A BIOLOGY-INFORMED SIMILARITY METRIC FOR SIMULATED PATCHES OF HUMAN CELL MEMBRANE

Anonymous authors
Paper under double-blind review

ABSTRACT

Complex scientific inquiries rely increasingly upon large and autonomous multiscale simulation campaigns, which fundamentally require *similarity metrics* to quantify "sufficient" changes among data and/or configurations. However, subject matter experts are often unable to articulate similarity precisely or in terms of well-formulated definitions, especially when new hypotheses are to be explored, making it challenging to design a meaningful metric. Furthermore, the key to practical usefulness of such metrics to enable autonomous simulations lies in *in situ* inference, which requires generalization to possibly substantial distributional shifts in unseen, future data. Here, we address these challenges in a cancer biology application and develop a meaningful similarity metric for *"patches"*— regions of simulated human cell membrane that express interactions between certain proteins of interest and relevant lipids. In the absence of well-defined conditions for similarity, we leverage several biology-informed notions about data and the underlying simulations to impose inductive biases on our metric learning framework, resulting in a suitable similarity metric that also generalizes well to significant distributional shifts encountered during the deployment. We combine these intuitions to organize the learned metric space in a multiscale manner, which makes the metric robust to incomplete and even contradictory intuitions. Our approach delivers a metric that not only performs well on the conditions used for its development and other relevant criteria, but also learns key temporal relationships from statistical mechanics without ever being exposed to any such information during training.

1 INTRODUCTION

Many scientific phenomena involve wide ranges of spatial and temporal scales, but computational models usually cannot cover all relevant scales with sufficient fidelity. This challenge has given rise to multiscale simulations (Ayton & Voth, 2010; Hoekstra et al., 2014; Krzhizhanovskaya et al., 2015; Voth, 2017; Enkavi et al., 2019; Ingólfsson et al., 2021), where coarse and, thus, inexpensive approximations are used to explore large scales, whereas more-detailed but significantly more-expensive models are used to provide details for smaller regions in space and time. Here, we are interested in developing a *similarity metric* to facilitate such multiscale simulations in the context of cancer biology. The overarching goal is to explore the interactions of RAS proteins and RAS-RAF protein complexes with the lipid bilayer that forms the human cell membrane (Ingólfsson et al., 2017; 2020; 2021). RAS-RAF activation is a crucial part of the signaling chain that controls cell growth, and up to a third of all human cancers are driven by mutations of RAS proteins that corrupt this chain (Simanshu et al., 2017; Prior et al., 2020). Consequently, understanding the signaling process in detail is of significant interest (Waters & Der, 2018; Travers et al., 2018; Kessler et al., 2019).

The primary challenge is that signaling events are thought to depend on the spatial arrangement of lipids in the neighborhood of RAS, the conformation of RAS, its orientation relative to the membrane, and range of other factors. Yet, even using some of the largest supercomputers, only a few potential events can be explored with fine-scale, molecular dynamics (MD) models. To maximize the opportunity for discovery, computational scientists seek to employ coarse-scale models to continuously create wide-ranging membrane configurations and, from these, select a diverse subset to explore in detail. Mathematically, this requirement translates into defining a similarity metric between *"patches"*, which represent local membrane configurations characterized by lipid concentrations and proteins constellations, *i.e.,* numbers and types of proteins (illustrated in Figure 1).

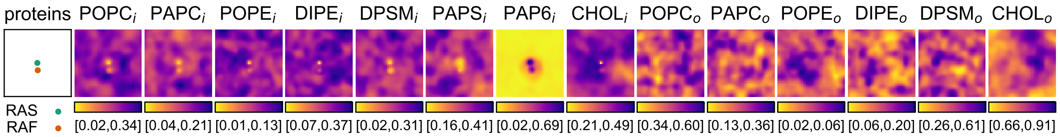

Figure 1: A "patch" comprises zero or more proteins of two types and concentration distributions of 14 types of lipids . Facilitating target multiscale simulations requires determining similarity among patches, even though there exists no well-formulated or widely-accepted notion of similarity.

Although similarity metrics are used widely, existing approaches employ simple, well-formulated measures, such as well-defined norms (*e.g.,* L_p-norms) or supervisory labels. However, in many scientific applications, including the one of interest here, there exist no explicit labels and standard norms do not match the biological understanding of the domain scientists. Instead, experts usually express various intuitions and hypotheses regarding what similar and dissimilar configurations might look like. For example, one may consider different mean lipid concentrations as different, or similar protein constellations as similar. However, the reverse usually does not hold, *i.e.,* same mean concentrations can differ in their spatial distribution and different protein configurations may show similar lipid behavior. Furthermore, these intuitions and hypotheses, which are usually based on experience or some initial observations, may also later turn out to be incomplete, inaccurate, or even incorrect, *e.g.,* when new data is obtained, new computational models are developed, or new phenomena are studied. Broadly, such biology-informed intuitions lead to *necessary-but-not-sufficient* conditions that might be under-constrained (too few necessary conditions) or even inconsistent

Contributions. We introduce *an approach to learn a similarity metric from a set of necessary-but-not-sufficient conditions* of similarity/dissimilarity for complex, multimodal data and demonstrate our method on patches generated from simulations of RAS-RAF-membrane interactions. Our framework uses metric learning (Xing et al., 2003; Lu et al., 2017; Kaya & Bílge, 2019; Suárez-Díaz et al., 2020) to incorporate biology-informed, but incomplete and contradictory, notions of similarity. We show that by casting such notions into a set of inductive biases, our approach yields a robust metric that *generalizes well to significant distributional shifts* encountered during deployment. We also demonstrate that our metric is *scientifically relevant* and captures the underlying biology, *e.g.,* preserves additional scientific constraints not part of the training. Most notably, our metric exhibits a strong correlation between similarity of patches and the timescales needed for one to evolve into the other. This correlation emerged naturally through metric learning and matches the fundamental assumptions of statistical mechanics approaches, despite the training data containing no notion of time evolution. Our metric was deployed for *in situ* inference as an enabling technology for a massive multiscale simulation of RAS-RAF-membrane biology to create the first-of-its-kind scientific campaign (Anonymous, 2021) to study this phenomena. Our technique can be *easily adapted* to a wide range of scientific problems, *e.g.,* other protein systems, leading to better design of experiments, more-stable predictive models, and better clustering, with the potential for significant impact.

2 Multiscale Simulations of RAS-RAF-Membrane Biology

Our goal is to facilitate massive multiscale simulations of RAS-RAF interactions with a human cell membrane. During a multiscale campaign, a continuum model (the "coarse-scale") simulation evolves density distributions of lipids and single particle representations of RAS and RAF proteins for a large (1 µm × 1 µm) portion of a membrane. With a particular interest in signaling events in the vicinity of RAS proteins, the focus is on exploring local *patches* (30 nm × 30 nm regions of the membrane) around proteins using molecular dynamics (MD, the "fine-scale") simulations. Whereas the ongoing continuum simulation creates a continuous stream of patches, all of which are candidates for MD simulations, the high computational cost of MD allows only a small fraction (*e.g.,* 0.05%) of the patches to be explored. Unfortunately, the distribution of patches is highly non-uniform with some types of configurations occurring orders of magnitude more often than others making a uniform random selection inefficient. Simultaneously, there exist little *a priori* insight into which patches might lead to interesting events to formulate a targeted acquisition function. Instead, our collaborators are aiming to instantiate a diverse set of patches covering as much of the phase space explored by the continuum model as possible. This implies a measure of similarity between patches which unfortunately is not well defined but only understood qualitatively.

A patch is a complex biological configuration that comprises concentration distributions of 14 types of lipids and two types of proteins (see Figure 1). Of particular interest is to understand how the different types of lipids rearrange themselves in response to these proteins — the different types of lipids respond differently and with different intensities. Experts hypothesize (Ingólfsson et al., 2021) that the spatial distribution of lipids correlate with types of protein constellations. Nevertheless, identifying this response or comparing the structure of these spatial patterns directly is not straightforward, especially outside the region of strong and direct influence of the proteins (about 2–9 nm). Whereas the biology community often compares patches and other similar configurations using simple criteria, such as protein constellations or mean lipid concentrations (Ingólfsson et al., 2020), these metrics are not descriptive enough to be useful here. On the other hand, a direct comparison of spatial patterns (*e.g.,* pixel-wise) is also not suitable because it does not preserve key invariances (discussed in Section 4) and is not biologically relevant. Instead, we utilize ideas from *metric learning* to combine several relevant intuitions and known constraints from experts to construct a suitable similarity metric for patches.

