Interpretable Pairwise Distillations for Generative Protein Sequence Models

Christoph Feinauer
Department of Decision Sciences
Bocconi Institute for Data Science and Analytics
Bocconi University, Milan, Italy
christoph.feinauer@unibocconi.it

Barthelemy Meynard-Piganeau
Laboratory of Computational and Quantitative Biology
CNRS - Sorbonne Université, Paris, France
Department of Applied Science and Technologies
Politecnico di Torino, Turin, Italy
barthelemy.meynard@polytechnique.edu

Carlo Lucibello
Department of Decision Sciences
Bocconi Institute for Data Science and Analytics
Bocconi University, Milan, Italy
carlo.lucibello@unibocconi.it

Abstract

Many different types of generative models for protein sequences have been proposed in literature. Their uses include the prediction of mutational effects, protein design and the prediction of structural properties. Neural network (NN) architectures have shown great performances, commonly attributed to the capacity to extract non-trivial higher-order interactions from the data. In this work, we analyze three different NN models and assess how close they are to simple pairwise distributions, which have been used in the past for similar problems. We present an approach for extracting pairwise models from more complex ones using an energy-based modeling framework. We show that for the tested models the extracted pairwise models can replicate the energies of the original models and are also close in performance in tasks like mutational effect prediction.

1 Introduction

While generative models for protein sequences promise a rich field of applications in biology and medicine [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], the question of what information they extract from the sequence data has received less attention. This is, however, a very interesting field of research since especially the more complex models might extract non-trivial higher-order dependencies between residues. This in turn might reveal interesting biological insights.

Some recent works address this interpretability issue. In Ref. [11], the authors introduce the notion of *pairwise saliency* and use it to quantify the degree to which more complex models learn structural information and how this relates to the performance in the prediction of mutational effects. Ref. [12]

Machine Learning for Structural Biology Workshop, NeurIPS 2021.

instead constructs pairwise approximations to categorical classifiers and showcases applications to models trained on protein sequence data.

In this work, we ask how close trained neural network (NN) based models are to the manifold of pairwise distributions. To this end, we train three different architectures on protein sequence data. Interpreting these models as energy-based models [13], we present a simple way to extract pairwise models from them and analyze errors in energy between extracted and original models. We show that the subtle question of gauge invariance is important for this purpose and address this invariance ambiguity using different objective functions for the extraction.

2 Methods

2.1 Protein Sequences and Energy-Based Models

We represent the aligned primary structure of a protein domain of length N as a sequence $s = (s_1, \dots, s_N)$, where we identify every possible amino acid with a number between 1 and q. Our input data are sets of evolutionary related sequences gathered in multiple sequence alignments (MSAs).

Energy-based models (EBMs) [13] are models that specify the negative unnormalized log-probability $E_\theta(s)$, for example by a neural network with weights and biases represented by θ. While the calculation of the exact probability $p(s) = {e^{-E_\theta(s)}}/{Z_\theta}$ is intractable since the normalization constant Z_θ is a sum over q^N terms, numerous ways of training such models have been developed.

In this work, we use the fact that *any* probability $p(s)$ can be thought of as an EBM by defining $E(s) = -\log p(s)$. We will use the term *energy* for both cases: when derived from a distribution $p(s)$, which is typically normalized, and when given by an explicit energy function, which is typically not normalized.

2.2 Energy Expansions and Gauge Freedom

We call $I = \{1, \dots, N\}$ the set of all positions in the sequence s and s_L the subsequence consisting of amino acids at positions in $L \subseteq I$. Then, we can expand any energy $E(s)$ as

$$E(s) = \sum_{L \subseteq I} f_L(s_L), \tag{1}$$

where f_L is a function depending only on the amino acids at positions at L. Models for which $f_L = 0$ for $|L| > 2$ are called *pairwise models* (or *Potts models*) and their energy can be written as a special case of Eq. 1 as

$$E^{pw}(s) = -\sum_{i=1}^{N} \sum_{j=i+1}^{N} J_{ij}(s_i, s_j) - \sum_{i=1}^{N} h_i(s_i) - C, \tag{2}$$

with J being commonly called couplings and h the fields [14]. The constant C is typically not added to the model definition since it does not change the corresponding probabilities, but we keep it in order to be consistent with the generic expansion in Eq. 1.

