nature reviews physics

https://doi.org/10.1038/s42254-024-00791-4

Supplementary information

Roadmap on machine learning glassy dynamics

In the format provided by
the authors and unedited

Supplementary Information: Roadmap on machine learning glassy dynamics

Gerhard Jung,[1,2] Rinske M. Alkemade,[3] Victor Bapst,[4] Daniele Coslovich,[5] Laura Filion,[3] François P. Landes,[6] Andrea Liu,[7,8] Francesco Saverio Pezzicoli,[6] Hayato Shiba,[9] Giovanni Volpe,[10] Francesco Zamponi,[11] Ludovic Berthier,[1,12,*] and Giulio Biroli[11]

[1] *Laboratoire Charles Coulomb (L2C), Université de Montpellier, CNRS, 34095 Montpellier, France*
[2] *Laboratoire Interdisciplinaire de Physique (LIPhy), Université Grenoble Alpes, 38402 Saint-Martin-d'Hères, France*
[3] *Soft Condensed Matter and Biophysics, Debye Institute for Nanomaterials Science, Utrecht University, Utrecht, Netherlands*
[4] *GoogleDeepMind, London, UK*
[5] *Dipartimento di Fisica, Università di Trieste, Strada Costiera 11, 34151, Trieste, Italy*
[6] *Université Paris-Saclay, CNRS, INRIA, Laboratoire Interdisciplinaire des Sciences du Numérique, TAU team, 91190 Gif-sur-Yvette, France*
[7] *Department of Physics and Astronomy, University of Pennsylvania, Philadelphia, PA 19104, USA*
[8] *Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM 87501, USA*
[9] *Graduate School of Information Science, University of Hyogo, Kobe 650-0047, Japan*
[10] *Department of Physics, University of Gothenburg, Origovägen 6B, Gothenburg 41296, Sweden*
[11] *Laboratoire de Physique de l'Ecole Normale Supérieure, ENS, Université PSL, CNRS, Sorbonne Université, Université de Paris, F-75005 Paris, France*
[12] *Gulliver, UMR CNRS 7083, ESPCI Paris, PSL Research University, 75005 Paris, France*

In this supplemental material (SM) we provide the technical details necessary to reproduce the data shown in the benchmarking section V of the main manuscript. Additionally, we present further benchmarking.

I. SYSTEMS AND METHODS

A. KA system (3D)

Most benchmarks were performed for the Kob-Andersen binary mixture (KA) in three dimensions [1]. The basic potential used in KA is the Lennard-Jones potential,

$$V_{\alpha\beta}(r_{ij}) = \begin{cases} 4\epsilon_{\alpha\beta}\left[\left(\frac{\sigma_{\alpha\beta}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{\alpha\beta}}{r_{ij}}\right)^{6}\right] & r_{ij} < r_{\alpha\beta}^{\rm cut} \\ 0 & \text{otherwise,} \end{cases}$$

where $r_{ij} = |\boldsymbol{r}_i - \boldsymbol{r}_j|$ is the distance between particles i and j, of type α and β, respectively. The KA mixture is characterized by its non-additive interactions, $\epsilon_{11} = 1.0$, $\epsilon_{12} = 1.5$, $\epsilon_{22} = 0.5$ and $\sigma_{11} = 1.0$, $\sigma_{12} = 0.8$, $\sigma_{22} = 0.88$, between the two particles types. The cutoff is $r_{\alpha\beta}^{\rm cut} = 2.5\sigma_{\alpha\beta}$.

All details on the specific KA mixture and the dataset used in this roadmap are given in Ref. [2]. The system, for example, includes additional contributions in the potential to make it continuous up to the second derivative. The relaxation dynamics of each structure has been simulated using $N_R = 32$ independent replicas to calculate the isoconfigurational average. All results have been extracted by solely considering type 1 particles. The networks have been trained on $N_S = 400$ structures, with $N_1 = 3277$ type 1 particles each, resulting in a total of $N_d \approx 10^6$ data points. The testset consists of $N_S = 100$ independent structures.

B. KA2D system

The two-dimensional KA2D system is a variation of the above described KA mixture. In addition to the change of dimensionality, KA2D consists of three particle types, instead of two. This change enables to apply the swap Monte Carlo algorithm [3, 4] and thus go to very low temperatures [5]. All details on the KA2D system are provided in the main text and the SM of Ref. [5].

The isoconfigurational average has been calculated over $N_R = 20$ independent replicas, and $N_S = 1500$ structures are provided for training. Each structure consists of 600 type 1 particles, which thus also yields a total of $N_d \approx 10^6$ data points for training.

For both the KA and the KA2D system, the structural relaxation time τ_α is defined as,

$$F_s(\tau_\alpha) = e^{-1}, \tag{1}$$

where $F_s(t) = \langle \sum_i \cos(q\Delta r_i(t)) \rangle$ is the incoherent intermediate scattering function with $q \approx 2\pi/\sigma$ and $\Delta r_i(t) = |\boldsymbol{r}_i(t) - \boldsymbol{r}_i(0)|$. The structural relaxation time τ_α therefore describes the time scale on which the particles move on average approximately one particle diameter σ.