The practical relevance and utility of this similarity metric is that it has to be deployed for inference in the target multiscale simulations to enable automated and *in situ* ML-driven selection of a diverse set of patches from those generated by the continuously running continuum simulation. A key practical challenge is that this metric has to be developed without even knowing in advance exactly what types of patches will be generated during the multiscale simulation, and there is no opportunity for retraining. Specifically, the multiscale simulation campaign evolves a coupled system of coarse and fine scales, consuming almost 3.6 million GPU hours — a task that cannot be repeated. Prior to the campaign, only an uncoupled continuum model may be run and only for a short period. As a result, the data available to train the metric (the *"pre-campaign data"*) is expected to differ considerably from the data generated for inference during the campaign (the *"campaign data"*), although the extent and the exact type of differences are mostly speculation, since such a coupled model has never been simulated before. Therefore, generalization of the metric from training data to other relevant simulations is key to practical applicability. Here, we show that the inductive biases used in our framework enable our metric to perform well, despite substantial drifts in the overall distribution and characteristics of patches between the *pre-campaign* and the *campaign* datasets.

3 Related Work

The concept of similarity is fundamentally important in almost all scientific fields and data-driven analyses, such as medicine (Ma et al., 2019; Wei et al., 2020), security (Luo et al., 2020; Li et al., 2020b), social media mining (Liu et al., 2017; 2018), speech recognition (Bai et al., 2020; Li et al., 2020a), information retrieval (Hu et al., 2019; López-Sánchez et al., 2019), recommender systems (Li & Tang, 2020; Wu et al., 2020), and computer vision (Nguyen & De Baets, 2019; Wang et al., 2020; Zhao et al., 2020). Broadly, similarity is modeled using some kind of *metric space*, $(\mathbf{Z}, \mathrm{d}\mathbf{z})$, such that the distances, $\mathrm{d}\mathbf{z}$, capture the notion of similarity. Examples of traditional similarity metrics include L_p-norms, Mahalanobis distance, cosine distance, and correlation coefficients.

Recent advances in ML have revitalized the detection of similarity through ability to focus on hidden features that cannot be captured by straightforward metrics. For natural images, deep features have been shown to be strongly correlated to perceptual quality (Zhang et al., 2018), resulting in capturing perceptual similarity in a tractable manner. More relevant to this work, *metric learning* (Xing et al., 2003; Lu et al., 2017; Kaya & Bílge, 2019; Suárez-Díaz et al., 2020) has emerged as a powerful approach that aims to learn a metric space that captures similarity. In this context, *triplet losses* (Schultz & Joachims, 2004; Hoffer & Ailon, 2015) have shown remarkable success, particularly in face recognition and object detection problems (Schroff et al., 2015; Ge et al., 2018). Fundamentally, the triplet loss relies on pairs of similar ($\mathbf{x}^{\mathrm{a}}$ and $\mathbf{x}^{\mathrm{p}}$) and dissimilar ($\mathbf{x}^{\mathrm{a}}$ and $\mathbf{x}^{\mathrm{n}}$) examples of the training data; the network is trained to minimize the distance $(\mathrm{d}\mathbf{z})$ between learned representations $(\mathbf{z})$ of the examples from the same class and place a margin (α) between those of different classes or categories. Formally, a triplet loss is given as $\mathcal{L}_\alpha(\mathbf{x}^{\mathrm{a}}, \mathbf{x}^{\mathrm{p}}, \mathbf{x}^{\mathrm{n}}) = \max(0,\ \alpha + \mathrm{d}\mathbf{z}(\mathbf{x}^{\mathrm{a}}, \mathbf{x}^{\mathrm{p}}) - \mathrm{d}\mathbf{z}(\mathbf{x}^{\mathrm{a}}, \mathbf{x}^{\mathrm{n}}))$. There also exist many other popular formulations of loss functions that employ similar strategies, *e.g.,* contrastive (Hadsell et al., 2006) and quadruple (Chen et al., 2017) losses, but all such approaches generally require well-formulated supervision, usually in the form of class labels.

Often suitable for scientific applications, unsupervised, autoencoder-based techniques (Bhowmik et al., 2018; Bhatia et al., 2021; Jacobs et al., 2021) have also been utilized to identify similarities. Nevertheless, reconstruction-based training in such approaches implies prescient notions of properties that need to be preserved, potentially disregarding other intuitions. This is also a limitation of the only known method that addresses this problem for patches (data of our interest; see Section 2). Specifically, Bhatia et al. (2021) use a variational autoencoder to create a latent space and use it to define similarity among patches, although, their patches represent a simpler system, including only one type of protein and capturing lipid concentrations at substantially lower resolution (5×5 as compared to 37×37 in our case). These differences in data along with our requirement to obtain invariance to certain rigid transformations of interest necessitate a departure from autoencoder-type networks because reconstruction losses rely on pixel-wise comparison that directly contradicts the invariance needed, making the problem unnecessarily difficult. Instead, our metric learning approach satisfies all necessary conditions and delivers a 2130× reduction in data (37×37×14 to 9-D), whereas the autoencoder-based method of Bhatia et al. (2021) reduces the data by 23× (5×5×14 to 15-D) only.

4 Similarity Metric for Patches

In this section, we describe the several biology-informed intuitions that we leverage to define a suitable similarity metric for patches and how to morph those ideas into a metric learning framework.

Shown in Figure 1, a patch comprises a multichannel image, $\mathbf{x}(i,j,c)$, where channels $0 \leq c < 14$ represent lipid concentrations on a grid with indices $0 \leq i,j < 37$. A patch also contains a tuple, $\mathbf{y} = (y_s, y_f)$, where $y_s \geq y_f$ and $y_f \geq 0$ are the numbers of RAS and RAF proteins, respectively. RAF localizes at the membrane only in association with RAS; therefore, for each RAF, there is a RAS in the system. By construction, each patch with $y_s > 0$ is centered around a RAS protein, providing a consistent reference frame across patches. Without loss of generality and in the context of the current application, patches are broadly categorized into four *types*: no protein, ($\mathbf{y} = (0,0)$) as control, 1 RAS ($\mathbf{y} = (1,0)$), 1 RAS-RAF ($\mathbf{y} = (1,1)$), and everything else ($y_s > 1, y_f > 1$).

We pose the goal of identifying similarity among patches as a metric learning problem and employ a neural network to learn this metric. Specifically, given a patch (**x**,**y**), we build a mapping, $(\mathbf{x},\mathbf{y}) \rightarrow \mathbf{z}$, where $\mathbf{z} \in \mathbf{Z} \subset \mathbb{R}^d$, such that $(\mathbf{Z}, \mathrm{d}\mathbf{z})$ defines a metric space using Euclidean distance in $\mathbb{R}^d$, *i.e.,* $\mathrm{d}\mathbf{z}(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{z}(\mathbf{x}_i) - \mathbf{z}(\mathbf{x}_j)\|_2$. We subsequently use $(\mathbf{Z}, \mathrm{d}\mathbf{z})$ to define similarity between patches and express three relevant scientific intuitions mathematically to learn a suitable $(\mathbf{Z}, \mathrm{d}\mathbf{z})$.

Supervised classification of protein constellations. Since the application focuses on exploring lipids' response to proteins (see Section 2), a metric that learns on lipid distributions and uses protein constellations as a dependent variable is suitable to provide new insights. Therefore, given a patch (**x**,**y**), we treat $\mathbf{x}$ as *input* and $\mathbf{y}$ as a *label*, and model this intuition as a clustering problem with a *classification loss* formulated as a triplet loss, $\mathcal{L}^{\text{lab}} = \mathcal{L}_{\alpha^{\text{lab}}}(\mathbf{x}_{\mathbf{y}_i}, \mathbf{x}_{\mathbf{y}_i}, \mathbf{x}_{\mathbf{y}_j})$, where $\mathbf{y}_i$ and $\mathbf{y}_j$ are distinct labels, and α^{lab} the desired margin. More generally, this is an example of common scenarios where task-specific information are available and may be used directly to supervise the learning.

Self-supervised invariance to axis-aligned rotations and reflections. As is common in many scientific applications, the coordinate system of the simulation is arbitrary, *i.e.,* the two coordinate axes could be rotated or inverted without changing the biology simulated. A suitable metric that learns the underlying biology must, therefore, be agnostic to the specific choice of the coordinate system. Stated differently, our learned metric must be invariant to such transformations of patches. Without loss of generality, let $\rho(\mathbf{x})$ denote transformations of interest. In this work, we consider only the three $\pi/2$ rotations and horizontal and vertical reflections of patches. Arbitrary transformations are irrelevant, *e.g.,* continuous rotations either lose the corners (if image size is preserved) or introduce "empty" corners (if image is expanded), in both cases resulting in unrealistic patches.

