evaluation and analysis of the model's predictive capabilities.

### E. Prediction of sequences from the models

We initiated the generation of 2000 sequences for each of the GPT models: LLPS+, LLPS-, and PDB*. These sequences, ranging in length from 20 to 200, were initialized with a random character from the vocabulary. It's noteworthy that the vocabulary includes X, U, and <blank space> as characters. To ensure the correctness of the predicted sequences, we made slight adjustments:

1. Any occurrence of <blank space> was replaced with the preceding amino acid.
2. Subsequently, all X's, if present, were substituted with the preceding amino acid.
3. Finally, any instance of U was replaced with C.

These corrections resulted in 2000 realistic amino acid sequences from each model, that are expected to reflect the properties of proteins associated with the datasets on which the respective models were trained.

### F. All-atom molecular dynamics simulation details

GROMACS-2023.2 was employed for conducting all-atomistic molecular dynamics simulations[30, 31].The amber99-disp[10] forcefield was utilized for low complexity sequences and potential Intrinsically Disordered Regions (IDRs), while charmm36m was applied for folded structures[32]. Initial steps involved solvating each simulation system, followed by achieving electro-neutrality by introducing $Na^+$ and $Cl^-$ ions. Additional 100 mM NaCl ions were incorporated to maintain physiological concentrations of NaCl throughout the simulations.

Subsequently, each system underwent energy minimization, followed by equilibration in an NVT ensemble for 100 ps, employing a timestep of 2 fs. This was succeeded by equilibration in an NPT ensemble for 100 ps, utilizing a timestep of 2 fs. All simulations were conducted at 310.15 K, with a reference pressure of 1 bar, where applicable. To control temperature and pressures, the v-rescale thermostat and the c-rescale barostat were employed. The production runs were then executed using the respective equilibrated systems, with a timestep of 2 fs.

### G. Phase co-existence molecular dynamics simulation details

Phase coexistence molecular dynamics were performed as per [14]. We used the Mpipi coarse-grained forcefield for this purpose[14]. For the generation of the initial conformations, we used `packmol`[33] and packed 100 protein chains inside an elongated box with the following dimensions: 25 nm, 25 nm, and 250 nm in the x, y, and z directions, respectively. We then performed NVT simulations at a high temperature (350.15 K) for 1 $\mu$s with a timestep of 10 fs using LAMMPS.

Starting with the final conformation obtained from the simulation at the high temperature (350.15 K), we performed simulations at different lower temperatures in the following manner:

1. We first cooled the system down to the desired temperature over 100,000 steps.

2. The system was then equilibrated at the desired temperature for 100,000 more steps.

3. The resultant conformation was then used for 2 $\mu$s production runs.

### H. Dataset preparation for the LLPS classifier

All the sequences (generated + real) were first pooled and subsequently labelled as LLPS forming (1) or not LLPS forming (0). This resulted in a total of The amino acid sequences were converted into vector representations by using prostT5[34], an encoder only PLM that has been pre-trained on 2.2 M protein sequences present in AlphaFold DB. Hence we can assume that prostT5 has learnt the generic grammar of proteins can be safely used to obtain context-based embeddings for a given sequence of amino acids. Therefore for a sequence of lenthg $l$, the output representation was a tensor of the shape $(l, 1024)$. A Mean Pooling was then performed along the sequence length dimension to obtain a single vector, 1024 long. Therefore each sequence could be represented by a single vector of a fixed length. The labelled vectors were then generously shuffled and 20% of them were separated pit

## I. List of software

We have used only open-source software for this study. All simulations have been performed using GROMACS-2021[30, 31]. Snapshots were generated using PyMOL 2.5.4[35–37]. Analysis were performed using Python[38] and MDAnalysis[39, 40]. Figures were prepared using Matplotlib[41], Jupyter[42] and Inkscape[43]. All deep neural network models were built using PyTorch[44]

## III. RESULTS AND DISCUSSION

We trained three Protein Language Models (PLMs) on distinct datasets: LLPS+, LLPS-, and PDB* [25]. These models are respectively referred to as LLPS+ GPT, LLPS- GPT, and PDB* GPT. Given a prompt and the desired output length, each PLM can generate sequences up to the specified length. Initially, we generated 18,000 sequences—6,000 from each PLM.

### A. Generated sequences display minimal resemblance to previously identified sequences

We employed the three trained models to produce 6000 sequences from each model, with lengths ranging from 20 to 200 amino acids (refer to *Methods*). To validate the distinctiveness of the generated sequences from those utilized during training, we initially conducted a multiple sequence alignment using BLAST against the datasets employed for model training. For each sequence we focus on two metrics provided by BLAST that denotes how similar a given sequence is to other sequences present in the Swiss prot database: E and percentage identity. E determines the statistical significance of a match between a two sequences. Higher E represents a random match, i.e. the match is not statiscally significant and vice-versa. Percentage identity quantifies the residue-level similarity between two sequences. A higher similar means more perfect match and vice-versa. Our analysis revealed minimal alignment between the generated sequences and those existing in the training datasets. This substantiates that the sequences generated by the model differ significantly from those encountered during training.

Subsequently, we conducted multiple sequence alignments against the SwissProt database for all 6000 sequences. We observed that 99%, 20% and 98% of the sequences generated from LLPS+, LLPS- and PDB* GPT models have E-values greater than 0.001. This suggests that for LLPS+ and PDB*, almost all of the sequences do not align well with of the already known proteins curated in the SwissProt dataset. We think that for LLPS-, the dataset coverage of the protein space might have been too small for the LLPS- GPT to be able to generate novel sequences and hence the lower dissimilarity percentage. A similar result is also being shown by the statistics for the percentage identity of the alignments (Table II).