Equilibrated structures and labels calculated from the relaxation dynamics for both KA and KA2D will be uploaded in combination with this roadmap for future research (see 10.5281/zenodo.10118191).

* ludovic.berthier@espci.fr

C. Propensity (labels)

The input of each ML method are the positions of each particle in a given structure or derived structural descriptors. The labels that are used for training and testing are dynamical quantities characterizing structural relaxation at various time scales t. As introduced in Sec. III of the main text we use propensities to remove any effects of the initial velocities.

For each figure in the main text, except Fig. 5, the results are based on the propensities of displacement, $\mathcal{R}_i = \langle \Delta r_i(t) \rangle_{\rm iso} = N_R^{-1} \sum_{k=1}^{N_R} \left| \boldsymbol{r}_i^k(t) - \boldsymbol{r}_i(0) \right|$, where $\boldsymbol{r}_i^k(t)$ denotes the position of particle i in replica k. In Fig. 5 we have reported results for the bond-breaking propensity of particle i, $\mathcal{C}_B^i(t) = \langle n_t^i / n_0^i \rangle_{\rm iso}$. Here, n_t^i describes the number of original nearest neighbours particle i still has after a time t and n_0^i is the initial number of neighbours [6]. The quantity therefore decays from $\mathcal{C}_B^i(t=0) = 1$ to $\mathcal{C}_B^i(t \to \infty) = 0$. Details and parameters are described in the SM of Ref. [5].

D. BOTAN

The model architecture of BOTAN [2] is based on that of the original GNN proposed by Bapst *et al.* [7]. The difference is that BOTAN is equipped with an additional MLP layer as a decoder for targets on graph edges. BOTAN is implemented as an "interaction network" [8] consisting of a pair of two-layer MLPs with the size of $(64, 64)$. Each of the MLPs is connected to nodes and edges, respectively, and performs message passing with the other.

The input graphs are constructed by choosing each particle as a node of the graph, and "connecting" nearest-neighbour pairs by edges. The threshold distance for the nearest neighbours are set to $2.0\sigma_{11}$ (both for KA and 2DKA). For recursive message passing, the MLPs repeat the message passing iteratively 7 (KA) or 8 (2DKA) times, but the results are not strongly affected by slight changes in this number. The target quantity on the edges is, similarly to Ref. [2], the isoconfigurational average of distance change between particle pairs which are connected via the edges. The loss function is defined as $\mathcal{L} = 0.4\mathcal{L}_\mathcal{N} + 0.6\mathcal{L}_\mathcal{E}$, where $\mathcal{L}_\mathcal{N}$ and $\mathcal{L}_\mathcal{E}$ denote the MSE losses for node target (particle propensity) and edge target (propensity for pair-distance change), respectively. Details and visualisation of the method are presented in Ref. [2].

The BOTAN GNN model contains about 54,200 weight parameters (both for 2D and 3D). The training is started by using pretrained model parameters (optimized to yield the edge target quantities in advance), and is performed over 1,000 epochs on 400 snapshots (KA) and over 2,000 epochs on 1,500 snapshots (2DKA), respectively. The batch size is fixed to one graph. The training has been conducted separately for each temperature and time, by using the Adam optimizer with learning rates of 10^{-4} (KA) and 2×10^{-4} (2DKA). The overall training time amounted to 4 hours (KA) and 11.5 hours (2DKA) on an NVIDIA A100 Tensor Core GPU (40GB SXM) on Aquarius subsystem of Wisteria/BDEC-01 Supercomputer at Information Technology Center, University of Tokyo, with the host CPUs being 2 Intel Xeon Platinum 8360Y. These training times do not include the elapsed time of the data loader. The code and pretrained model parameters for 3D are provided on a GitHub repository (`https://github.com/h3-Open-BDEC/pyg_botan`).

E. CAGE

As described in Ref. [9], we train a simple Ridge regression model to fit the dynamics, based on the structure of both the initial state and the cage state. The cage state is defined as the average position of particles before any rearrangement occurs. To obtain the cage state, we perform a Monte Carlo simulation where all the particles are confined to a sphere of radius r_c^α. For both glassy systems we use $r_c^\alpha = 1.25\sigma_{\alpha\alpha}$ with $\sigma_{\alpha\alpha}$ the particle diameter of type α.

To capture the local structure, we use recursive, rotationally invariant parameters that capture both the radial density, as well as the n-fold symmetry in various shells around each particle. For the 3D system, we use exactly the same set of descriptors as in Ref. [9]. For the 2D system we use the same density parameters as for 3D [9], however the angular parameters are altered to better reflect 2D symmetry. Specifically, to capture the n-fold symmetry for the 2D system we use

$$\Phi_i^l(r, \delta) = \sqrt{\phi_i^l \cdot (\phi_i^l)^*}$$

with

$$\phi_i^l(r, \delta) = \frac{1}{Z} \sum_{i \neq j} e^{\frac{-(r_{ij} - r)^2}{2\delta^2}} e^{il\theta_{ij}},$$

and

$$Z = \sum_{i \neq j} e^{\frac{-(r_{ij} - r)^2}{2\delta^2}},$$

where θ_{ij} is the angle between a fixed axis (e.g., x- or y-axis) and the bond joining the i^{th} particle with a particle j. In both 2D and 3D we consider $l \in \{1, 12\}$.