The expansion in Eq. 1 is not unique and additional constraints can be imposed to fix the expansion coefficients (gauge fixing). A common route is to impose the so-called *zero-sum* gauge [12], which aims to shift as much of the coefficient mass to lower orders as possible (see, e.g., Ref[12] and Appendix B.2). We will show that the question of gauge is crucial for understanding the structure of the fitness landscape induced by the NN models.

2.3 MSE Formulation

We formulate the problem of extracting a pairwise model from more general models by using a loss function $\mathcal{L}$ that measures the average mean squared difference between energies with respect to a distribution D over sequences. We use the loss

$$\mathcal{L}(c, h, J) = \mathbb{E}_{s\sim D}\left[\left(E^{M}(s) - E^{pw}(s)\right)^{2}\right], \tag{3}$$

where E^{pw}(s) is the energy of a pairwise model described in Eq. 2 and $E^{M}(s)$ is the energy of the original model. We minimize the loss function with respect to h, J and C and use the resulting pairwise model as an approximation to E^{M}.

The distribution D is central in this formulation of the problem and is closely related to the question of gauges. It can be shown that if D is the uniform distribution over sequences, the minimizer of $\mathcal{L}(c, h, J)$ is equivalent to the pairwise part of E^{M} in the zero-sum gauge (see Appendix B for a proof). By changing D it is possible to give more weight to these regions and construct a pairwise model that might be worse in replicating E^{M} globally, but better in regions of interest. This is equivalent to extracting the pairwise interactions in a different gauge of E^{M}.

A natural candidate for D is the distribution induced by E^{M}, leading to pairwise models that aim to reproduce the original distribution well on typical sequences of that distribution. With this choice, the loss corresponds to an f-divergence ($f(t) = log^{2}(t)$) in the unnormalized distribution space [15].

3 Results

3.1 Extraction of Fourier Coefficients

We train three different probabilistic models (the autoregressive architecture presented in [5] (ArDCA), an energy based model expressed by a multi-layer perceptron with a single hidden layer (MLP) and a variational autoencoder [8] (VAE), on five different MSAs taken from [3]. We prepare samples from the uniform distribution and the model distribution and calculate their energies for all models and minimize the loss in Eq. 3 using these samples (see Appendix A.2 for details of the models and the training procedure).

3.2 Energy Errors

In Fig. 1 we show the error in the energies of extracted pairwise models with respect to the energies in the original models. We use two different distributions D in Eq. (3) for sampling the sequences used for the extraction of the pairwise models: U stands for the uniform distribution; M for the distributions of the original trained models. The error in the plot is the mean squared error, normalized by the range (see Appendix A.1). For all models, the error drops by several orders of magnitude when using the model distribution M for extraction instead of the uniform distribution U. However, for ArDCA the error is already considerably smaller than for the other models when using the uniform distribution, which can be taken as evidence that this model is close to a pairwise distribution after training. The MLP and VAE on the other hand, show very large errors when using the uniform distribution. This can be taken as evidence that these models are not pairwise models *globally*, but close to pairwise in the space of sequences on which the models are typically used.

3.3 Mutational Effect Prediction using Extracted Models

The prediction of mutational effects is a typical field of application for the type of models analyzed in this work. In Fig. 2 we show the Spearman correlations between the experimental data and the energies in the original models (O), the energies of models extracted using samples from a uniform distribution (U) and the energies extracted from the original model distribution (M). There is no clear tendency with respect to the relative performance of the original and the extracted models. This is evidence that most of the explanatory power of the original models can be reproduced by simpler pairwise models, even though the exact distribution used for extraction does not seem to be important. This indicates that only coarse features in the energy are used in this prediction task.