Analyzing the multiple sequence alignment results against SwissProt, a notable observation emerges. The majority of protein sequences generated from the LLPS+ and PDB* models exhibit a noteworthy degree of uniqueness, evident in their E-value and percentage identity statistics. Conversely, sequences generated from the LLPS- model display a higher alignment with proteins

TABLE II: Statistics of percentage identity from MSA

| sequences | minimum | maximum | mean | standard deviation |
|---|---|---|---|---|
| LLPS+ | 19.42 | 100.0 | 37.8 | 12.53 |
| LLPS- | 22.76 | 100.0 | 82.55 | 19.57 |
| PDB* | 17.35 | 86.67 | 37.45 | 10.29 |

already documented in the SwissProt database. It is crucial to emphasize that, during the training phase, our models were never exposed to the SwissProt database. This underscores the remarkable capability of the Protein Language Models (PLMs) trained on protein sequences. Not only do these models adeptly predict entirely novel and unseen proteins, but they also navigate the intricate protein sequence space with remarkable efficiency. This affirms the models' capacity to generate diverse and distinctive protein sequences, showcasing their potential to contribute meaningfully to the exploration of uncharted areas within the realm of protein sequences.

### B. Different models learn different embeddings of the sequences

Given the substantial reliance of Protein Language Models (PLMs) on the effective embedding of input tokens, it becomes crucial to assess whether diverse models acquire distinct embeddings, even when the input tokens appear visually similar to the human eye. Consequently, we gather sequence embeddings for all 6000 predicted sequences, labeling them based on the originating model. Due to variations in the shapes of embeddings, we adopt mean pooling along the sequence length to ensure a consistent vector representation for each sequence, with a fixed length of 256, corresponding to the embedding dimension value.

To further analyze and visualize the relationships between these representations, we employ an Autoencoder[45], projecting the 256-dimensional data onto a 2-dimensional latent space (refer to Figure 2a). The specifics of the autoencoder architecture are detailed in the *Supplementary Methods* and a schematic of it has been shown in Figure 2a.

Figure 2b shows the loss function profile of the autoencoder during training. The smooth decay of the initial high loss to a constant value and the indifference between the training and the validation losses instill confidence regarding accurate projection of the 256 dimensional representation vector space onto a 2 dimensional latent space.

As illustrated in Figure 2c, the latent space projections reveal a clear separation of embeddings for the real and the generated sequences. This observation indicates distinct inherent patterns in

the learned embeddings of different models.

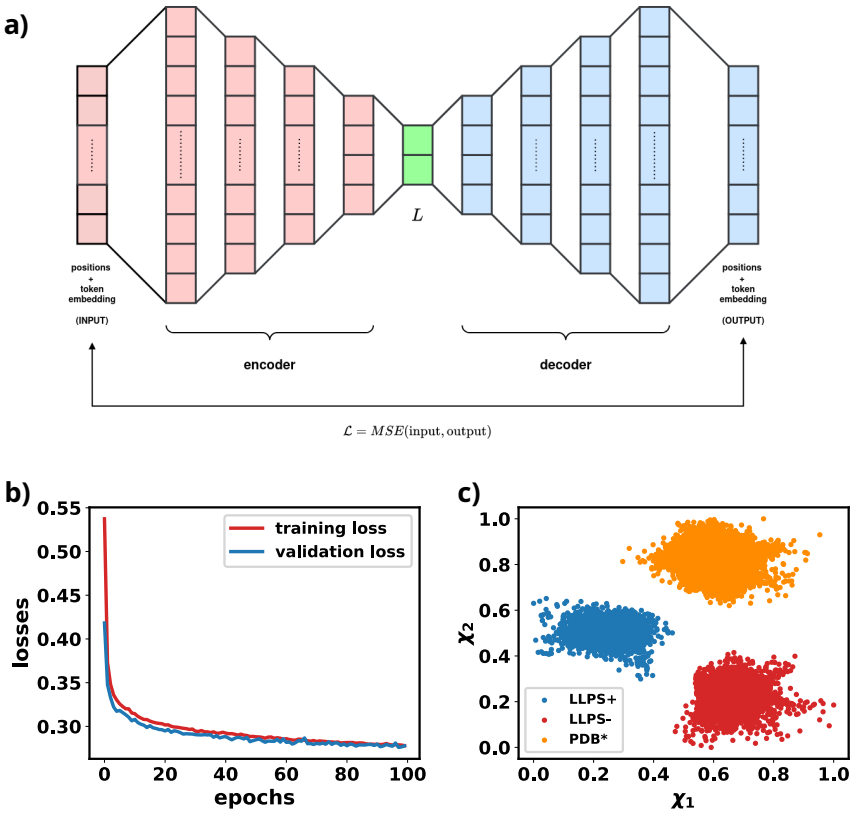


FIG. 2: **a)** A schematic of the fully connected neural network based deep autoencoder we have used. **b)** Training and validation losses for the autoencoder. **c)** Latent space projections of embeddings obtained for generated and real sequences from the trained GPT models. **c)** Latent space projections of embeddings otained for sequences in the LLPS+ (blue), LLPS- (red) and PDB* (yellow) datasets from their respective GPT models.

### C. Understanding the molecular grammar of proteins

While we observe that the PLMs successfully capture the molecular grammar of the trained sequences, it remains in an abstract and unintelligible form to us. To gain insight into the probabilistic interpretation of the rules governing amino acid sequences, we construct transition probabilities with a lag of 1 amino acid (refer to Figures 3, and *SI Methods*).