Combining both the radial density and the angular parameters, we obtain a total of 462 parameters for the KA2D system:

- 294 parameters that described the radial density up to the 5^{th} minimum in the pair correlation function (which is located at $4.8\sigma_{11}$).
  - 60 equally spaced in the interval $r/\sigma_{11} \in (0.5, 2.0]$ with $\delta = 0.025$.

  - 20 equally spaced in the interval $r/\sigma_{11} \in (2.0, 3.0]$ with $\delta = 0.050$.
  - 18 equally spaced in the interval $r/\sigma_{11} \in (3.0, 4.8]$ with $\delta = 0.100$.

  Note that the radial density functions are type specific, such that in the ternary system the total number of functions above is multiplied by three.

- 168 parameters that capture the n-fold symmetry up to the second minimum of the pair correlation function (located at $2.3\sigma_{11}$).
  - 14 equally spaced in the interval $r/\sigma_{11} \in [1.0, 2.3]$ with $\delta = 0.1$ and $l \in [1, 12]$.

  Note that the radial functions do not take the particle species into account.

As described in Ref [9], for the 3D system we obtain a total of 366 parameters. In addition to the zeroth order generation parameters, for both the 2D and 3D system, we additionally include two generations of structural parameters that are iteratively averaged over the nearest neighbours (see Ref. [10]).

The dataset on which we train our Ridge regression thus includes three generations of structural descriptors for both the initial structure, as well as the cage state. Additionally, for each particle we include the distance between the cage state and the initial positions $\Delta r_i^{\rm cage} = |\boldsymbol{r}_i^{\rm cage} - \boldsymbol{r}_i^{\rm init}|$. This means that the local structure of each particle in the 2D system is described by a total of 2197 parameters, and in the 3D system a total of 2773 parameters.

To predict the dynamics, we standardise the data and then train the Ridge regression model on 300 (KA2D) and 100 (3D KA) snapshots, respectively. The only free parameter that is tuned for this ML method is the regularization parameter α, which sets the strength of the penalty for large weights in the Ridge regression. We train the model on various values of $\alpha \in \{10^{-5}, 10^{5}\}$, and then choose the α that yields the highest correlation. The training time per timestep in both 2D and 3D is less than 10 minutes on a standard CPU.

F. GlassMLP

The methodology employed in this manuscript corresponds exactly to the procedure described in detail in Ref. [5], including all training meta data. In particular, we calculate a set of M_S physics-inspired descriptors, including the coarse-grained local density, $\overline{\rho}^i_{L,\beta} = \sum_{j\in N^i_\beta} e^{-r_{ij}/L}$, which is coarse-grained over a distance L by summing over all N^i_β particles of type β within distance $r_{ij} = |\boldsymbol{r}_i^{\rm inh} - \boldsymbol{r}_j^{\rm inh}| < 20\sigma_{11}$ of particle i. Importantly, particle positions are extracted from the inherent structures $\boldsymbol{r}_i^{\rm inh}$, and not directly from the thermal structure. Similarly, we include the potential energy, perimeter of the Voronoi cell and local variance of potential energy (see Ref. [5] for details).

Different from Ref. [5] we additionally include one further structural descriptor, based on the distance between the inherent and the thermal states of particle i, $\Delta r_i^{\rm inh} = |\boldsymbol{r}_i^{\rm inh} - \boldsymbol{r}_i^{\rm th}|$: $\overline{\Delta r}^i_{L,\beta} = \sum_{j\in N^i_\beta} \Delta r_i^{\rm inh} e^{-r_{ij}/L}$. This descriptor does not affect the long-time prediction and was therefore discarded in Ref. [5], but is reintroduced here mainly to improve the short-time performance of GlassMLP.

To predict propensity in the KA (3D) system, GlassMLP has in total 618 free parameters. In the ternary KA2D system, the number is slightly increased (765). The training just takes 2-5 minutes with an octa-core CPU or a Laptop GPU (NVIDIA T600 Laptop).

G. SE(3)

The model is exactly that presented as the main model of Ref. [11] (version 2, august 2023).

We recall the main features for completeness and to avoid possible confusion. In the SE(3) GNN we use 8 layers. The structure of each layer is the same: 8 channels for each l component, with $l = 0, 1, 2, 3$ (namely, $l_{max} = 3$). The input data consists of the thermal positions (not quenched to the IS) and of the particles potential energy (but computed from the inherent state's particles positions). We predict simultaneously propensities for all particle types and time steps. The connectivity graph is computed using a threshold distance of $2.0\sigma_{11}$. The input node features are the particle type (one-hot encoded) concatenated with the potential energy (of the IS). For the 2D case, we introduce an artificial third component to the position of all particles ($\forall i, z_i = 0$) and used spherical harmonics embedding. This is overkill but is a quick way to adapt the 3D scheme to 2D data.