Given ρ, we require that $\mathrm{d}\mathbf{z}(\mathbf{x}, \rho(\mathbf{x})) = 0$ for all $\mathbf{x}$, and by implication $\mathrm{d}\mathbf{z}(\mathbf{x}_1, \mathbf{x}_2) = \mathrm{d}\mathbf{z}(\mathbf{x}_1, \rho(\mathbf{x}_2))$ for all $\mathbf{x}_1$ and $\mathbf{x}_2$. We model this invariance using a hard-triplet loss to force all transformations of every patch, $\mathbf{x}_i$, closer to each other than any other pair of patches, $\mathbf{x}_i$ and $\mathbf{x}_j$ with $i \neq j$, by some margin α^{inv}. Formally, we define an *invariance loss* as $\mathcal{L}^{\text{inv}} = \mathcal{L}_{\alpha^{\text{inv}}}(\mathbf{x}_i, \rho(\mathbf{x}_i), \mathbf{x}_j)$.

Whereas in principle, two different and unrelated patches could still be exact transformations of each others, in practice, the probability of finding such a pair is virtually zero. Therefore, we assume all given patches are different and perform data augmentation in the form of online triplet mining by transforming each patch in a given training batch to its five relevant transformations. Such augmentations lead to a self-supervised approach for training the metric to identify invariance.

Feature proportionality for lipid spatial patterns. The spatial arrangement of lipids is also key to describing similarity among patches (see Figure 1 and Figure 5). To this end, we define a new feature that is inspired by radial distribution functions, which are used extensively to study such biological simulations and whose intuition is well favored by subject matter experts. This feature, a *radial profile*, is attractive because it captures similarity in lipid arrangements, helps alleviate contradictions with protein constellations, and synergizes with the required invariance.

Formally, given a patch $\mathbf{x}(i, j, c)$ and a reference pixel thereof, (i_*, j_*), we first compute a radial profile per channel as $\gamma_{(i_*,j_*)}(\mathbf{x}, r, c) = \max(\mathbf{x}(i, j, c))$, where $r = \text{round}(\sqrt{(i - i_*)^2 + (j - j_*)^2})$. We use $\max$ aggregation as it is sensitive to variations across r and, thus, is suitable to distinguish patches. Since a patch has a protein at its center of the patch (when $y_s > 0$), we compute the radial profile from the center pixel spanning the incircle of the image (*i.e.,* $r < 19$). To account for the lipids outside the incircle, we compute radial profiles from each corner for $r < 19$ but only within the corresponding quadrant (see Section A.1), and to preserve invariance to transformations of interest, the corner profiles are averaged. Both the center profile and the mean of corner profiles are $[19 \times 1]$ vectors, which are then then concatenated to give a $[38 \times 1]$ vector for each channel. In this work, we used only 8 lipid channels that the experts deemed to be more important than others. Our final radial profile feature for a patch, $\Upsilon(\mathbf{x})$ is, therefore, a $[38 \times 8]$ vector. We next define a *feature proportionality loss* that forces the metric to keep similar Υ together in the learned metric space. Specifically, we use $\mathcal{L}^{\text{rad}}(\mathbf{x}_i, \mathbf{x}_j) = |\text{d}\mathbf{z}(\mathbf{x}_i, \mathbf{x}_j) - \lambda^{\text{rad}}\, \text{d}\Upsilon(\mathbf{x}_i, \mathbf{x}_j)|$, where, $\text{d}\Upsilon(\mathbf{x}_i, \mathbf{x}_j) = \|\Upsilon(\mathbf{x}_i) - \Upsilon(\mathbf{x}_j)\|_2$ and λ^{rad} controls the span of the learned metric, $\text{d}\mathbf{z}$, with respect to the valid ranges of $\text{d}\Upsilon$.

Ultimately, the definition and use of radial profiles is a valuable heuristic that imposes inductive biases on the metric to learn and capture spatial patterns without making specific assumptions about the data and, therefore, is useful for generalization across datasets and simulations.

Combination of necessary-but-not-sufficient intuitions. Each of the three conditions discussed above may be straightforward to satisfy in isolation; however, potential contradictions between them pose significant challenges. Consider the scenario where two patches with different protein constellations exhibit remarkably similar radial profiles, yet other pairs of patches within same protein constellation classes exhibit a larger variability in spatial patterns. Given such contradictory conditions, we develop a single and consistent framework that absorbs such contradictions by aiming to organize the metric with respect to growing neighborhoods of data points, as illustrated in Figure 2.

Given a reference patch, we require all its transformations to be “really close” to it (as compared to non-transformations) and all patches with different labels to be “much farther” (as compared to the patches with the same label). The set of patches that are not transformations and have the same label are distributed in between, based on the similarity in the radial profiles. We realize this multiscale organization using a small margin for invariance, α^{inv}, a larger margin for labels, α^{lab}, and a proportionality factor that maps the range of distances between radial profiles to the desired range in the latent space, *i.e.,* $\lambda^{\text{rad}} \propto 1/(\max(\text{d}\Upsilon) - \min(\text{d}\Upsilon))$. To combine the three requirements, we use a weighted sum of the corresponding loss functions, *i.e.,* $\mathcal{L} = w^{\text{inv}}\mathcal{L}^{\text{inv}} + w^{\text{rad}}\mathcal{L}^{\text{rad}} + w^{\text{lab}}\mathcal{L}^{\text{lab}}$. Our framework allows balancing the conflicts described above through controlling the hyperparameters and the weights of the three loss functions. For instance, relatively large values of α^{lab} and w^{lab} will aim to obtain separability with respect to labels at the cost of separability in the feature space. Therefore, the targets of discovery are identifying suitable hyperparameters that support good generalizability: the weights (w^{inv}, w^{rad}, and w^{lab}), the margins (α^{inv} and α^{lab}), the ranking proportionality factor (λ^{rad}), together with the dimension (d), of the resulting metric space.

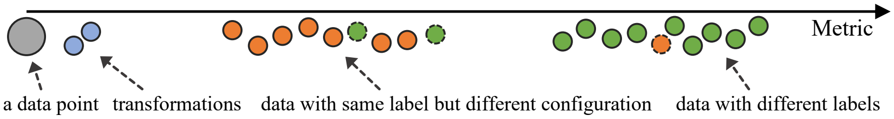
Metric
a data point
transformations
data with same label but different configuration
data with different labels

Figure 2: Our framework imposes inductive biases using biology-informed intuitions to organize neighborhoods of data points into a “multiscale hierarchy” that represent different conditions (blue, orange, and green). The “out-of-order” points (green in the orange cloud and orange in the green cloud) represent contradictions in the intuitions that are resolved by the metric.

5 Evaluation, Selection, and Utility of Our Metric

We use our framework to develop (see Section A.2) a similarity metric for patches using the *“pre-campaign”* dataset (see Section 2). A suitable model was selected by systematically evaluating models developed for ranges of relevant hyperparameters, the dimensionality of the metric space, different network architectures, and ablation studies of the three conditions of interest (see Table 1).

	Metric	Metric dimensionality (↓)	Overlap % (↓)	MARE*(↓)	AUC (↑)
Standard:	ℓ_2 ($\mathbf{x}$; images)	19,166	99.99	13.54	0.34
	ℓ_2 (μ; mean conc.)	14	0	19.64	0.32
	ℓ_2 (Υ; RDFs)	308	0	0	0.36
Ablations:	$\mathcal{L}^{\text{inv}}$	1	0.09	551.08	0.58
	$\mathcal{L}^{\text{rad}}$	9	0.06	0.98	0.30
	$\mathcal{L}^{\text{lab}}$	9	99.73	117.26	0.76
	$\mathcal{L}^{\text{rad}} + \mathcal{L}^{\text{lab}}$	9	99.99	0.26	0.68
Our metric:	$\mathcal{L}^{\text{inv}} + \mathcal{L}^{\text{rad}} + \mathcal{L}^{\text{lab}}$	9	0.05	2.6	0.61

Table 1: Comparison with benchmarks and ablations indicate that our approach provides the best balance among the criteria and maintains a low dimensionality of the metric space. This comparison uses MARE* (units of $\mathrm{d}\Upsilon$), which differs from MARE (units of $\mathrm{d}\mathbf{z}$) by a scaling factor ($\mathcal{L}^{\text{rad}}$).