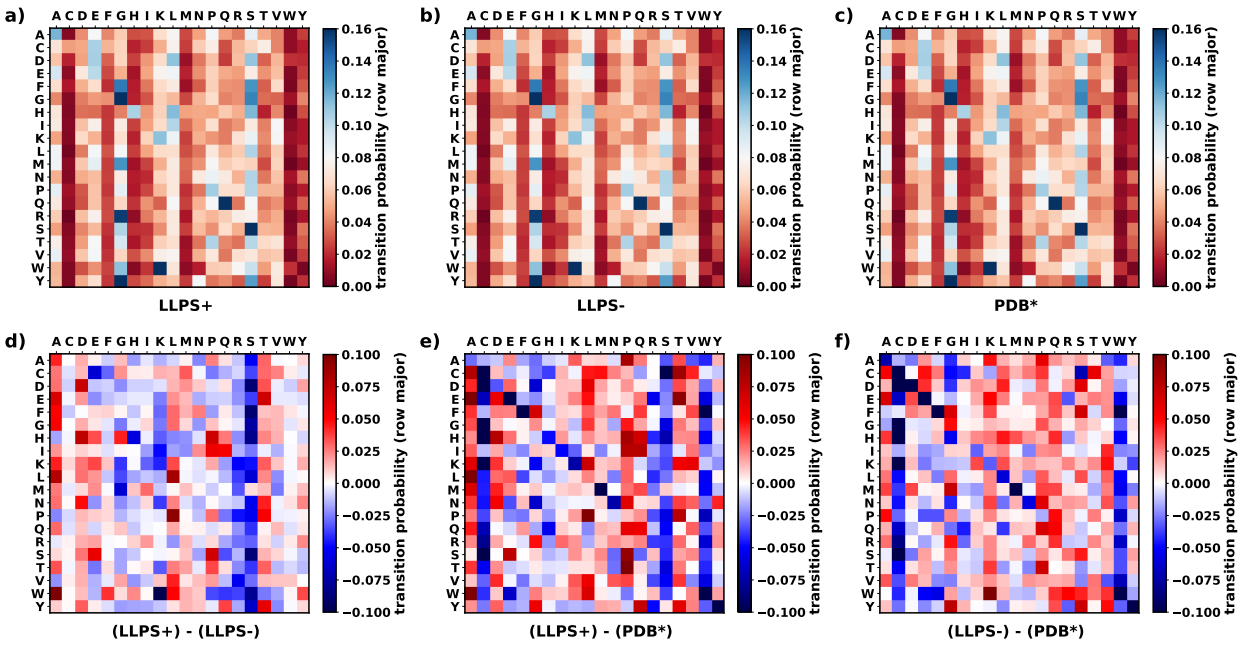


FIG. 3: transition probability among amino acids for lag=1 for the sequences generatared from the GPT model of : **a)** LLPS+. **b)** LLPS-. **c)** PDB*.

In Figures 3a-c it is evident that certain probabilities, specifically S-S, G-G, R-G, Y-H, W-K, and Q-Q, consistently exhibit high values regardless of the model as can be seen from the blue colour of those matrix elements. This means that in most of the sequences analyzed, one would find a serine followed by another serine, a glutamine by another glutamine and so on. These findings imply a universal trend across all proteins for these residue pairs, indicating a generic amino acid grammar. However, our primary interest lies in examining the distinctions in the transition probabilities of amino acids within the sequences generated by the three distinct GPT models.

Figure 3d illustrates the distinctions in transition probabilities between sequences generated by LLPS+ and LLPS- GPT models. An examination of the matrix reveals that a majority of the elements in the difference matrix between LLPS+ and LLPS- for datasets approach zero, indicating a high degree of similarity in their molecular grammar. Conversely, in Figures 3e, and 3f, there is a presence of dark red or dark blue colored elements, signaling that sequences generated by the PDB* GPT model exhibit a notably distinct molecular grammar from those produced by LLPS+ or LLPS- models. Notably, between LLPS+ and PDB*, we find that cysteine, serine, glycine and tryptophan have higher probabilities of appearing after any given amino acid in the LLPS+ sequences than the sequences belonging to PDB*. Apart from serine, the other three amino acids have higher probabilities of appearing in the LLPS- sequences as compared to PDB*. Therefore a

major difference between LLPS+ and LLPS- sequences is the occurance of serine after an amino acid in a sequence.

However, across all the difference plots (Figures 3d-f), the maximum disparity in transition probabilities remains low (around ∼0.1). This suggests that the manifestation of Intrinsic Disordered Proteins (IDPs) does not arise from a drastic difference in their molecular grammar compared to that of folded proteins. Instead, it is these subtle differences that facilitate their ability to transition among multiple conformations, some or all of which may be aggregation-prone.

### D. Properties of the predicted sequences

In Figure 4a, the distribution of sequence lengths for all generated sequences is depicted, with distinct colors corresponding to the model of origin. It is evident that there is no inherent bias discernible in the lengths of amino acid sequences generated across different models.

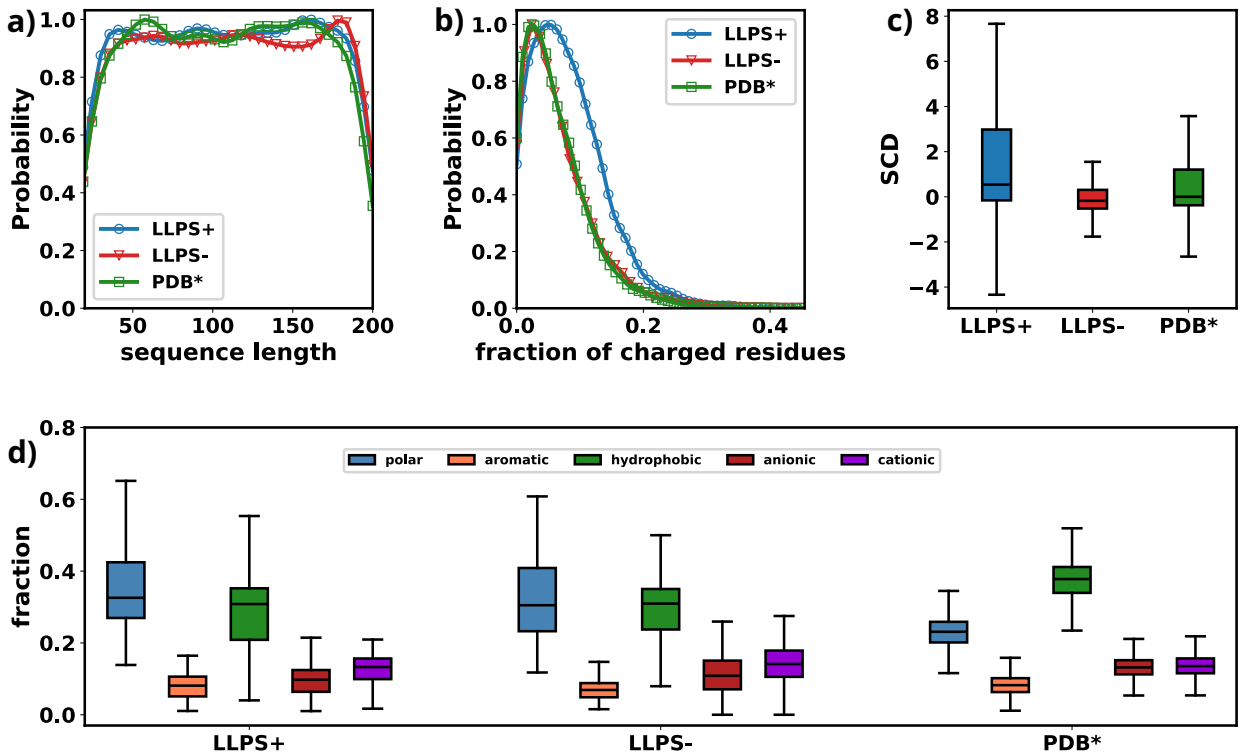


FIG. 4: **a)** Distribution of sequence lengths for all three generated sequence sets. **b)** Distribution of the mean charge of a sequence for all three generated sequence sets. **c)** Comparison of sequence charge decoration among the sequences generated from the different GPT PLMs. **d)** Comparison of the probability of occurrence of amino acids in a sequence among the three generated sequence sets.