The radial-encoding MLP encodes the radius on (Bessel) basis functions using 10 basis vectors, followed by a hidden layer of 16 neurons, and uses a dropout of 0.3. We use batch normalization between each convolution layer. We minimise the MSE loss with a L^2 regularization coefficient $\beta = 10^{-7}$, using a batch size of 2 graphs (8 in 2D), a learning rate of 10^{-3} with Adam ($\beta_1 = 0.99$). As opposed to Ref. [11], we do not use a validation set in this manuscript and therefore do not perform early stopping. We simply take the last epoch model (the accuracy and losses are basically flat when we interrupt learning).

The model has $52,660$ parameters for 3D data (respectively $53,394$ in 2D). We perform 200 epochs (100 in 2D), which takes approximately 11 hours (27 hours in 2D) using an NVIDIA Tesla V100 (32GB) (the CPUs on the node are 2 Intel Xeon Gold 6148 20 cores (40 threads) at 2.4 GHz (Skylake)). Note that we train simultaneously the 10 time steps (6 for KA2D) of a given state point

(temperature), therefore the training does not have to be repeated for each timescale.

H. SBO

Unsupervised learning of local structure fluctuations is carried out for the 3D KA samples along the lines of Ref. [12], using the `partycls` Python package [13]. Namely, we characterise the local structure around particle i using the smooth bond-order (SBO) descriptor

$$X^{\mathrm{SBO}}(i) = (Q_0^S(i), \ldots, Q_{l_{\max}}^S(i)), \tag{2}$$

where Q_l^S are smoothed bond-orientational invariants of order l computed over the first coordination shell of particle i, see Refs. [12] for full details. The maximum order $l_{\max}$ is equal to 8. A PCA is then carried out for each chemical species, to identify the directions in the descriptor space that capture the largest structural fluctuations. In particular, the projection on the first principal component, $\tilde{X}_1^{\mathrm{SBO}}(i)$, is a measure of structural heterogeneity that is correlated to some extent with the dynamics of glassy binary mixtures [12]. The correlation between the first projection of the SBO descriptor and the propensity of motion is calculated directly on the test dataset, since there is usually no separate training stage in unsupervised learning. Finally, along with the bare SBO descriptor, we consider a coarse-grained (CG) version

$$\tilde{X}_i^{CG} = \frac{\sum_j \tilde{X}_j \cdot w(r_{ij}; L)}{\sum_{j=1}^N w(r_{ij}; L)}.$$

where the sum over j runs over the particles of the same species as i. As in Ref. [14] we use an exponential function $w(r; L) = e^{-r/L}$, and we set $L = 1$.

I. Other structural descriptors

In addition to these models we have also included several traditional structural descriptors.

Density:
The density is connected to the structural input of GlassMLP (see definition Sec. I F above). In particular, we have used $\overline{\rho}^i_{L=5,\beta=\mathrm{all}}$, i.e. coarse-grained over a distance $L = 5$ considering particles of all types. The only difference to GlassMLP is that we have evaluated the density in the thermal states.

Potential energy $\overline{E}^{\mathbf{inh}}_{\mathbf{pot}}$:
The potential energy in the inherent state, $\overline{E}^{\mathrm{inh}}_{\mathrm{pot}} = \overline{E}^i_{L=5,\beta=\mathrm{all}} = \sum_{j\in N^i_\beta} E^j e^{-r_{ij}/L}/\bar{\rho}^i_{L,\beta}$, is similarly part of the structural descriptors which is used as input for GlassMLP [5]. Here, $E^i = \sum_{j\neq i} V(r_{ij})/2$ is the potential energy of particle i.

Bond-order 3D (Ψ_6):
The bond-order descriptor Ψ_6^i of particle i for the KA system in three dimensions is defined via the complex coefficient [15]

$$q^i_{l,m} = \frac{1}{N_b} \sum_{j\in N_b} Y_l^m(\boldsymbol{r}_{ij}), \tag{3}$$

where N_b is the number of bonds of particle i, defined as all neighbours within a cutoff of $r_{\mathrm{cut}} = 2.0\sigma_{11}$, and $Y_l^m(\boldsymbol{r}_{ij})$ are the spherical harmonics. Using q^i we define,

$$\Psi^i_l = \sqrt{\frac{4\pi}{2l+1} \sum_{m=-l}^{m=l} \left| q^i_{l,m} \right|^2}. \tag{4}$$

As before, we also further coarse-grain the descriptor, $\overline{\Psi}^i_{6,L=5,\beta=\mathrm{all}} = \sum_{j\in N^i_\beta} \Psi^j_6 e^{-r_{ij}/L}/\bar{\rho}^i_{L,\beta}$.

Bond-order 2D (Ψ_4):
For the KA2D system, we slightly adapt the definition,

$$q^i_l = \frac{1}{N_b} \sum_{j\in N_b} e^{\mathrm{i}l\theta_{ij}}, \tag{5}$$

which is strongly related to the n-fold symmetry defined in Sec. I E. From this we calculate,

$$\Psi^i_l = \sqrt{q^i_l \cdot (q^i_l)^*}, \tag{6}$$

which is subsequently coarse-grained in the same way as defined in the 3D case.