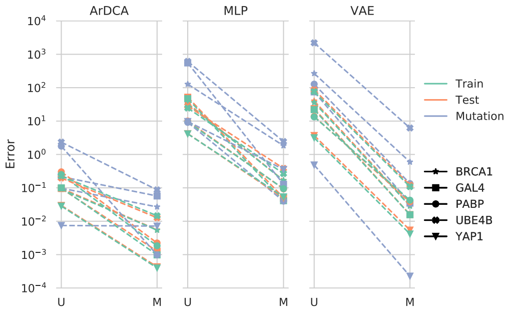

Figure 1: **Errors in energies of the extracted pairwise models with respect to the original models.** The three columns correspond to the three different models tested (ArDCA, MLP and VAE). The colors indicate which dataset is tested: Train data (green), test data (orange) and data from the mutational assays (blue). The markers distinguish the different protein families tested. Within every column, the left (U) corresponds to pairwise models extracted with samples from the uniform distribution, the right (M) to pairwise models extracted with samples from the distribution of the original models. The error shown is the normalized root-mean squared error (see Appendix A.1). Note the logarithmic scale.

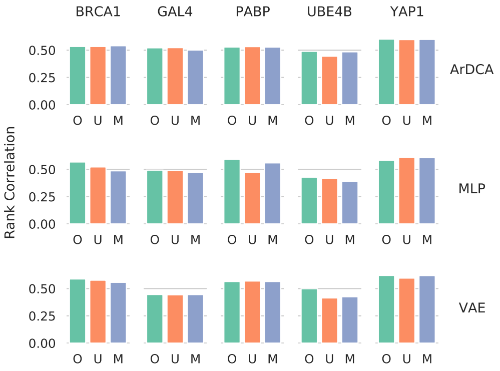

Figure 2: **Spearman Correlation with experimental data of original (O) and extracted models (U, M)**. Every plot corresponds to a combination of original model type (ArDCA, MLP and VAE) with a mutational assay. Shown is the Spearman rank correlation between the experimental data and the energies of the original model (O), the model extracted using samples from a uniform distribution (U) and using samples from the original model distribution (M).

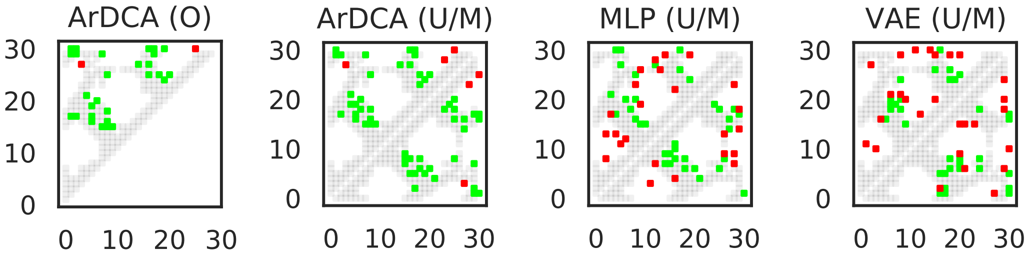

Figure 3: **Contact prediction using extracted models** Contact predictions vs. ground truth for the top $N = 30$ predicted contacts for models extracted from ArDCA, the VAE and the MLP. Horizontal and vertical axes show positions. True contacts are grey, true positives are green and false positives are red. In the three right plots, the upper parts show the contacts for models extracted with the uniform distribution, the lower parts show the same for models extracted with the original model distribution. The left-most plot shows the contact predictions for ArDCA from the original method in [5].

3.4 Contact Prediction

Given that the extracted models are pairwise models, we can use standard methods from this field to predict structural contacts [14, 16] (see Appendix A.3). For ArDCA, the contact predictions for these two methods of extraction are largely the same, and also very similar to the predictions from the original method. This is consistent with the idea that ArDCA is very similar to a pairwise model. The predictions for the MLP are also very similar between the two methods and the overall performance is worse than ArDCA. The results for the VAE are similar, indicating that the VAE and the MLP are either not relying on structural information for predicting mutational effects or our method is not able to extract this information. We note similar results in [11].