Subsequently, we scrutinized the fraction of charged residues (FCR) present in the sequences

generated from each model (Figure 4b). It reveals a higher FCR for LLPS+ than LLPS- and PDB*. Additionally, they highlight the similarity in distributions of net and average charges between LLPS+ and PDB*. Notably, sequences generated LLPS+ GPT exhibit the presence of more charged residues in a given sequence than the sequences generated from LLPS- or PDB*. This hints at the presence of extensive charge patterning in sequences from LLPS+ GPT.

We next calculated the sequence charge decoration (SCD) for all the sequences which plot in Figure 4c as per Eq-2. SCD determines the charge distribution in a given sequence and a higher value of SCD can hint at a higher LLPS forming propensity. Figure 4b compares the scd values of the sequences between corresponding real and generated sequences as well between different sets of sequences. We observe that the real sequences all exhibit the same extent of charge patterning. On the other hand, sequences generated from LLPS+ GPT show the higher values of SCD than the sequences generated from LLPS- and PDB* GPT. This hints at a LLPS forming nature of the predicted sequences and imparts confidence in the PLMs we trained.

$$SCD = \frac{1}{N_{pairs}} \sum_{i=0}^{N-2} \sum_{j=i+1}^{N-1} \frac{q_i q_j}{\sqrt{|i-j|}} \tag{2}$$

We next categorized the amino acid sequences into five types: polar, aromatic, hydrophobic, anionic and cationic. We then calculated the fraction of each type of residues per sequence for the generated sequences. We then plot their distributions via boxplots in Figure 4d. From Figure 4d, we observe that sequences generated from LLPS+ GPT have a higher fraction of polar residues (blue) than the rest. Aromatic residues (orange) seems to be low in all sequences in the dataset, which is a feature which the ML models learnt from the real dataset. We next observe that LLPS+ GPT sequences have a low fraction of hydrophobic residues while PDB* GPT sequences have the highest fraction of hydrophobic residues. The cationic and anionic residues seem to be present to the same extent in all sequences with a slight bias in the LLPS+ GPT sequences.

Overall, Figure 4 shows that sequences generated from LLPS+ GPT have higher amount charged residues than the others, with a higher extent of charge patterning and a lower amount of hydrophobic residues. These strongly suggest that the sequences generated from LLPS+ GPT can potentially undergo phase-separation. However molecular simulations are required to confirm whether these sequences actually have phase separating potential or not.

### E. Phase co-existence simulations validate the LLPS forming nature of the predicted sequences

To validate whether the sequences predicted by LLPS+ GPT actually have a tendency to undergo LLPS, we conducted phase co-existence simulations on a few generated sequences using MPIPI forcefield[14]. For ease of simulations, we generated 200 sequences using LLPS+ GPT with sequence lengths varying between 20 to 25. For each sequence the mean LCR probability was then calculated, which is the mean of the individual LCR forming probabilities of each amino acid in a single sequence. The LCR forming probabilities were predicted using the PrDOS webserver[46].

#### 1. *Example protein 1*

The first protein we use as an example as the sequence **GGGRGGRGRGGMGRGGFG-GSRGYS**. A multiple sequence alignment of the sequence against the LLPS+ dataset reports an E-value of 1.5 while MSA with the SwissProt database shows an E-value of 6e-7 with residues 367 to 386 of FUS from *Bos taurus* with a percentage identity of 85%. Therefore the sequence is not present in the LLPS+ dataset and is also not a part of the FUS LCD. Hence it is a new sequence generated by the model.

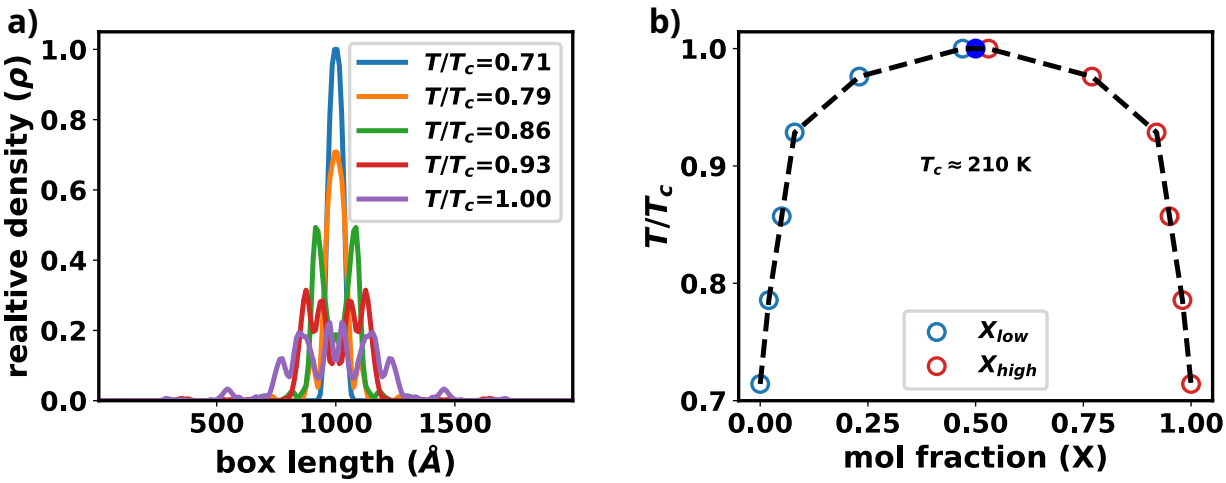


FIG. 5: **a)** Linear density profiles along the major axis of the simulation box for various temperatures below Tc, the critical temperature. **b)** relative temperature vs. mole fraction phase-diagram for the predicted sequence.