Tanaka's Θ order parameter:
In Ref. [16] Tong and Tanaka proposed the usage of a structural descriptor Θ^i, which quantifies the strength of local packing around particle i (see Ref. [16] for details and definitions). We similarly coarse-grain this descriptor $\overline{\Theta}^i_{L=5,\beta=\mathrm{all}} = \sum_{j\in N^i_\beta} \Theta^j e^{-r_{ij}/L}/\bar{\rho}^i_{L,\beta}$.

In a 2D system of polydisperse harmonic spheres, Ref. [16] reports a Spearman's rank correlation coefficient of around $\rho_{\mathrm{S}} \approx 0.9$ for the coarse-grained $\overline{\Theta}$ with propensity. ρ_{S} is usually strongly related to Pearson correlation ρ_{P}. In this roadmap, we show that the performance is significant reduced when applied to systems which are not prone to crystallization. Additionally, some of the performance difference might emerge from calculating Pearson correlation over all particles independent of their radii in Ref. [16], as opposed to making independent predictions for each particle type as done in this manuscript. This emphasises the importance of using standardized datasets and performance metrics to quantify and validate the performance of newly developed descriptors and techniques.

II. ADDITIONAL BENCHMARKING

In the following, we will discuss additional metrics and ways to visualise the performance of the various

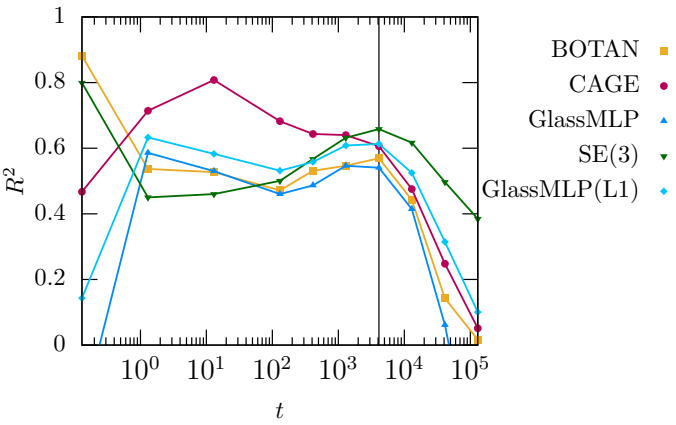

FIG. 1. Coefficient of determination $R^2 = 1 - \sum_i \left(\mathcal{R}_i^{\mathrm{MD}} - \mathcal{X}_i^{\mathrm{ML}}\right)^2 / \mathrm{var}(\mathcal{R}_i^{\mathrm{MD}})$ between the model predictions and the ground truth in the KA system (3D) at $T = 0.44$ for various ML models. The dynamical variable is the propensity of displacements. The vertical line marks the structural relaxation time τ_α. GlassMLP(L1) denotes the GlassMLP model [5] trained without the additional terms in the loss function using only the L^1 norm.

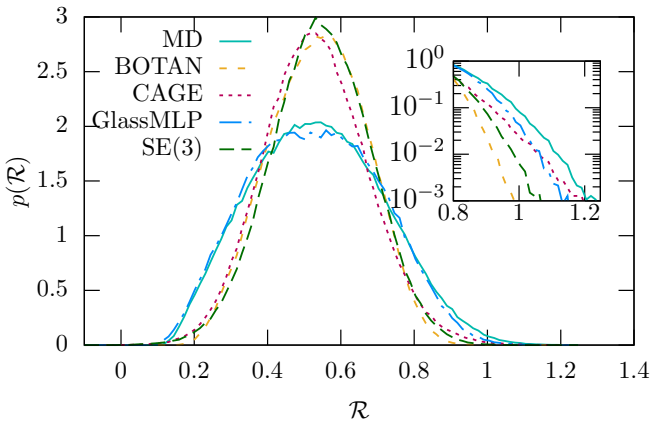

FIG. 2. Probability distribution of predicted and simulated propensities at $t = \tau_\alpha$ for the KA system (3D). The dynamical variable is the propensity of displacements $\mathcal{R}$. The inset shows a zoom to the tail, with logarithmic y-axis.

methodologies which have been benchmarked intensively in Sec. V of the main text.

A. Coefficient of determination R^2

The coefficient of determination $R^2 = 1 - \sum_i \left(\mathcal{R}_i^{\mathrm{MD}} - \mathcal{X}_i^{\mathrm{ML}}\right)^2 / \mathrm{var}(\mathcal{R}_i^{\mathrm{MD}})$ is a popular measure to quantify the quality of a fit, including, in particular, supervised ML models. In Fig. 1 the coefficient of determination is shown for exactly the same data as was analysed with the Pearson correlation in Fig. 3. It can be observed that, in general, the two measures ρ_{P} and R^2 do not show very different results. In particular, SE(3) still features the best performance at $t = \tau_\alpha$. The most pronounced difference is that GlassMLP performs worse when measured with R^2 instead of ρ_{P}. This can be explained by the usage of a loss function which contains additional terms on top of the standard L^1 or L^2 norm [5]. To validate this explanation we show the results for GlassMLP(L1) trained without these additional contributions and find results significantly better than for the original GlassMLP (see Fig. 1).