4 Discussion

In this work, we provide evidence that the neural network based generative models for protein sequences analyzed by us can be approximated well by pairwise distributions. The autoregressive architecture on which ArDCA is based seems to be closest to a pairwise model after training. For the MLP and the VAE, the results seem to indicate that while these models are less well approximated by a pairwise model globally, their pairwise projection is a very close approximation in the part of the sequence space in which they are typically used, close to the data manifold.

We cannot of course exclude that the neural network models tested by us do extract some meaningful higher-order interactions from the data, but the results seem to indicate that their effect is rather subtle. This suggests that the general strategy outlined in [17], where the pairwise part of the model is kept explicitly and an universal approximator is used for extracting higher-order interactions, might be promising.

Several interesting further lines of research suggest themselves. While the general idea of approximating a pairwise distribution over fixed-length sequences to models trained on unaligned data (like recent very large attention-based models [18]) seems to be ill-defined, the approach of *locally* extracting a pairwise model highlighted in this work might still be feasible. Another interesting question is whether sparse higher-order interactions can be efficiently extracted from neural network based models. It is for example possible that methods like the Goldreich-Levin algorithm [19] might be adapted for pseudo-boolean functions based on generative models for protein sequence data.

References

[1] Zachary Wu, Kadina E Johnston, Frances H Arnold, and Kevin K Yang. Protein sequence design with deep generative models. *Current Opinion in Chemical Biology*, 65:18–27, 2021.

[2] Simona Cocco, Christoph Feinauer, Matteo Figliuzzi, Rémi Monasson, and Martin Weigt. Inverse statistical physics of protein sequences: a key issues review. *Reports on Progress in Physics*, 81(3):032601, 2018.

[3] Thomas A Hopf, John B Ingraham, Frank J Poelwijk, Charlotta PI Schärfe, Michael Springer, Chris Sander, and Debora S Marks. Mutation effects predicted from sequence co-variation. *Nature biotechnology*, 35(2):128–135, 2017.

[4] Christoph Feinauer and Martin Weigt. Context-aware prediction of pathogenicity of missense mutations involved in human disease. *arXiv preprint arXiv:1701.07246*, 2017.

[5] Jeanne Trinquier, Guido Uguzzoni, Andrea Pagnani, Francesco Zamponi, and Martin Weigt. Efficient generative modeling of protein sequences using simple autoregressive models. *arXiv preprint arXiv:2103.03292*, 2021.

[6] Jung-Eun Shin, Adam J Riesselman, Aaron W Kollasch, Conor McMahon, Elana Simon, Chris Sander, Aashish Manglik, Andrew C Kruse, and Debora S Marks. Protein design and variant prediction using autoregressive generative models. *Nature communications*, 12(1):1–11, 2021.

[7] Adam J Riesselman, John B Ingraham, and Debora S Marks. Deep generative models of genetic variation capture the effects of mutations. *Nature methods*, 15(10):816–822, 2018.

[8] Xinqiang Ding, Zhengting Zou, and Charles L Brooks III. Deciphering protein evolution and fitness landscapes with latent space models. *Nature communications*, 10(1):1–13, 2019.

[9] Alex Hawkins-Hooker, Florence Depardieu, Sebastien Baur, Guillaume Couairon, Arthur Chen, and David Bikard. Generating functional protein variants with variational autoencoders. *PLoS computational biology*, 17(2):e1008736, 2021.

[10] Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R Eguchi, Po-Ssu Huang, and Richard Socher. Progen: Language modeling for protein generation. *arXiv preprint arXiv:2004.03497*, 2020.