5.1 Metric Evaluation and Selection

In the absence of any ground truth, we use three evaluation criteria to assess the quality of metrics learned using our method and to select a suitable model for the target application. These critera correspond to the three conditions presented in Section 4. Here, we first describe these criteria using our final, selected model that delivers a 9-D metric space and then present our model selection procedure and other models. This evaluation was done on a set of 30,000 randomly selected patches from the validation data, providing almost 450 million pairs of points that we compute distances for.

Separability of protein constellations. Figure 3 visualizes a 2-D t-SNE (van der Maaten & Hinton, 2008) of $\mathbf{z}$. As expected, the no-protein patches (type 1) form an isolated cluster, and 1-RAS and 1-RAS-RAF patches (types 2 and 3) are also relatively well separated. Nevertheless, although all other patches (type 4) appear to have a mostly-well-defined cluster, we observe notable overlap with types 2 and 3 — these are the patches where the lipid configurations (images) do not exhibit distinctive responses to protein compositions (classes), and are of substantial interest to the application. As a result, it is important to not over-penalize this model for such misclassifications, but instead strive for a balance with similarity in multichannel images. To quantify the model's classification capability, we use the *area under the precision-recall curve* (AUC) in the metric space (see Section A.3). For each validation data point, we compute these metrics for increasing numbers of neighbors and compute the precision-recall curve. For the chosen model, the AUC is 0.61.

Invariance and separability of transformations. We define an *overlap* metric to quantify a model's ability to capture invariance to transformations and separate them from any pairs of patches that are not transformations of each other. Overlap reports the proportion of points that cannot be distinguished any more than the transformations of some data points. Specifically, overlap counts pairs ($\mathbf{x}_j$, $\mathbf{x}_k$), where $\mathrm{d}\mathbf{z}(\mathbf{x}_j, \mathbf{x}_k) \leq \max(\mathrm{d}\mathbf{z}(\mathbf{x}_i, \rho(\mathbf{x}_i)))$ for all five transformations of interest of all points, $\mathbf{x}_i$. Figure 3(middle) shows the distributions of these distances with an overlap of only 0.05% and, thus, demonstrates an excellent separability of transformations by the chosen model.

Preservation of proportionality with the similarity of lipid patterns. Finally, we quantify the model's capability to preserve the distances given by the feature space (radial profiles). Figure 3(right) shows the correlation between our metric and distances in the feature space for all 450 million pairs of points in the validation dataset. The plot highlights an excellent linear correlation with an expected proportionally factor 0.1 (=λ^{rad} used for this model). This result demonstrates the model's capability to preserve the given distances (within the proportionality factor), even if at the cost of clustering quality (discussed above), thus, addressing the contradictions where needed. We

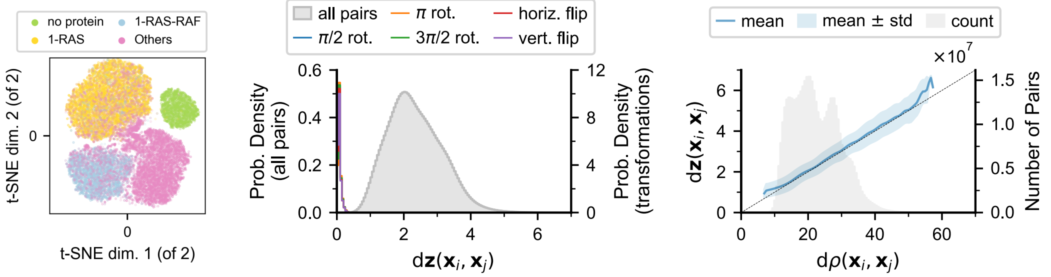

Figure 3: We evaluate our similarity metrics using three criteria: (left) high ability to separate different classes of data, using *AUC*. (middle) low *overlap* between the distances among data points and those among transformations. (right) low residual error (*MARE*) between the distances captured by the metric and those given by the feature space (within the chosen proportionality factor).

further note that the model shows low error despite the large sample size (shown in gray) in the short-range distances — which are of significant interest. We quantify this evaluation using the *median absolute residual error* (MARE) of the correlation, *i.e.,* the median of $|\mathrm{d}\mathbf{z}(\mathbf{x}_i, \mathbf{x}_j) - \lambda^{\mathrm{rad}}\mathrm{d}\Upsilon(\mathbf{x}_i, \mathbf{x}_j)|$ over all pairs of points. For the chosen model, this error is 0.26 (units of $\mathrm{d}\mathbf{z}$), which is well below the overall distribution of distances in this metric space (see Figure 3(midlde) and Figure 3(right)).

Model selection. Figure 4 summarizes our evaluation of over 100 models using a parallel coordinates plot and highlights the chosen model (black line), which provided the best balance between the quality of the model (evaluation criteria) and the dimensionality of the metric. These models represent different hyperparameters, different architectures, as well as ablations (*e.g.,* MARE $= -1$, *i.e.,* an invalid value, indicates models without the radial profile condition). Zooming further on narrow ranges of the three criteria (Figure 4(right)), we notice that several models offer good and comparable quality, providing empirical indication that our framework is generally robust. We note one 10-D model (purple) that marginally outperforms the chosen model (black) that we chose to ignore in favor of a smaller dimensionality. For the selected model, α^{inv} = 1, w^{inv} = 2, λ^{rad} = 0.1, w^{rad} = 2, α^{lab} = 8, w^{lab} = 1, and d = 9. The architecture of the chosen model is given in Section A.2.

5.2 Metric Utility and Generalizability to Distributional Shifts

The selected model was deployed on the *Summit* supercomputer as part of a large-scale workflow to facilitate the target multiscale simulation campaign (Anonymous, 2021), which ran for more than 3 months and consumed over 3.6 million GPU hours. This simulation campaign generated over 6,000,000 patches (the *"campaign"* dataset) and our metric was used for *in situ* inference to select patches for MD simulations. Here, we present retrospective analysis of our model's performance on the *campaign* dataset. First, we assess the quality of our metric visually by showing what the model considers as similar vs. dissimilar. Figure 5 shows a randomly chosen reference patch (a) with two proteins and its similarity ($\mathrm{d}\mathbf{z}$) to three other patches with same and different protein counts. When compared against thousands of other two-protein patches (irrespective of the class label), (b) and (c) are found to be most similar and most dissimilar patches, respectively. The figure also shows a patch with ten proteins to illustrate that our metric returns high $\mathrm{d}\mathbf{z}$ (low similarity) for this comparison.

Our metric performs well despite distributional shifts. Unlike the *pre-campaign* data (used for training), the *campaign* data (used for inference) is generated from a coupled model (see Section 2). To experts' surprise, the *campaign* data exhibits significantly greater extent of distributional shifts than expected (see Section A.4), emphasizing the need for generalizabity in our metric. As described above, we utilized our framework's flexibility to reduce the emphasis on the separability of protein constellations, which ultimately allowed us to support the *campaign* data, as a stronger emphasis

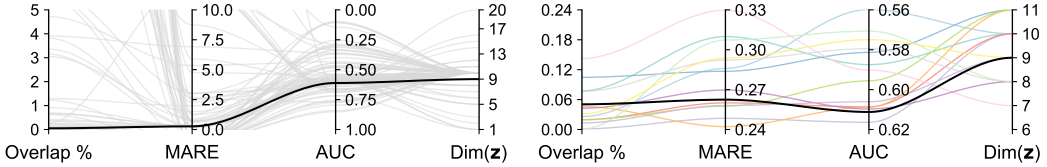

Figure 4: We evaluated over 100 models using three quantitative criteria and the dimension of the metric space. Each model is shown as a curve along the four axes of this parallel coordinates plot (lower is better on all axes). The chosen model (black) was selected by evaluating all models (left) and then focusing on a subset (right) that provided good and comparable performance.