B. Probability distribution of propensities

Another interesting observable is the probability distribution of the propensities of displacement, $p(\mathcal{R})$, which can be extracted both from the MD simulations and from the ML models. A similar analysis has been performed in Ref. [5]. As can be seen in Fig. 2 the propensities predicted by the models tend to underestimate the variance of the distribution. This observation can be rationalized, since predicting outliers could be very costly in the loss function during training, hence networks tend to predict small variances. The only network which agrees quite well to the MD prediction is GlassMLP, which has been constructed to adapt the variance by an explicit contribution in its loss function [5]. However, as can be seen in the inset of Fig. 2, even for GlassMLP the tail of strongly moving particles is not perfectly reproduced. Focusing on this tail would likely significantly worsen the overall performance of the model.

C. Scatter plots of propensity predictions

The Pearson correlation ρ_{P} or the coefficient of determination R^2 are very good measures to quantify the performance of a network using a scalar quantity (see Figs. 3-6 in the main text and Fig. 1). However, they obviously cannot capture all details of the connection between the true and predicted propensities. In Fig. 3 we therefore show a 2D histogram visualizing this connection in greater detail. The overall impression of these figures is very similar to what we have concluded before. The most obvious difference between the methods is that GlassMLP is symmetric around the straight line, while the other techniques are slightly asymmetric. This observation is connected to the differences in the probability distribution as discussed in the previous paragraph.

D. Snapshots

The best way to visualise the predictions are snapshots of the amorphous structure, where each particle is coloured according to its propensity $\mathcal{R}$. In Fig. 4 we show an exemplary structure and the predicted propensi-

ties of displacement for various different models. Generally, strong correlations between the models and the MD simulations can be observed, as was expected from the high Pearson correlations. The strongest visible difference between MD and the ML models is that the latter are much smoother, in particular the ones based on GNNs (BOTAN and SE(3)).

E. Cross correlations

To study connections between the different models, we also investigate the cross correlations between the different ML techniques and other structural descriptors. Unsurprisingly, the strongest cross correlations are between the various ML models, since they all correlate strongly with propensity. The strongest correlation can, in fact, be detected between SE(3) and GlassMLP (see Fig. 5). This can be rationalized by the observation that both techniques also have the strongest correlation with the coarse-grained potential energy, $\overline{E}^{\mathrm{inh}}_{\mathrm{pot}}$. This could be expected since both techniques receive the potential energy as explicit input.

Another strong cross correlation is observable between the unsupervised SBO CG technique and density, showing that these two descriptors are strongly linked. This indicates that the strongest asymmetries in the structure emerge from density fluctuations.

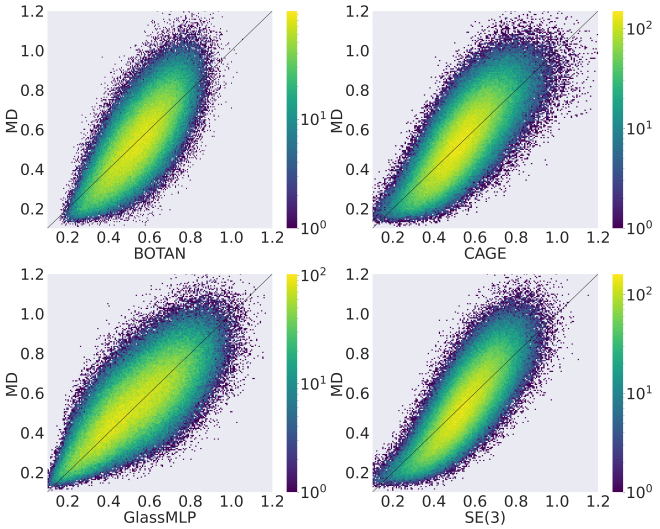

FIG. 3. 2D histograms of particle propensity $\mathcal{R}_i$ to compare different ML techniques to the MD ground truth in the KA system (3D, $t = \tau_\alpha$). Each entry in the histogram represents one particle, i, where the x/y-coordinates correspond to the propensities as predicted by ML and from MD simulations, respectively.