[11] Dylan Marshall, Haobo Wang, Michael Stiffler, Justas Dauparas, Peter Koo, and Sergey Ovchinnikov. The structure-fitness landscape of pairwise relations in generative sequence models. *bioRxiv*, 2020.

[12] Stefano Zamuner and Paolo De Los Rios. Interpretable neural networks based classifiers for categorical inputs. *arXiv preprint arXiv:2102.03202*, 2021.

[13] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

[14] Faruck Morcos, Andrea Pagnani, Bryan Lunt, Arianna Bertolino, Debora S Marks, Chris Sander, Riccardo Zecchina, José N Onuchic, Terence Hwa, and Martin Weigt. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences*, 108(49):E1293–E1301, 2011.

[15] I. Csiszár. Information-type measures of difference of probability distributions and indirect observation. *Studia Scientiarum Mathematicarum Hungarica*, 2:229–318, 1967.

[16] Magnus Ekeberg, Cecilia Lövkvist, Yueheng Lan, Martin Weigt, and Erik Aurell. Improved contact prediction in proteins: using pseudolikelihoods to infer potts models. *Physical Review E*, 87(1):012707, 2013.

[17] Christoph Feinauer and Carlo Lucibello. Reconstruction of pairwise interactions using energy-based models. *arXiv preprint arXiv:2012.06625*, 2020.

[18] Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alexander Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. *bioRxiv*, 2021.

[19] Ryan O'Donnell. *Analysis of boolean functions*. Cambridge University Press, 2014.

[20] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

[21] Kurt Binder, Dieter Heermann, Lyle Roelofs, A John Mallinckrodt, and Susan McKay. Monte carlo simulation in statistical physics. *Computers in Physics*, 7(2):156–157, 1993.

[22] 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*, 32:8026–8037, 2019.

[23] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[24] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[26] Stanley D Dunn, Lindi M Wahl, and Gregory B Gloor. Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction. *Bioinformatics*, 24(3):333–340, 2008.

[27] Rama Ranganathan, Kun Ping Lu, Tony Hunter, and Joseph P Noel. Structural and functional analysis of the mitotic rotamase pin1 suggests substrate recognition is phosphorylation dependent. *Cell*, 89(6):875–886, 1997.

A Methods

A.1 Energy Error

We measure the error in energies in the extracted models with respect to the energies in the original models using the *normalized root-mean square deviation*, i.e.

$$error \;=\; \frac{\frac{1}{M}\sum_{m=1}^{M}\left(E^{M}(s_m)-E^{pw}(s_m)\right)^2}{\max_m E^{M}(s_m)-\min_m E^{M}(s_m)}, \tag{4}$$

where $\{s_m\}_{m=1}^{M}$ is the set of sequences on which we calculate the error, E^M is the energy of the original model, E^{pw} the energy of the extracted pairwise model and $\max_m E^M(s_m)$ and $\min_m E^M(s_m)$ are the maximum and minimum energies of the original model on the dataset.

A.2 Models and Sampling

A.2.1 ArDCA

The model used in ArDCA [5] decomposes the probability $p(s)$ of a sequence of amino acids as

$$p(s)=\prod_{i=1}^{N}p(s_i|s_{<i}), \tag{5}$$

where s_i is the amino acid at position i and $s_{<i}$ are the amino acids that come before i in the ordering. The conditional probability $p(s_i|s_{<i})$ is then defined as

$$p(s_i|s_{<i})=\frac{\exp\left(h_i(s_i)+\sum_{j=1}^{i-1}J_{ij}(s_i,s_j)\right)}{z_i(s_{<i})} \tag{6}$$

where $z_i(s_{<i})$ is the sum of the denominator over all possible values of $s_{<i}$. We use the code by the authors for training the model and calculating $\log p(s)$ for the samples used for extraction. Training was done with sequence reweighting as implemented by the authors of [5].