We then perform phase co-existence simulations as described in *Methods* at various temperatures. We then calculate the linear densities along the major axis which we plot in Figure 5a.

Using the mole fraction of the proteins present in the dense and the dilute phase calculated from the linear densities, we plot that vs the relative temperature of the simulations in Figure 5b to obtain a phase diagram. A fit of these points with Eq-3, we obtain $A$= 0.3 and $T_c$= 210 K where $T_c$ is the critical temperature and $A$ is a parameter unique to the amino acid sequence. A value of $T_c$ = 210 K conveys that the generated sequence can undergo LLPS.

$$\chi_H - \chi_L = A(Tc - T)^{\beta} \tag{3}$$

### 2. *Example protein 2*

The first protein we use as an example as the sequence **YGQTPKEPEAVSPSLRRMGRP**. A multiple sequence alignment of the sequence against the LLPS+ dataset reports an E-value of 3.1 while MSA with the SwissProt database shows an E-value of 1.3 with residues 56 to 70 of FUS from *Mycobacterium tuberculosis* variant bovis AF2122/97 with a percentage identity of 60%. Therefore the sequence is not present in the LLPS+ dataset and is also not a part of any known protein. Thus it is a new and unique sequence generated by the model.

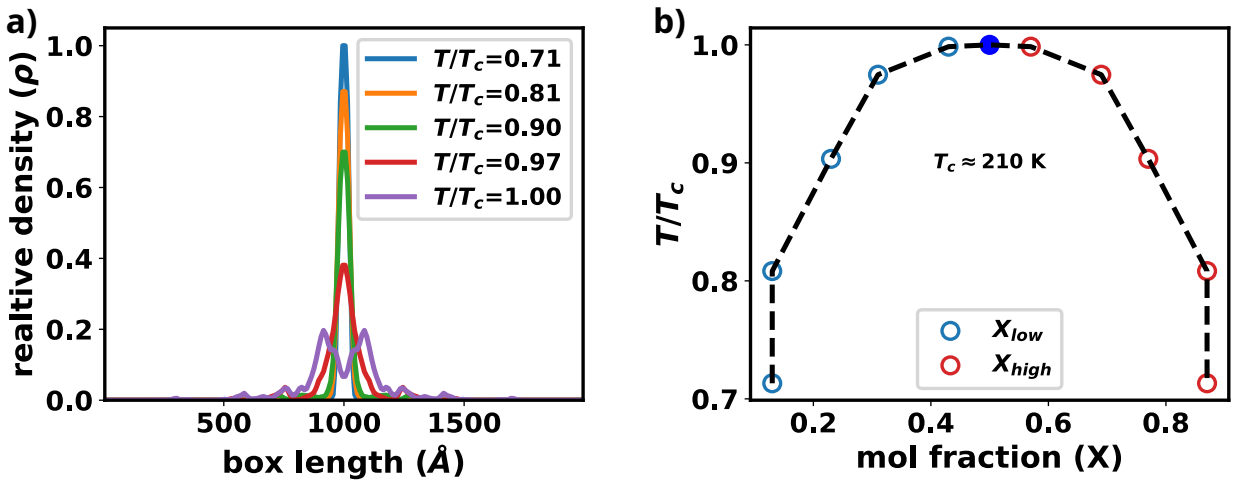


FIG. 6: **a)** Linear density profiles along the major axis of the simulation box for various temperatures below Tc, the critical temperature. **b)** relative temperature vs. mole fraction phase-diagram for the predicted sequence.

Similar to the previous example, we perform phase co-existence simulations as described in *Methods* at various temperatures. From the linear densities of the protein along the major axis (Figure 6a), we calculate the mol fractions of the protein in the dense and the dilute phase to obtain a phase diagram (Figure 6b). A fit with Eq-3 provides us with $A$=0.2 and $T_c$=210.3 K.

Thus both the examples show a high tendency of the sequences predicted by the LLPS+ GPT model to undergo LLPS. This validates that the LLPS+ GPT has properly learnt the molecular grammar of IDPs and can be used to generate amino acid sequences prone to LLPS.

### F. All-atom simulations of a predicted folded protein

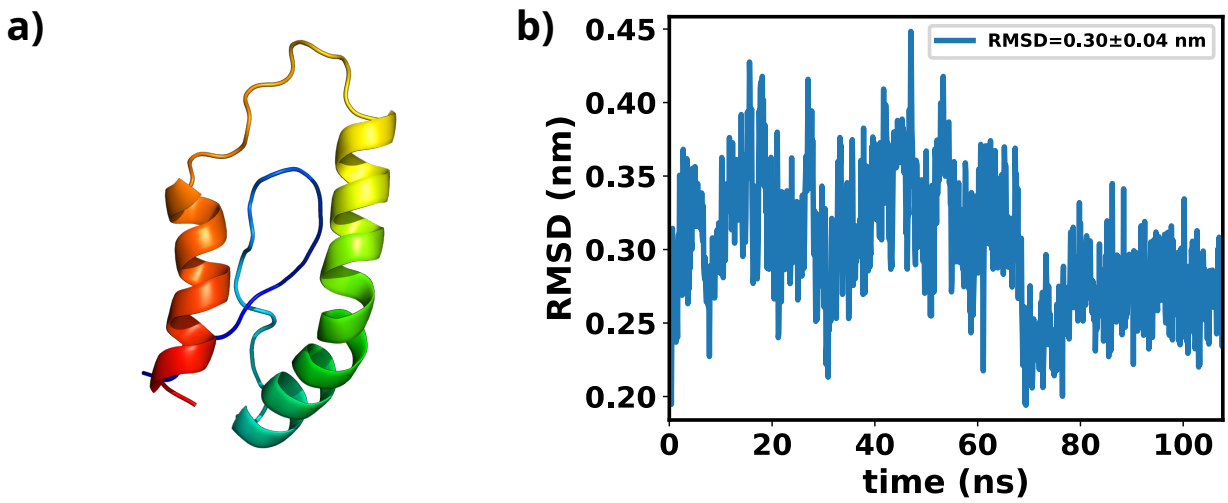


FIG. 7: **a)** Initial conformation obtained from AlphaFold 2.0.**b)** Time profile of RMSD for prot80

To also validate whether the PDB* GPT model has learnt the molecular grammar of folded proteins, we generate a 67 residue protein (sequence provided below) and obtain its possible folded structure using AlphaFold 2.0[47] (Figure 7a).