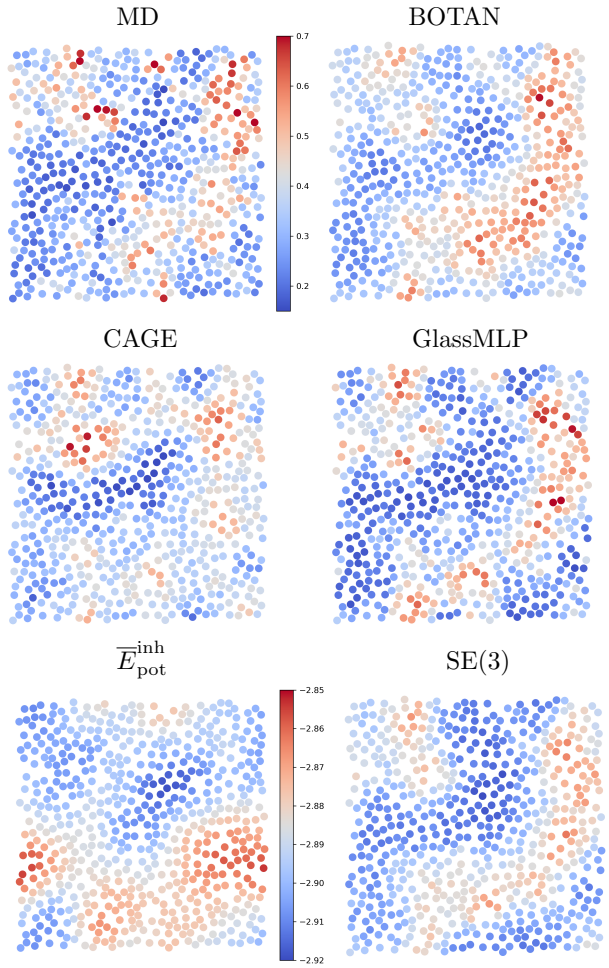

FIG. 4. Snapshots of an exemplary configuration of the KA2D system (shown are only type A particles). Each particle is coloured according to its propensity of displacement $\mathcal{R}$ at $t = \tau_\alpha$, as calculated from MD simulations or predicted from various ML techniques (see Fig. 3 in the main text for the respective Pearson correlations). Each snapshot has the same color bar, except $\overline{E}^{\mathrm{inh}}_{\mathrm{pot}}$.

F. Learning curves

An important feature of ML models is also how they cope with limited training data. In standard fitting procedures, one would usually expect that the more fitting parameters a model has, the more training data is required to achieve a certain accuracy. This 'rule of thumb' has, however, been repeatedly questioned for deep networks, which often perform well even when overparameterized. When investigating Fig. 6 we observe that indeed SE(3) already performs excellent, even with only $N = 3277$ data points, despite having $>$ 50k fitting parameters. Furthermore, the increase in performance is very continuous, implying that the network might perform even better with $>$ 400 independent structures (which was the maximum of the dataset available). The

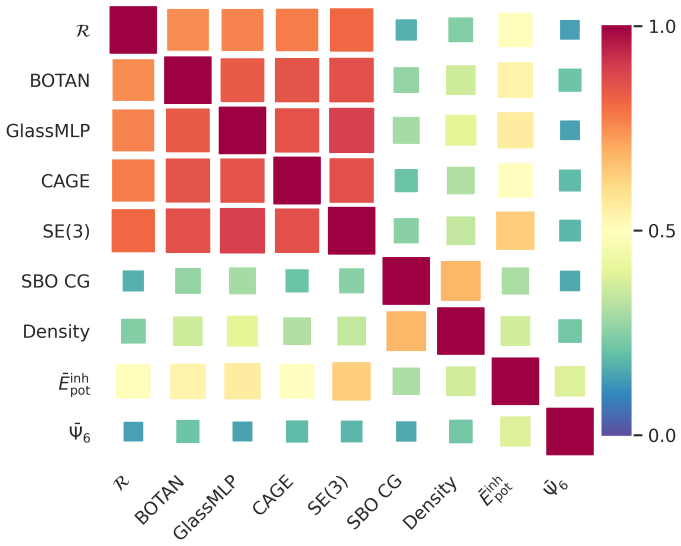

FIG. 5. Pearson cross-correlations $\rho_{\text{P}}(X, Y)$ between different ML models and structural descriptors in the KA system (3D) at $t = \tau_\alpha$.

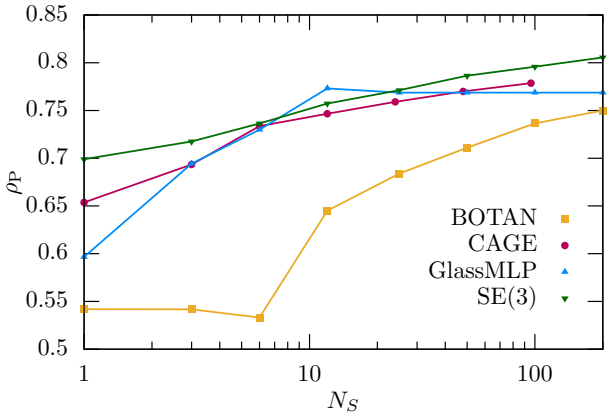

FIG. 6. Performance of network, as quantified by the Pearson correlation coefficient ρ_{P}, trained with differently sized training sets in the KA system (3D) at $t = \tau_\alpha$. N_S denotes the number of samples, where each sample consists of $N = 3277$ particles.

learning curves of CAGE and BOTAN closely resembles SE(3) just with a very small reduction in performance.