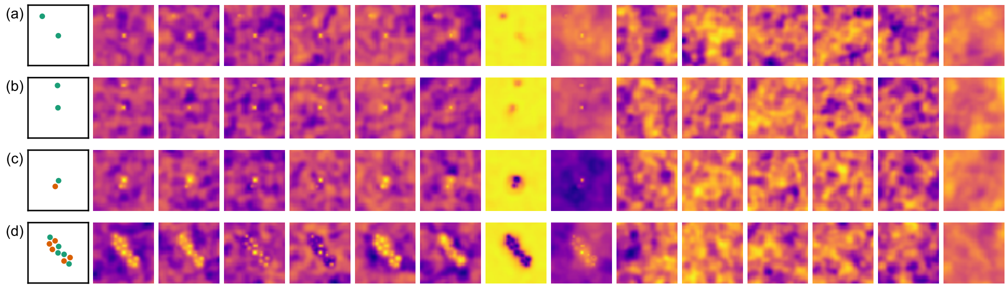

Figure 5: Visual depiction of our similarity metric for (a) a two-protein reference patch. (b) is a two-protein patch that is similar to (a); $\mathrm{d}\mathbf{z}(a, b)$ = 0.262. (c) is a two-protein patch that is dissimilar to (a); $\mathrm{d}\mathbf{z}(a, c)$ = 4.052. (d) is a ten-protein patch that is dissimilar to (a); $\mathrm{d}\mathbf{z}(a, d)$ = 7.502. Patch visualizations (rows) use the same layout as Figure 1; colormap is same for each lipid (column).

on such separability would have led to poor inference. Instead, our framework absorbed such differences by resolving them in favor of the radial profiles. We support this claim by computing our evaluation criteria on a random subset of 30,000 patches taken from the *campaign* data and note that the model provides comparable quality, with overlap = 0.01%, MARE = 0.03, and AUC = 0.59.

Our metric is superior to standard alternatives. We next evaluate the model against a standard alternative, mean lipid concentrations, μ. Here, we show the correlation between the distances in the space of $\mathrm{d}\mu$ with those given by the learned metric, $\mathrm{d}\mathbf{z}$, for the *pre-campaign* and the *campaign* data (see Figure 6) and draw attention to two observations. First, the range of the horizontal axes in both plots is the same, which is the property of the simulated system. Although one expects to find fluctuations for individual lipids, the system itself is conserved (when certain lipids deplete, others replete the space) and, hence, the net ranges of the differences remain approximately the same. Second, given considerable shifts in lipid concentrations, the campaign data produces markedly different μ vectors that our metric captures by populating the extremes (previously unpopulated regions) of the learned metric space, as indicated by the larger ranges of the similarity metric (vertical axes in the plots). Regardless, even for the *campaign* data, our metric performs well at preserving the differences in the mean concentrations for more than 78% of the data evaluated, whereas the extreme regions in the metric space (*e.g.,* $\mathrm{d}\mathbf{z} > 4$) appear to be less well understood.

Our metric understands the spatial locations of patches. Thus far, we have considered a patch as comprising only lipids and proteins. In the simulation, however, each patch has a position in space (and time). Whereas the spatial distance between patches is not a direct measure of similarity, the goal of preventing redundant simulations requires delivering high similarity for patches that overlap spatially. Figure 7 shows our similarity metric between patches that are within same time-steps of the simulation. This result demonstrates that our metric naturally understands the spatial context of patches, as it correlates spatial positions with similarity for overlapping patches, and shows no correlation between patches that are more than 15 nm (half the patch size) away.

Our metric understands decorrelation time of patches. We further demonstrate the biological relevance of our metric through the implicit connection between time and similarity founded in statistical mechanics. Specifically, there is an expectation that, given infinite time, any patch may evolve into any other one. Considering such evolution, a given patch is expected to remain considerably similar for short time periods, but become arbitrarily dissimilar for large enough time. This so-called *decorrelation time* (when patches along an evolution are not correlated anymore) is a key concept often used to guide sampling, I/O rates, and lengths of simulations. More importantly, the (shortest) time necessary for one patch to evolve into another represents a very intuitive and biologically relevant measure of similarity. However, given two arbitrary patches, directly computing this time scale is practically infeasible (an upper-bound estimate is about 500 ns, see Appendix A.6).

To evaluate our metric in this context, we performed a *post-campaign* simulation that followed 300 unique RAS proteins saved at 100× the temporal frequency used for training (*pre-campaign* data) and inference (*campaign* data), resulting in 300 histories that represent the temporal evolution of the respective patches. Figure 7 plots our similarity metric against the time difference between pairs of patches within the same history, *i.e.,* around the same RAS protein. We note a strong correlation between both concepts for hundreds of ns, flattening only around 500 ns with a marked increase in standard deviation, which matches the expected decorrelation time past which there should not exist any relationship between patches. This result demonstrates that our metric, which is trained on patches at lower temporal resolution and without any explicit information on time, naturally learns the inherent correlation between patch similarity and evolution time. Most importantly, establishing this correlation, especially for shorter time scales, enables us, for the first time, to estimate the

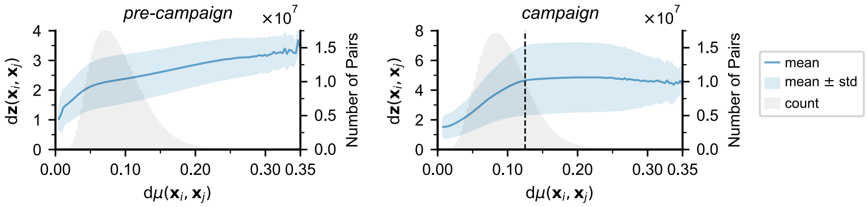

Figure 6: Our metric ($\mathrm{d}\mathbf{z}$) captures the differences in mean concentration vectors, $\mu(\mathbf{x})$, for both *pre-campaign* and *campaign* datasets. For the latter, the simulation generates many new and previously unseen configurations, leading to a wider exploration in our metric space, resulting in a larger range of $\mathrm{d}\mathbf{z}$. Yet, the metric performs well on more than 78% of the data (left of vertical line).

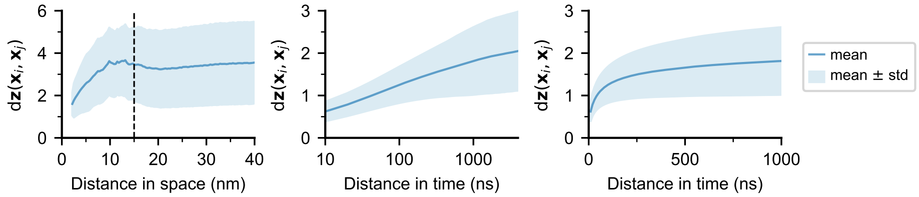

Figure 7: Despite trained agnostic to spatiotemporal locations of patches, our model captures inherent relationships with respect to both space and time. (left) As expected, patches with little to no overlap (>15 nm apart) show no correlation with our metric. (middle and right) Our metric captures the high similarity existent at small temporal neighbhoroods and also hints at the decorrelation time in the simulation, after which, patches are considered equally dissimilar in statistical mechanics.

evolution time as a factor of the distance in the metric space. These insights will enable an entirely new viewpoint for the analysis of the resulting simulation by directly correlating observed properties of the membrane or the proteins with respect to their estimated difference in time.

6 Conclusion and Discussion

We present a framework to compute similarities among data generated by a computational biology application crucial to cancer research. We address two key challenges. First, since complex scientific applications, such as the one presented here, work at the cutting edge of contemporary knowledge, well-formulated criteria for similarity are usually not available. In the absence of straightforward metrics, we instead utilize biology-informed criteria gathered from experts and cast them into a metric learning framework to learn a meaningful similarity metric. We show that our metric fuses these necessary-but-not-sufficient conditions well and is robust to potential contradictions between them. Through close collaboration with subject matter experts (*e.g.,* to identify relevant conditions and suitable features), our framework turns the lack of well-defined similarity criteria into a strength by imposing the experts' biology-informed intuitions as inductive biases that allow the metric to generalize to new simulations exhibiting significant distributional shifts in data — addressing the second challenge of deploying the model for *in situ* inference on unseen data. We demonstrate this generalization on two new datasets and show that our model learns key behavior of interest in the simulations, rather than focusing on the specific datasets themselves.

Our framework and the resulting similarity metric has been deployed as the key driving component in the first-of-its-kind multiscale simulation campaign (Anonymous, 2021) to explore RAS-RAF-membrane interactions, with a potential for significant impact in cancer biology (Anonymous, 2022a;b). Unfortunately, the massive scale of such simulations (3–4 months, consuming 3–5 million GPU hours) makes it computationally infeasible to rerun the simulations to compare different metrics, models, techniques, or benchmarks, necessitating evaluation and comparisons in a proxy or development setting (Table 1 and Figure 4). Through suitable evaluation metrics and validation of external criteria (*e.g.,* decorrelation time), this manuscript also addresses the challenges of a lack of ground truth to compare against and demonstrates a meaningful metric until the biology community can test more hypotheses and develop a more widely-accepted understanding of similarity. By showing a direct relationship between distance in the metric space and the expected evolution time between patches, our work also leads to new insights and opens up new directions of research.

Our data and experiments highlight that realistic simulation are not restricted to small distributional shifts. Therefore, despite our demonstration of generalizability, an opportunity for improvement lies in updating the model *in situ* using new data from the running simulations. As such, we will explore challenges, both computational and fundamental, associated with automatic detection of a model's suitability and online learning approaches to absorb new data to update the model.