1 - **HNCILYTEGCRWGLIPNMNLNDVRDWWRYWWP** - 43

44 - **MGPLLLHEWTSLKPPNWPLPEPESYTLLFHLEWYL** - 67

Using that as the initial conformation, we perform multiple short 100 ns NPT production runs to attain a sampling of 1 $\mu$s. Subsequent analysis of RMSD shows that the protein conformation does not change over 100 ns (Figure 7b) proving the fact that the generated sequence is that of a folded protein. This renders more credibility to the PLMs we have trained.

### G. Design of a classifier between LLPS and non-LLPS forming proteins from their sequence

As discussed earlier that the number of known sequences that can undergo LLPS is extremely low in number. Thus training a classifier would have not been possible due to either a very small

dataset size or due to the large imbalance in the dataset with a dominant majority of it being composed of non-LLPS forming protein sequences. However, with the LLPS+ and LLPS- PLMs, we generated synthetic sequences that can undergo LLPS in an attempt to create a balanced dataset for training a classifier. We thus generated 6000 sequences each from LLPS+ and LLPS- PLMs, providing us with a total of 12000 sequences that can potentially have LLPS forming properties. We then extracted out the first 12000 protein sequences which do not contain any missing residues from the BMRB database. Combining these with the already present LLPS+, LLPS- and PDB* databases, we obtained a total of about 24500 protein sequences that we used to train a classifier. We labelled sequences from the LLPS+ or LLPS- PLM/dataset as 1, represeting their potential to form LLPS. The sequences from the BMRB database and PDB* were given a label of 0, i.e. that they cannot form LLPS.

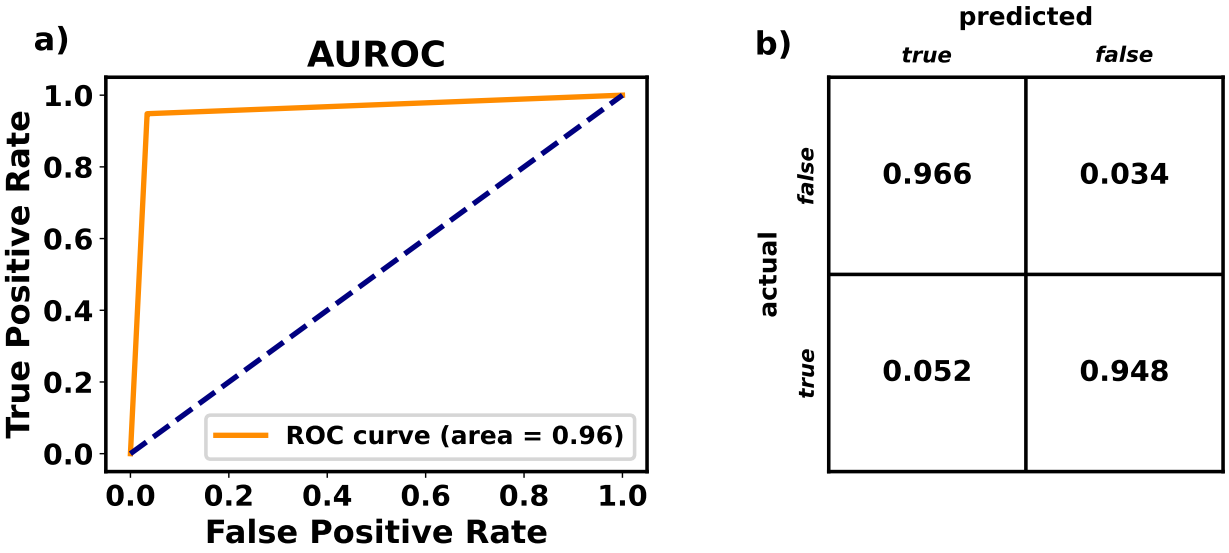


FIG. 8: **a)** Area under receiving operator characteristics (AUROC) curve for the trained classifier. **b)** Confusion matrix for the classifier.

However, the sequences are textual data and cannot directly be used in training. Therefore we used the recently reported ProstT5[34], an encoder based pre-trained PLM to obtain embeddings for the sequences. We used prostT5 to obtain vectorized representations of each amino acid for 24000 protein sequences followed by mean pooling along the amino acid sequence dimension to obtain a single vector representation of a fixed length (1024) for each sequence.

We next prepared the training and testing datasets from the labelled, vectorized representations of the amino acid sequences as described in *Methods*. These were then used to train an **e**xtreme **g**radient **b**oosted (XGB) classifier form the XGBoost python library[48] in a supervised manner.

The primary task was to determine the optimum hyper parameters for the classifier. For accurate determination of the this, we used the GridSearchCV function from python's scikit-learn[49, 50] library and obtained an optimum value of maximum tree depth = 4 and number of estimators = 1024 for the classifier given it is being trained on the dataset we designed. Upon a supervised training using these parameters. We then made predictions on the sequences we had reserved for testing (please see *Methods* for details). The area under receiving operator characteristics (AUROC) curve (Figure 8a) shows a accuracy of 96% on the test dataset. The corresponding confusion matrix (Figure 8b) also shows that the classifier is also very accurate with a F1 score of 95%.

Therefore with the help of the PLMs we trained, we were able to expand the dataset of LLPS forming proteins. Using these were then trained an XGB classifier which is very accurate in predicting whether a given sequence can undergo LLPS or not.

## IV. CONCLUSION

In conclusion, our study offers a comprehensive exploration of the molecular grammar of proteins through the utilization of specialized language models and advanced computational methodologies. By training GPT models on datasets representing varying propensities for liquid-liquid phase separation (LLPS), we generated 18000 amino acid sequences, 6000 from each model. Through residue-level transition probability matrices, we provided a probabilistic interpretation of amino acid grammar specific to each dataset, shedding light on the underlying molecular mechanisms governing LLPS. Furthermore, our investigation revealed intriguing insights into the behavior of proteins highly prone to LLPS, including the identification of sequences exhibiting notable intrinsically disordered protein behavior, validated through atomistic simulations and/or multi-chain coarse grained simulations, which underscore their LLPS potential. These PLMs can be used to explore the infinitely large protein space in an efficient manner thereby facilitating the discovery and characterization of unknown protein sequences. Additionally, we have developed and trained a classifier capable of predicting whether a given sequence of amino acids can undergo LLPS or not.