A.2.2 MLP

The MLP is a simple feed-forward network with one hidden layer of size H. The energy E^{MLP} for sequence s is calculated as

$$E^{MLP}(s)=\sum_{k=1}^{H}W_k^2\, f\left(\sum_{i=1}^{Nq}W_{ki}^1\hat{s}_i+b_k\right) \tag{7}$$

where $\hat{s}$ is a one-hot encoding of the sequence s, W^1 and W^2 are a weight matrix and a weight vector respectively, and b is the bias vector. The activation function f was chosen as the leaky ReLU [20]. We used $H = 64$ hidden units, a L2 regularization of 0.001. The training was done using Pseudolikelihoods inspired by [16]. See [17] for the definition of the loss function when using EBMs on proteins. We did not use sequence reweighting [14] for the MLP. Training was done for 200 epochs. After training, the energy can be calculated using a single forward pass. For sampling from this model, we resorted to standard MCMC techniques [21]. Since we have to evaluate the energy a large number of times during sampling, we used a very small number of MC sweeps (MC steps divided by the length of the sequence) for thermalization (1000 sweeps) and sampling (every 5 MC sweeps after thermalization). While this certainly does not lead to high-quality samples, we note that we are only interested in biasing the extraction towards sequences more typical of the distribution. The model was implement in PyTorch [22].

A.2.3 Variational Autoencoder

The model and code we use is based on the work and implementation of [8]. For a more detailed introduction to the variational autoencoder we refer to the original work [23]. Both encoder and decoder use a single hidden layer with 100 hidden neurons and $tanh$ activations. The dimension of the latent space is 20. During training, an $L2$ regularization of 0.1 was used and the training was run for 10000 epochs. Following the implementation of [8], we used full-batch gradient descent with and Adam optimizer.

The probabilities were estimated using importance sampling [24] using 1000 ELBO samples. Training was done with sequence reweighting as implemented by the authors of [8].

A.2.4 Extraction

We use 10^7 samples from the uniform and 10^7 samples from the original model distributions after training for extracting the pairwise models. The samples are drawn independently for each combination of original model and dataset.

For the samples from the model distributions, we minimize the loss in Eq. 3 using a batch size of 10000 and the Adam optimizer [25]. We keep a running average l of the loss function using the equation $l_k = \alpha\, l_{k-1} + (1-\alpha)\mathcal{L}_k$ with the initial condition $l_1 = \mathcal{L}_1$ where $\mathcal{L}_k$ is the loss after gradient step k and l_k is the running average of the loss after gradient step k. We set $\alpha = 0.1$ and stop optimization if the running average has not reached a new minimum for 1000 gradient steps. The extraction runs in seconds to minutes on an Nvidia RTX 2080 GPU.

For samples from the uniform distribution, the minimizer of the loss in Eq. 3 can be calculated directly from the samples without gradient descent (see Appendix B). We use the same number of samples to approximate the conditional energy expressions in Eq. 13 this case.

A.3 Contact Prediction

We use standard methods for contact prediction from pairwise models, following mainly [16]. We transform the extracted pairwise models into the zero-sum gauge and calculate the Frobenius norm of the $q-1 \times q-1$ submatrices J_{ij} corresponding to the pair of positions i and j (we do not sum over gap states, hence $q-1$ instead of q). We apply the *average-product correction* [26] and sort the positions pairs by the resulting score, excluding pairs for which $abs(i-j) < 5$. We map PDB 1PIN:A [27] to the MSA and use it to differentiate contacts from non-contacts (8 Å, Heavy-Atom criterion [14]).

B Zero-Sum Gauge

In the following we prove that the pairwise model E^{pw} corresponding to the minimizer of Eq. 3 is equivalent to the pairwise part of E^M in the zero-sum gauge when using the uniform distribution D for extraction.