Finally, although our work is presented in a specific application context, our framework is broadly applicable to other applications, such as experimental design and other types of autonomous multiscale simulations, that face such challenges. Specifically, our framework can be easily adapted to support different types of simulations in the space of biology or elsewhere using intuitions from the application. As examples, we have since customized our framework to support a different biological system where patches are periodic in both spatial dimensions, and we are also currently applying this framework to different biological systems with GPCR proteins (Rosenbaum et al., 2009).

REFERENCES

Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pp. 265–283, 2016.

Anonymous. Generalizable coordination of large multiscale workflows: Challenges and learnings at scale. In *Proc. of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '21, New York, NY, USA, 2021. ACM. To appear.

Anonymous. Illuminating the interactions of RAS-RAF complex with cell membrane, 2022a. In preparation.

Anonymous. Breeching scales with MuMMI, A next-generation simulation infrastructure, 2022b. In preparation.

Gary S. Ayton and Gregory A. Voth. Multiscale simulation of protein mediated membrane remodeling. *Seminars in Cell & Developmental Biology*, 21(4):357–362, June 2010. ISSN 10849521. doi: 10.1016/j.semcdb.2009.11.011.

Zhongxin Bai, Xiao-Lei Zhang, and Jingdong Chen. Speaker verification by partial auc optimization with mahalanobis distance metric learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:1533–1548, 2020. doi: 10.1109/TASLP.2020.2990275.

Harsh Bhatia, Timothy S. Carpenter, Helgi I. Ingólfsson, Gautham Dharuman, Piyush Karande, Shusen Liu, Tomas Oppelstrup, Chris Neale, Felice C. Lightstone, Brian Van Essen, James N. Glosli, and Peer-Timo Bremer. Machine learning based dynamic-importance sampling for adaptive multiscale simulations. *Nature Machine Intelligence*, 3:401–409, 2021. doi: 10.1038/s42256-021-00327-w.

Debsindhu Bhowmik, Shang Gao, Michael T. Young, and Arvind Ramanathan. Deep clustering of protein folding simulations. *BMC Bioinformatics*, 19, 2018. doi: 10.1186/s12859-018-2507-5.

W. Chen, X. Chen, J. Zhang, and K. Huang. Beyond triplet loss: A deep quadruplet network for person re-identification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1320–1329, Los Alamitos, CA, USA, jul 2017. IEEE Computer Society. doi: 10.1109/CVPR.2017.145.

François Chollet et al. Keras. `https://github.com/fchollet/keras`, 2015.

Giray Enkavi, Matti Javanainen, Waldemar Kulig, Tomasz Róg, and Ilpo Vattulainen. Multiscale Simulations of Biological Membranes: The Challenge To Understand Biological Phenomena in a Living Substance. *Chemical Reviews*, March 2019. ISSN 0009-2665, 1520-6890. doi: 10.1021/acs.chemrev.8b00538.

Weifeng Ge, Weilin Huang, Dengke Dong, and Matthew R. Scott. Deep metric learning with hierarchical triplet loss. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (eds.), *Computer Vision – ECCV 2018*, pp. 272–288, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01231-1.

R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pp. 1735–1742, 2006. doi: 10.1109/CVPR.2006.100.

Alfons Hoekstra, Bastien Chopard, and Peter Coveney. Multiscale modelling and simulation: A position paper. *Philosophical Transactions of The Royal Society A*, 372:20130377, 2014. doi: 10.1098/rsta.2013.0377.

Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In Aasa Feragen, Marcello Pelillo, and Marco Loog (eds.), *Similarity-Based Pattern Recognition*, pp. 84–92, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24261-3.

Haifeng Hu, Kun Wang, Chenggang Lv, Jiansheng Wu, and Zhen Yang. Semi-supervised metric learning-based anchor graph hashing for large-scale image retrieval. *Trans. Img. Proc.*, 28(2): 739–754, February 2019. ISSN 1057-7149. doi: 10.1109/TIP.2018.2860898.

Helgi I. Ingólfsson, Timothy S. Carpenter, Harsh Bhatia, Peer-Timo Bremer, Siewert J. Marrink, and Felice C. Lightstone. Computational Lipidomics of the Neuronal Plasma Membrane. *Biophysical Journal*, 113(10):2271–2280, November 2017. ISSN 00063495. doi: 10.1016/j.bpj.2017.10.017.

Helgi I. Ingólfsson, Harsh Bhatia, Talia Zeppelin, W. F. Drew Bennett, Kristy A. Carpenter, Pin-Chia Hsu, Gautham Dharuman, Peer-Timo Bremer, Birgit Schiøtt, Felice C. Lightstone, and Timothy S. Carpenter. Capturing biologically complex tissue-specific membranes at different levels of compositional complexity. *The Journal of Physical Chemistry B*, 124(36):7819–7829, 2020. doi: 10.1021/acs.jpcb.0c03368. PMID: 32790367.

Helgi I. Ingólfsson, Chris Neale, Timothy S. Carpenter, Rebika Shrestha, Cesar A López, Timothy H. Tran, Tomas Oppelstrup, Harsh Bhatia, Liam G. Stanton, Xiaohua Zhang, Shiv Sundram, Francesco Di Natale, Animesh Agarwal, Gautham Dharuman, Sara I. L. Kokkila Schumacher, Thomas Turbyville, Gulcin Gulten, Que N. Van, Debanjan Goswami, Frantz Jean-Francios, Constance Agamasu, De Chen, Jeevapani J. Hettige, Timothy Travers, Sumantra Sarkar, Michael P. Surh, Yue Yang, Adam Moody, Shusen Liu, Angel E. García, Brian C. Van Essen, Arthur F. Voter, Arvind Ramanathan, Nicolas W. Hengartner, Dhirendra K. Simanshu, Andrew G. Stephen, Peer-Timo Bremer, S. Gnanakaran, James N. Glosli, Felice C. Lightstone, Frank McCormick, Dwight V. Nissley, and Frederick H. Streitz. Machine Learning-driven Multiscale Modeling Reveals Lipid-Dependent Dynamics of RAS Signaling Proteins. *Proceedings of the National Academy of Sciences (PNAS)*, 2021. doi: 10.21203/rs.3.rs-50842/v1. Preprint.

Sam Ade Jacobs, Tim Moon, Kevin McLoughlin, Derek Jones, David Hysom, Dong H Ahn, John Gyllenhaal, Pythagoras Watson, Felice C Lightstone, Jonathan E Allen, Ian Karlin, and Brian Van Essen. Enabling rapid covid-19 small molecule drug design through scalable deep learning of generative models. *The International Journal of High Performance Computing Applications*, 35 (5):469–482, 2021. doi: 10.1177/10943420211010930.

Mahmut Kaya and Hasan Şakir Bílge. Deep metric learning: A survey. *Symmetry*, 11(9), 2019. ISSN 2073-8994. doi: 10.3390/sym11091066.

Dirk Kessler, Michael Gmachl, Andreas Mantoulidis, Laetitia J. Martin, Andreas Zoephel, Moriz Mayer, Andreas Gollner, David Covini, Silke Fischer, Thomas Gerstberger, Teresa Gmaschitz, Craig Goodwin, Peter Greb, Daniela Häring, Wolfgang Hela, Johann Hoffmann, Jale Karolyi-Oezguer, Petr Knesl, Stefan Kornigg, Manfred Koegl, Roland Kousek, Lyne Lamarre, Franziska Moser, Silvia Munico-Martinez, Christoph Peinsipp, Jason Phan, Jörg Rinnenthal, Jiqing Sai, Christian Salamon, Yvonne Scherbantin, Katharina Schipany, Renate Schnitzer, Andreas Schrenk, Bernadette Sharps, Gabriella Siszler, Qi Sun, Alex Waterson, Bernhard Wolkerstorfer, Markus Zeeb, Mark Pearson, Stephen W. Fesik, and Darryl B. McConnell. Drugging an undruggable pocket on kras. *Proceedings of the National Academy of Sciences*, 116(32):15823–15829, 2019. ISSN 0027-8424.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

V.V. Krzhizhanovskaya, D. Groen, B. Bozak, and A.G. Hoekstra. Multiscale modelling and simulation workshop: 12 years of inspiration. *Procedia Computer Science*, 51:1082–1087, 2015. ISSN 1877-0509. doi: 10.1016/j.procs.2015.05.268. International Conference On Computational Science, ICCS 2015.