Moving forward, our findings lay the groundwork for further exploration and refinement of predictive models and computational tools, opening avenues for advancements in protein engineering, drug discovery, and bioinformatics research. As a future prospect, we aim to develop a simple web interface for easy access and utilization of the models trained and used in this study, thereby enhancing accessibility and facilitating broader adoption of our methodologies for the scientific

community. This study represents a significant step towards unraveling the complexities of protein behavior and underscores the importance of interdisciplinary approaches in deciphering the molecular grammar of biological systems.

## V. DATA AVAILABILITY

The scripts and models used to train GPT models and predict sequences have been uploaded to https://github.com/JMLab-tifrh/idpGPT. Data not being shared via GitHub can be made available upon request.

## VI. ACKNOWLEDGEMENT

We would like to thank Prof. Anand Srivastava, Mr. Santosh Prajapati, and Dr. Jerelle Joseph for their insights in performing on phase-coexistence simulations using MPIPI forcefield. We thank Prof. Anand Srivastava for suggesting the calculation of SCD. We acknowledge support of the Department of Atomic Energy, Government of India, under Project Identification No. RTI 4007. JM acknowledges Core Research grants provided by the Department of Science and Technology (DST) of India (CRG/2023/001426).

---

[1] James C Whisstock and Arthur M Lesk. Prediction of protein function from protein sequence and structure. *Quarterly reviews of biophysics*, 36(3):307–340, 2003.

[2] Ambrish Roy, Alper Kucukural, and Yang Zhang. I-tasser: a unified platform for automated protein structure and function prediction. *Nature protocols*, 5(4):725–738, 2010.

[3] Jianyi Yang and Yang Zhang. Protein structure and function prediction using i-tasser. *Current protocols in bioinformatics*, 52(1):5–8, 2015.

[4] Soumik Ray, Nitu Singh, Rakesh Kumar, Komal Patel, Satyaprakash Pandey, Debalina Datta, Jaladhar Mahato, Rajlaxmi Panigrahi, Ambuja Navalkar, Surabhi Mehra, et al. $\alpha$-synuclein aggregation nucleates through liquid–liquid phase separation. *Nature chemistry*, 12(8):705–716, 2020.

[5] Yufan Xu, Runzhang Qi, Hongjia Zhu, Bing Li, Yi Shen, Georg Krainer, David Klenerman, and Tuomas PJ Knowles. Liquid–liquid phase-separated systems from reversible gel–sol transition of protein microgels. *Advanced Materials*, 33(33):2008670, 2021.

[6] Azeem Ahmad, Vladimir N Uversky, and Rizwan Hasan Khan. Aberrant liquid-liquid phase separation and amyloid aggregation of proteins related to neurodegenerative diseases. *International Journal of Biological Macromolecules*, 220:703–720, 2022.

[7] Samrat Mukhopadhyay, Anamika Avni, Ashish Joshi, Anuja Walimbe, and Swastik G Pattanashetty. A deep dive into biomolecular condensates using single-droplet surface-enhanced raman spectroscopy. *Biophysical Journal*, 122(3):60a, 2023.

[8] Paresh C Shrimali, Sheng Chen, Anirban Das, Rachel Dreher, Matthew K Howard, Jeremy J Ryan, Jeremy Buck, Darren Kim, Macy L Sprunger, Jai S Rudra, et al. Amyloidogenic propensity of self-assembling peptides and their adjuvant potential for use as dna vaccines. *Acta Biomaterialia*, 169:464–476, 2023.

[9] Semanti Mukherjee, Arunima Sakunthala, Laxmikant Gadhe, Manisha Poudyal, Ajay Singh Sawner, Pradeep Kadu, and Samir K Maji. Liquid-liquid phase separation of $\alpha$-synuclein: a new mechanistic insight for $\alpha$-synuclein aggregation associated with parkinson's disease pathogenesis. *Journal of Molecular Biology*, 435(1):167713, 2023.

[10] Paul Robustelli, Stefano Piana, and David E Shaw. Developing a molecular dynamics force field for both folded and disordered protein states. *Proceedings of the National Academy of Sciences*, 115(21):E4758–E4766, 2018.

[11] Gregory L Dignon, Wenwei Zheng, Young C Kim, Robert B Best, and Jeetain Mittal. Sequence determinants of protein phase behavior from a coarse-grained model. *PLoS computational biology*, 14(1):e1005941, 2018.

[12] Andrew P Latham and Bin Zhang. Maximum entropy optimized force field for intrinsically disordered proteins. *Journal of chemical theory and computation*, 16(1):773–781, 2019.

[13] Zakarya Benayad, Soren von Buulow, Lukas S Stelzl, and Gerhard Hummer. Simulation of fus protein condensates with an adapted coarse-grained model. *Journal of Chemical Theory and Computation*, 17(1):525–537, 2020.

[14] Jerelle A Joseph, Aleks Reinhardt, Anne Aguirre, Pin Yu Chew, Kieran O Russell, Jorge R Espinosa, Adiran Garaizar, and Rosana Collepardo-Guevara. Physics-driven coarse-grained model for biomolecular phase separation with near-quantitative accuracy. *Nature Computational Science*, 1(11):732–743, 2021.

[15] F Emil Thomasen, Francesco Pesce, Mette Ahrensback Roesgaard, Giulio Tesei, and Kresten Lindorff-Larsen. Improving martini 3 for disordered and multidomain proteins. *Journal of Chemical Theory and Computation*, 18(4):2033–2041, 2022.

[16] Abdul Wasim, Sneha Menon, and Jagannath Mondal. Modulation of $\alpha$-synuclein aggregation amid diverse environmental perturbation. *bioRxiv*, pages 2023–10, 2023.

[17] Subinoy Adhikari and Jagannath Mondal. Machine learning subtle conformational change due to phosphorylation in intrinsically disordered proteins. *The Journal of Physical Chemistry B*, 127(44):9433–9449, 2023.