B.1 Notation

We denote by $\mathcal{A} = \{1, .., q\}$ the (numeric) alphabet of the q possible amino acids. The terms $f_L : \mathcal{A}^{|L|} \to \mathcal{R}$ in the general expansion in Eq. 1 are functions mapping sequences of amino acids of length $|L|$ to a real number, where $L \subseteq I = \{1, \dots, N\}$ is a subsequence of positions. In this notation, the pairwise model we train using the loss in Eq. 3 can be written as

$$E^{pw}(s) = \sum_{i=1}^{N} \sum_{j=i+1}^{N} f_{ij}^{pw}(s_i, s_j) + \sum_{i=1}^{N} f_i(s_i) + f_\emptyset. \tag{8}$$

In Eq. 2 we use a different notation for the pairwise model, but in this Appendix we decide to keep all notations compatible with the generic expansion in Eq. 1. The notations can be connected by identifying $f_i^{pw}(a) := -h_i(a)$, $f_{ij}^{pw}(a,b) := -J_{ij}(a,b)$ and $f_\emptyset := -C$ for arbitrary amino acids a and b.

Equivalently we define $f_L^M : \mathcal{A}^{|L|} \to \mathcal{R}$ as the interaction coefficients between the sites belonging to the set of positions $L \subseteq I$ in E^M in a certain gauge.

We will use $f_L(a_L)$ in order to denote a specific interaction coefficient for a fixed sequence of amino acids a_L of length $|L|$, for both pairwise models and models with higher-order interactions. We will use f^{pw} to denote the set of all parameters of the pairwise model and f^M for the set of all parameters of the original model.

B.2 Zero-Sum Gauge

The zero-sum gauge is a reparameterization of the interaction coefficients which leaves the energy invariant (see also Ref. [12] who discuss this gauge). In this gauge, if $|L| > 0$, summing $f_L(a_L)$ over any of the amino acids in a_L while keeping the others fixed is 0. It can be applied both to the parameters of the extracted pairwise model f^{pw} and the parameters f^M of the original model. Since the sum over an amino acid is proportional to the expectation of $f_L(a_L)$ when the corresponding amino acid is sampled uniformly, this condition can be written as

$$\mathbb{E}_{s\sim U}[f_L(s_L)|s_J = a_J] = 0 \quad \forall\, J \subset L, \tag{9}$$

where $\mathbb{E}_{s\sim U}[f_L(s_L)|s_J = a_J]$ the expectation of $f_L(s_L)$ if the subsequence s_J is fixed to a_J. Any model can be transformed into the zero-sum gauge using the identity $f_L(a_L) = (f_L(a_L) - \hat{f}_L(a_L)) + \hat{f}_L(a_L)$ with

$$\hat{f}_L(a_L) := \sum_{J\subseteq L} (-1)^{|J|} \frac{1}{q^{|J|}} \sum_{a_J} f^L(a_L). \tag{10}$$

It is easy to show that $\hat{f}(a_L)$ satisfies the condition in Eq. 9 and that $f_{a_L}(a_L) - \hat{f}_{a_L}(a_L)$ contains only interactions of order strictly less than $|L|$. Therefore, any model can be transformed into the zero-sum gauge by first applying the transformation to the interaction coefficients at the highest order $N = |I|$. This will lead to interaction coefficients at order N that satisfy the condition in Eq. 9 and new interaction coefficients of order lower than N. These can be absorbed in the interaction coefficients in the lower orders of the expansion. Repeating this procedure at $N-1$, then at $N-2$ etc. leads to a final model where all interaction coefficients of all orders satisfy the condition in Eq. 9.

Since the expansion of E^M has exponentially many interaction coefficients in general, this procedure has no practical use in our setting. However, in the next Section we show that the lower orders of E^M in the zero-sum gauge representation can be extracted with a simple sampling estimator.