Ruirui Li, Jyun-Yu Jiang, Jiahao Liu Li, Chu-Cheng Hsieh, and Wei Wang. Automatic speaker recognition with limited data. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, WSDM '20, pp. 340–348, New York, NY, USA, 2020a. Association for Computing Machinery. ISBN 9781450368223. doi: 10.1145/3336191.3371802.

Tie Li, Gang Kou, and Yi Peng. Improving malicious urls detection via feature engineering: Linear and nonlinear space transformation methods. *Information Systems*, 91:101494, 2020b. ISSN 0306-4379. doi: https://doi.org/10.1016/j.is.2020.101494.

Xiaotong Li and Yan Tang. A social recommendation based on metric learning and network embedding. In *2020 IEEE 5th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA)*, pp. 55–60, 2020. doi: 10.1109/ICCCBDA49378.2020.9095610.

Yang Liu, Zhonglei Gu, Tobey H. Ko, and Jiming Liu. Multi-modal media retrieval via distance metric learning for potential customer discovery. In *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pp. 310–317, 2018. doi: 10.1109/WI.2018.00-75.

Yufei Liu, Dechang Pi, and Lin Cui. Metric learning combining with boosting for user distance measure in multiple social networks. *IEEE Access*, 5:19342–19351, 2017. doi: 10.1109/ACCESS.2017.2756102.

Daniel López-Sánchez, Angélica González Arrieta, and Juan M. Corchado. Visual content-based web page categorization with deep transfer learning and metric learning. *Neurocomputing*, 338: 418–431, 2019. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2018.08.086.

Jiwen Lu, Junlin Hu, and Jie Zhou. Deep metric learning for visual understanding: An overview of recent advances. *IEEE Signal Processing Magazine*, 34(6):76–84, 2017. doi: 10.1109/MSP.2017.2732900.

Yong Luo, Han Hu, Yonggang Wen, and Dacheng Tao. Transforming device fingerprinting for wireless security via online multitask metric learning. *IEEE Internet of Things Journal*, 7(1): 208–219, 2020. doi: 10.1109/JIOT.2019.2946500.

Zongqing Ma, Shuang Zhou, Xi Wu, Heye Zhang, Weijie Yan, Shanhui Sun, and Jiliu Zhou. Nasopharyngeal carcinoma segmentation based on enhanced convolutional neural networks using multi-modal metric learning. *Physics in Medicine & Biology*, 64(2):025005, jan 2019. doi: 10.1088/1361-6560/aaf5da.

Bac Nguyen and Bernard De Baets. Kernel distance metric learning using pairwise constraints for person re-identification. *IEEE Transactions on Image Processing*, 28(2):589–600, 2019. doi: 10.1109/TIP.2018.2870941.

Ian A. Prior, Fiona E. Hood, and James L. Hartley. The frequency of ras mutations in cancer. *Cancer Research*, 80(14):2969–2974, 2020. ISSN 0008-5472.

Daniel M. Rosenbaum, Søren G. F. Rasmussen, and Brian K. Kobilka. The structure and function of G-protein-coupled receptors. *Nature*, 459(7245):356–63, May 2009. doi: 10.1038/nature08144.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815–823, 2015. doi: 10.1109/CVPR.2015.7298682.

Matthew Schultz and Thorsten Joachims. Learning a distance metric from relative comparisons. In *Advances in Neural Information Processing Systems*, volume 16, pp. 41–48. MIT Press, 2004.

Dhirendra K. Simanshu, Dwight V. Nissley, and Frank McCormick. RAS Proteins and Their Regulators in Human Disease. *Cell*, 170(1):17–33, June 2017. doi: 10.1016/j.cell.2017.06.009.

Juan Luis Suárez-Díaz, Salvador García, and Francisco Herrera. A tutorial on distance metric learning: Mathematical foundations, algorithms, experimental analysis, prospects and challenges (with appendices on mathematical background and detailed algorithms explanation), 2020.

Timothy Travers, Cesar A. López, Que N. Van, Chris Neale, Marco Tonelli, Andrew G. Stephen, and Sandrasegaram Gnanakaran. Molecular recognition of RAS/RAF complex at the membrane: Role of RAF cysteine-rich domain. *Scientific Reports*, 8(1):8461, May 2018. ISSN 2045-2322. doi: 10.1038/s41598-018-26832-4.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

Gregory A. Voth. A Multiscale Description of Biomolecular Active Matter: The Chemistry Underlying Many Life Processes. *Accounts of Chemical Research*, 50(3):594–598, March 2017. ISSN 0001-4842. doi: 10.1021/acs.accounts.6b00572.

Chen Wang, Guohua Peng, and Bernard De Baets. Deep feature fusion through adaptive discriminative metric learning for scene recognition. *Information Fusion*, 63:1–12, 2020. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2020.05.005.

Andrew M. Waters and Channing J. Der. KRAS: The Critical Driver and Therapeutic Target for Pancreatic Cancer. *Cold Spring Harbor Perspectives in Medicine*, 8(9):a031435, September 2018. ISSN 2157-1422. doi: 10.1101/cshperspect.a031435.

Guohui Wei, Min Qiu, Kuixing Zhang, Ming Li, Dejian Wei, Yanjun Li, Peiyu Liu, Hui Cao, Mengmeng Xing, and Feng Yang. A multi-feature image retrieval scheme for pulmonary nodule diagnosis. *Medicine*, 99(4):e18724, 2020. doi: 10.1097/MD.0000000000018724.

Hao Wu, Qimin Zhou, Rencan Nie, and Jinde Cao. Effective metric learning with co-occurrence embedding for collaborative recommendations. *Neural Networks*, 124:308–318, 2020. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2020.01.021.

Eric Xing, Andrew Y Ng, Michael I Jordan, and Stuart J Russell. Distance metric learning with application to clustering with side-information. In S. Becker, S. Thrun, and K. Obermayer (eds.), *Advances in Neural Information Processing Systems*, volume 15, pp. 521–528. MIT Press, 2003.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Cairong Zhao, Xuekuan Wang, Wangmeng Zuo, Fumin Shen, Ling Shao, and Duoqian Miao. Similarity learning with joint transfer constraints for person re-identification. *Pattern Recognition*, 97: 107014, 2020. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2019.107014.

A Appendix

A.1 Computation of Radial Profiles

To compute radial profiles efficiently, we use predefined spatial kernels (see Figure 8) that represent distance r from the reference pixel (i_*, j_*) (shown in red) rounded to the nearest integer; each kernel is defined for $r < 19$ pixels. For a given channel c within a patch $\mathbf{x}$, a radial aggregation $(\max)$ is applied using these kernels to create a 1-D radial profile, $\gamma_{(i_*,j_*)}(\mathbf{x}, r, c)$. Each profile is represented as a $[19 \times 1]$ vector. To preserve invariance to transformations, the four corner profiles are averaged and concatenated to the center profile, resulting in a $[38 \times 1]$ vector. Figure 9 shows these vectors for each channel. Finally, in this work, we use the profiles of eight most-relevant channels are appended to create Υ, which is a feature of size $[38 \times 8]$

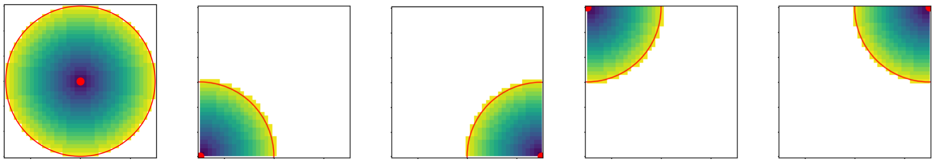

Figure 8: Spatial kernels for creating radial profiles. From left to right, kernels for center radial profile, $\gamma_{(18,18)}$ and the four corner profiles, $\gamma_{(0,0)}$, $\gamma_{(36,0)}$, $\gamma_{(0,36)}$, and $\gamma_{(36,36)}$ are shown.

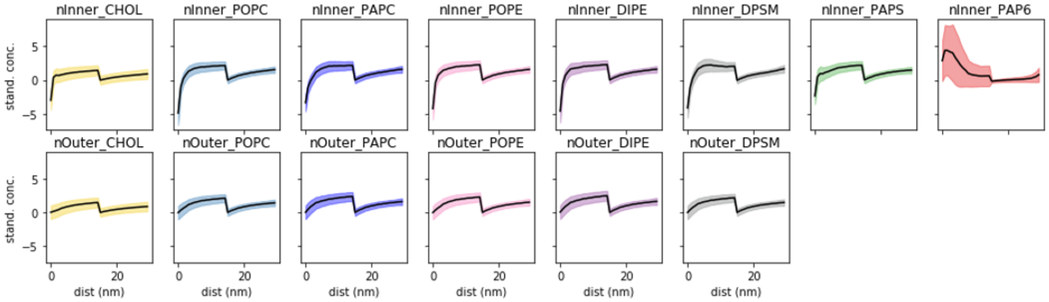

Figure 9: Mean (black) and standard deviation (color) of radial profiles for patches in the training dataset. The profiles are computed after the standardization of channels.