[18] Sneha Menon and Jagannath Mondal. Conformational plasticity in $\alpha$-synuclein and how crowded environment modulates it. *The Journal of Physical Chemistry B*, 127(18):4032–4049, 2023.

[19] Clifford P Brangwynne, Peter Tompa, and Rohit V Pappu. Polymer physics of intracellular phase

transitions. *Nature Physics*, 11(11):899–904, 2015.

[20] Jeong-Mo Choi, Furqan Dar, and Rohit V Pappu. Lassi: A lattice model for simulating phase transitions of multivalent proteins. *PLoS computational biology*, 15(10):e1007028, 2019.

[21] Mina Farag, Samuel R Cohen, Wade M Borcherds, Anne Bremer, Tanja Mittag, and Rohit V Pappu. Condensates of disordered proteins have small-world network structures and interfaces defined by expanded conformations. *bioRxiv*, pages 2022–05, 2022.

[22] Frank Jülicher and Christoph A Weber. Droplet physics and intracellular phase separation. *Annual Review of Condensed Matter Physics*, 15, 2023.

[23] Hannes Ausserwöger, Daoyuan Qian, Georg Krainer, Ella de Csilléry, Timothy J Welsh, Tomas Sneideris, Titus M Franzmann, Seema Qamar, Nadia A Erkamp, Jonathon Nixon-Abell, et al. Quantifying collective interactions in biomolecular phase separation. *bioRxiv*, pages 2023–05, 2023.

[24] Mina Farag, Alex S Holehouse, Xiangze Zeng, and Rohit V Pappu. Fireball: A tool to fit protein phase diagrams based on mean-field theories for polymer solutions. *Biophysical Journal*, 2023.

[25] Kadi L Saar, Alexey S Morgunov, Runzhang Qi, William E Arter, Georg Krainer, Alpha A Lee, and Tuomas PJ Knowles. Learning the molecular grammar of protein condensates from sequence determinants and embeddings. *Proceedings of the National Academy of Sciences*, 118(15):e2019053118, 2021.

[26] Zhaoming Chen, Chao Hou, Liang Wang, Chunyu Yu, Taoyu Chen, Boyan Shen, Yaoyao Hou, Pilong Li, and Tingting Li. Screening membraneless organelle participants with machine-learning models that integrate multimodal features. *Proceedings of the National Academy of Sciences*, 119(24):e2115369119, 2022.

[27] Jeong-Mo Choi, Anthony A Hyman, and Rohit V Pappu. Generalized models for bond percolation transitions of associative polymers. *Physical Review E*, 102(4):042403, 2020.

[28] Shengming Zhou, Yetong Zhou, Tian Liu, Jia Zheng, and Cangzhi Jia. Predllps_pssm: a novel predictor for liquid–liquid protein separation identification based on evolutionary information and a deep neural network. *Briefings in Bioinformatics*, 24(5):bbad299, 2023.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[30] David Van Der Spoel, Erik Lindahl, Berk Hess, Gerrit Groenhof, Alan E Mark, and Herman JC Berendsen. Gromacs: fast, flexible, and free. *Journal of computational chemistry*, 26(16):1701–1718, 2005.

[31] Mark James Abraham, Teemu Murtola, Roland Schulz, Szilárd Páll, Jeremy C Smith, Berk Hess, and Erik Lindahl. Gromacs: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*, 1:19–25, 2015.

[32] Jing Huang, Sarah Rauscher, Grzegorz Nawrocki, Ting Ran, Michael Feig, Bert L De Groot, Helmut Grubmüller, and Alexander D MacKerell Jr. Charmm36m: an improved force field for folded and

intrinsically disordered proteins. *Nature methods*, 14(1):71–73, 2017.

[33] Leandro Martínez, Ricardo Andrade, Ernesto G Birgin, and José Mario Martínez. Packmol: A package for building initial configurations for molecular dynamics simulations. *Journal of computational chemistry*, 30(13):2157–2164, 2009.

[34] Michael Heinzinger, Konstantin Weissenow, Joaquin Gomez Sanchez, Adrian Henkel, Martin Steinegger, and Burkhard Rost. Prostt5: Bilingual language model for protein sequence and structure. *bioRxiv*, 2023.

[35] Schrödinger, LLC. The AxPyMOL molecular graphics plugin for Microsoft PowerPoint, version 1.8. November 2015.

[36] Schrödinger, LLC. The JyMOL molecular graphics development component, version 1.8. November 2015.

[37] Schrödinger, LLC. The PyMOL molecular graphics system, version 1.8. November 2015.

[38] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009.

[39] Naveen Michaud-Agrawal, Elizabeth J Denning, Thomas B Woolf, and Oliver Beckstein. Mdanalysis: a toolkit for the analysis of molecular dynamics simulations. *Journal of computational chemistry*, 32(10):2319–2327, 2011.

[40] Richard J Gowers, Max Linke, Jonathan Barnoud, Tyler JE Reddy, Manuel N Melo, Sean L Seyler, Jan Domanski, David L Dotson, Sébastien Buchoux, Ian M Kenney, et al. Mdanalysis: a python package for the rapid analysis of molecular dynamics simulations. In *Proceedings of the 15th python in science conference*, volume 98, page 105. SciPy Austin, TX, 2016.

[41] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[42] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, and Carol Willing. Jupyter notebooks – a publishing format for reproducible computational workflows. In F. Loizides and B. Schmidt, editors, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87 – 90. IOS Press, 2016.

[43] Inkscape Project. Inkscape.

[44] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[45] Wei Wang, Yan Huang, Yizhou Wang, and Liang Wang. Generalized autoencoder: A neural network framework for dimensionality reduction. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 490–497, 2014.

[46] Takashi Ishida and Kengo Kinoshita. Prdos: prediction of disordered protein regions from amino acid sequence. *Nucleic acids research*, 35(suppl_2):W460–W464, 2007.

[47] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Applying and improving alphafold at casp14. *Proteins: Structure, Function, and Bioinformatics*, 89(12):1711–1721, 2021.

[48] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[49] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[50] Oliver Kramer and Oliver Kramer. Scikit-learn. *Machine learning for evolution strategies*, pages 45–53, 2016.