B.3 Proof of Equivalence of Minimizer of Loss and Zero-Sum Gauge

The partial derivative of the loss in Eq. 3 with respect to a parameter $f_L^{pw}(a_L)$ in the pairwise model (note that $|L| \leq 2$ in this case) can be written as

$$\frac{\partial \mathcal{L}(f^{pw})}{\partial f_L^{pw}(a_L)} = 2\, \mathbb{E}_{s\sim U}\left[\left(E^{pw}(s) - E^M(s)\right) \frac{\partial E^{pw}(s)}{\partial f_L^{pw}(a_L)}\right]. \tag{11}$$

Setting the gradient to 0 leads to

$$\mathbb{E}_{s\sim U}[E^M(s)|s_L = a_L] = \mathbb{E}_{s\sim U}[E^{pw}(s)|s_L = a_L] \quad \forall L : |L| \leq 2 \tag{12}$$

which means that the minimisation of the loss with respect to the parameters of the pairwise model is equivalent to fitting the conditional expectation of the energy under uniform distribution up to the second order of the expansion.

Since the loss in Eq. 3 is invariant with respect to a gauge change in the pairwise model E^{pw}, we can assume without loss of generality that we extract the pairwise model in the zero-sum gauge representation. Using a hat to denote the parameters $\hat{f}^{pw}$ of the pairwise model in this specific gauge, it is easy to see from Eq. 8 and the condition in Eq. 9 that

$$\begin{aligned}\mathbb{E}_{s\sim U}[E^{pw}(s)] &= \hat{f}^{pw}_{\emptyset}\\ \mathbb{E}_{s\sim U}[E^{pw}(s)|s_i=a] &= \hat{f}^{pw}_{i}(a)+\hat{f}^{pw}_{\emptyset}\\ \mathbb{E}_{s\sim U}[E^{pw}(s)|s_i=a,s_j=b] &= \hat{f}^{pw}_{i,j}(a,b)+\hat{f}^{pw}_{i}(a)+\hat{f}^{pw}_{j}(b)+\hat{f}_{\emptyset}.\end{aligned}$$

Combining this with Eq. 12 we get at the minimum of the loss the conditions

$$\begin{aligned}\mathbb{E}_{s\sim U}[E^{M}(s)] &= \hat{f}^{pw}_{\emptyset}\\ \mathbb{E}_{s\sim U}[E^{M}(s)|s_i=a] &= \hat{f}^{pw}_{i}(a)+\hat{f}^{pw}_{\emptyset}\\ \mathbb{E}_{s\sim U}[E^{M}(s)|s_i=a,s_j=b] &= \hat{f}^{pw}_{i,j}(a,b)+\hat{f}^{pw}_{i}(a)+\hat{f}^{pw}_{j}(b)+\hat{f}_{\emptyset}.\end{aligned} \tag{13}$$

Similar to the pairwise model, we will use a hat to denote the parameters $\hat{f}^M$ of the model E^M in the zero-sum gauge. While the corresponding expansion

$$E^M(s)=\sum_{L\subseteq I}\hat{f}^M_L(s_L)$$

has interaction coefficients of all orders, we can again use the conditions in Eq. 9 to arrive at

$$\begin{aligned}\mathbb{E}_{s\sim U}[E^{M}(s)] &= \hat{f}^{M}_{\emptyset}\\ \mathbb{E}_{s\sim U}[E^{M}(s)|s_i=a] &= \hat{f}^{M}_{i}(a)+\hat{f}^{M}_{\emptyset}\\ \mathbb{E}_{s\sim U}[E^{M}(s)|s_i=a,s_j=b] &= \hat{f}^{M}_{i,j}(a,b)+\hat{f}^{M}_{i}(a)+\hat{f}^{M}_{j}(b)+\hat{f}_{\emptyset}.\end{aligned}$$

Taking these relations together leads to the minimizer condition

$$\hat{f}^{pw}_L=\hat{f}^M_L \quad \forall L:|L|\leq 2$$

which means that the E^{pw} minimizing the loss in Eq. 3 is the pairwise part of E^M in its zero-sum gauge representation. Note that the loss is still invariant with respect to a gauge change in the extracted pairwise model, so the extracted model can be in any gauge representation.

We also note that Eqs. 13 can be used to estimate the coefficients of the extracted pairwise model directly using uniform samples and the corresponding energies from the original models in order to approximate the expectations.