A.2 Model Development

A set of 334,000 patches was generated through a small and uncoupled continuum simulation run prior to the multiscale simulation campaign. This *pre-campaign* dataset was made available to us for developing the ML model. We used this dataset to train and evaluate several models as well as comparison against benchmarks (described in Table 1 and Figure 4).

The multichannel images were standardized per channel (to zero mean and unit standard deviation) across the dataset to account for differences in ranges of different lipid concentrations. Labels (for protein constellations) and radial profiles were precomputed and saved as auxiliary information for model training. A 90%-10% random split of the dataset was used for training and validation.

All models were developed with TensorFlow v2.1 (Abadi et al., 2016) and Keras v2.2.4 (Chollet et al., 2015) using one NVIDIA Volta V100 GPU each. Models were optimized using the Adam optimizer (Kingma & Ba, 2017), and most models used a piecewise-constant learning rate decay ([1, 5, 0.1, 0.01]$\times 10^{-3}$ switched after 10, 20, and 30 epochs). Training was performed until the total loss appeared converged, which took 60–100 epochs (4–6 hours of walltime).

Whereas our framework appears generally robust to architecture changes, the chosen architecture) gave superior results. Empirical evidence suggests that a separable convolution layer to account for correlations across lipid channels unsurprisingly improves the model performance. The architecture of the chosen model is as follows.

```
x → SeparableConv2D(filters=6, depth_mult=6, kernel_sz=1, strides=1,
relu) → BatchNorm → Conv2D(filters=16, kernel_sz=3, strides=2, relu) →
BatchNorm → Conv2D(filters=16, kernel_sz=3, strides=2, relu) → BatchNorm
→ Flatten → Dense(shape=9) → BatchNorm → z.
```

A.3 Precision-Recall Study for Classification of Class Labels

To supplement the discussion in Section 5.1 and Figure 3, we present the precision-recall curve as well as accuracy of classification. To measure these quantities, we consider each point in the evaluation dataset (30,000 points), and note the types of patches found within neighborhoods as the number of neighbors are increased. Using the number of false positives and false negatives, we compute accuracy, precision, and recall using the standard formulations. Figure 10 shows the variations in these quantities with respect to the size of the neighborhoods. In this work, we are mostly interested in short-range neighborhoods and note that the accuracy remains high (greater than about 75%) for up to about 12,000 neighbors. The figure also shows precision-recall curve, whose area under the curve is used for quantitative evaluation of our metric.

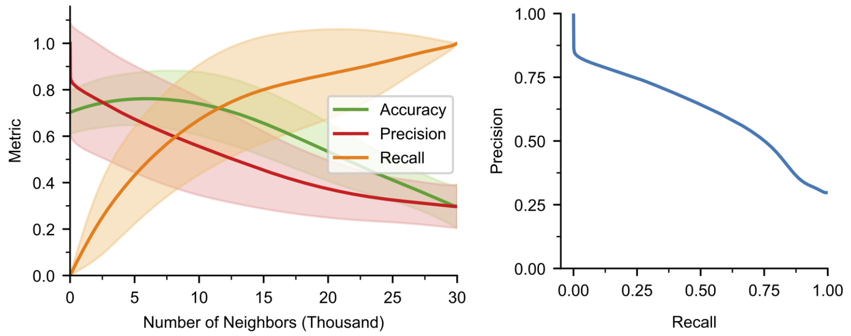

Figure 10: Accuracy, precision, and recall for our selected metric (left) with respect to the patch type and increasing neighborhoods in the metric space. We use the area under the precision-recall curve (right) to evaluate the metric.

A.4 Distributional Shifts in Data: *Pre-Campaign* vs. *Campaign* Datesets

As described in Section 2, the subject matters expect to explore significantly different types of patches during the simulation campaign than what may be modeled prior to the campaign. Although the continuum simulation during the campaign starts with the same parameters and models as available for pre-campaign, the primary reason of these anticipated differences is the evolution of a coupled multiscale system during the campaign. In particular, during the multiscale campaign, all of the several hundred thousand MD (fine scale) simulations are analyzed *in situ* and the resulting analysis is aggregated and used to improve the parameters of the continuum (coarse scale) model. As a result, the continuum model evolves and is expected to start exploring different regions of the phase space of patches. This *active feedback loop* is a key characteristic of such autonomous multiscale simulations (Ingólfsson et al., 2021).

In this work, the MD simulations indicate a higher degree of protein aggregation than previously hypothesized, which also leads to significant differences in lipid accumulation around the proteins. These shifts (highlighted in Figure 11) can be consequential to any ML model since inference has to be made on patches that the model has not seen before. For example, inference in a patch that contains around 10 proteins, whereas training data contains patches with only up to 4 or 5 proteins.

One of the key challenges in our model development was to guard against such expected yet not fully understood differences between the training and inference datasets, necessitating focusing on biology-informed inductive biases, rather than tailoring specifically to the data at hand.

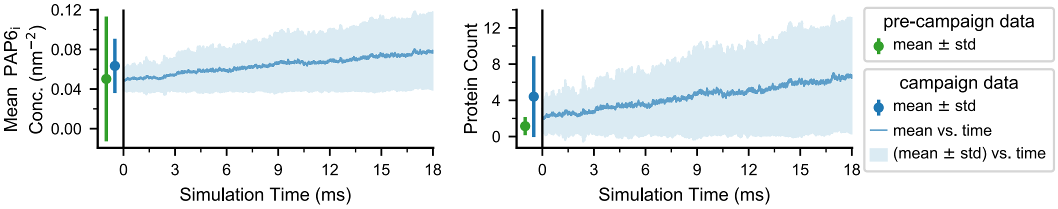

Figure 11: The distributional shifts observed between the *pre-campaign* data (used for training) and the *campaign* data (used for inference) highlights the need for generalizability of the similarity metric. The shifts become more drastic as the *campaign* simulation progresses in time.

A.5 Ablation Studies

As highlighted in Table 1, the different metrics optimize for individual criterion, *e.g.,* the RDF-related metrics provide low overlap and low MARE* but perform poorly on AUC. On the other hand, simply combining two of the metrics does not necessary give additive benefits (*e.g.,* $\mathcal{L}^{\text{rad}}$ and $\mathcal{L}^{\text{lab}}$). We note that our framework therefore produces a metric that is more useful than the sum of its parts and allows finding the right balance between the criteria. Beyond the quantitative evaluation (overlap, MARE*, and AUC), we also note the role of dimensionality of the corresponding space for computational efficiency. During the campaign, several thousand of distance computations (for nearest neighbor queries) are to be made in real- time. Therefore, a large dimensionality makes the approach infeasible.

A.6 Derivation of Decorrelation Time

The coarse-scale, continuum simulation, from which patches are collected, computes the time evolution of the membrane system. Patches that evolve from times t_1 through t_2 ($t_1 \leq t_2$) are physically very similar if $t_2 - t_1$ is small, and become arbitrarily dissimilar as $t_2 - t_1$ goes to infinity. This presents an opportunity to verify that the metric indeed shows decreased similarity as a function of $t_2 - t_1$. Here we derive an estimate of how long time it takes for a patch to get decorrelated.

In the continuum model, diffusion of lipids is a major source of entropy production and, thus, decorrelation. Given the system size and diffusivity, we can estimate the decorrelation time using the diffusion equation:

$$\frac{\partial c}{\partial t} = D\left(\frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2}\right),$$

where $c = c(x, y, t)$ is the lipid distribution (for some lipid species) and D is the corresponding diffusion coefficient. Given an $L \times L$ patch (for our data, L = 30 nm) and a standard value of D = 43.36 nm^2/µs (taken as average over several relevant lipids). The slowest decaying Fourier mode of the diffusion equation above decays as $\exp\left[-D\left(\frac{2\pi}{L}\right)^2 t\right]$, where using the values above we get $D\left(\frac{2\pi}{L}\right)^2 \approx$ 0.53 µs.

The motion of the protein and inherent randomness (noise) in the lipid evolution equations introduce further entropy and thus shortens the decorrelation time compared to the above estimate. We can see that Figure 7 shows a decorrelation time roughly similar to this diffusive estimate.