Contrarily, the learning curve of GlassMLP starts at a significantly smaller Pearson correlation, but quickly catches up with SE(3) to slightly surpass its performance at around $N_S = 10$ (i.e. 32770 data points). However, the performance of GlassMLP does not further improve for larger $N_S > 20$. It might be necessary to increase the number of structural input descriptors to improve GlassMLP in situations where much training data is available. It should be mentioned that there is an important difference between the training procedures of SE(3) and the other models. While GlassMLP uses for

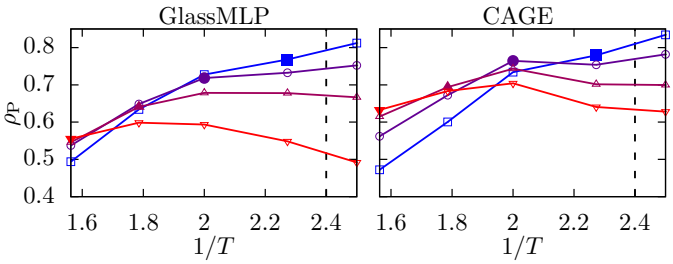

FIG. 7. Transferability in temperature T of trained networks in the KA system (3D) at the structural relaxation time τ_α. The figures is identical to Fig. 6 in the main text, just featuring an additional temperature $T = 0.4$.

$N_S = 1$ only twice the number of epochs for training (i.e. $N_e = 600$), since no improvement in performance can be observed beyond this point, SE(3) requires a significant increase in epochs for small N_S: $N_e(N_S) = 8 \cdot 10^4/N_S$. This implies that for $N_S = 1$, the training time of GlassMLP is roughly 5 seconds, while for SE(3) it remains roughly 11 hours.

G. Transferability

We have investigated in detail the transferability of trained networks in the benchmarking section V of the main text. Here, we study for two models the performance of GlassMLP and CAGE when applied to an even lower temperature $T = 0.4$, taken from the dataset presented in Ref. [17]. Different from the main KA dataset we only have $N_S = 12$ individual structures available, but use $N_R = 100$ different isoconfigurational replicas. Due to the larger number of replicas we expect the Pearson correlations to be slightly larger than for $N_R = 32$. The results in Fig. 7 indeed confirm this expectation, nevertheless, they also highlight the strong transferability of the models. A network trained at $T = 0.44$, with a relaxation time an order of magnitude smaller than at $T = 0.4$, CAGE can predict structural relaxation with a Pearson correlation of up to $\rho_{\text{P}} = 0.84$. We believe that this is an important result towards using transferability to study glassy liquids at extremely low temperature.

H. Bond-breaking propensity in the KA system (3D)

Finally, we also briefly investigate the performance of ML models and other structural descriptors in predicting the bond-breaking propensity in the KA system (see Fig. 8). While the predictability for small times is much weaker than for the propensity of displacements, $\mathcal{R}$, analysed in Fig. 3, we observe better performance for longer times around the structural relaxation time scale $t = \tau_\alpha$. This observation is identical to the discussion in Sec. V of the main text after Fig. 5 for the bond-

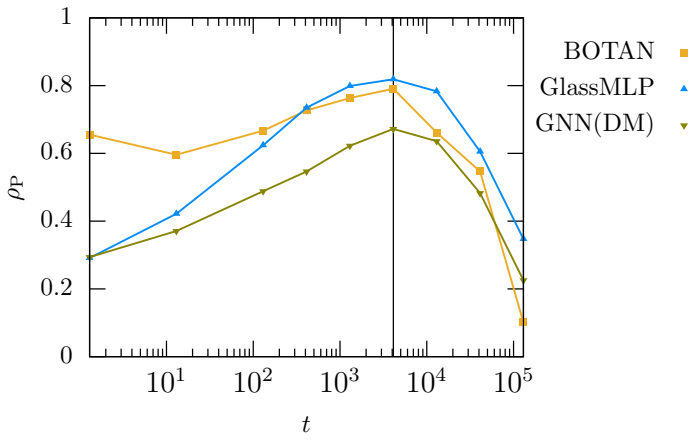

FIG. 8. Pearson correlation ρ_P with the ground truth in the KA system (3D) at $T = 0.44$ for various different techniques. Different from Fig. 3, here the dynamical variable is the bond-breaking propensity. The vertical line marks the structural relaxation time τ_α. GNN(DM) refers to the original GNN [7].

breaking propensity in the KA2D system. We further find that both BOTAN and GlassMLP significantly outperform the original GNN proposed by DeepMind ([7], GNN(DM)). This shows the great advancements in the field within the past three years. We hope that this roadmap further fuels the development and application of ML techniques to analyse glassy liquids.

[1] W. Kob and H. C. Andersen, Testing mode-coupling theory for a supercooled binary lennard-jones mixture i: The van hove correlation function, Phys. Rev. E **51**, 4626 (1995).

[2] H. Shiba, M. Hanai, T. Suzumura, and T. Shimokawabe, BOTAN: BOnd TArgeting Network for prediction of slow glassy dynamics by machine learning relative motion, The Journal of Chemical Physics **158**, 084503 (2023).

[3] A. Ninarello, L. Berthier, and D. Coslovich, Models and algorithms for the next generation of glass transition studies, Phys. Rev. X **7**, 021039 (2017).

[4] L. Berthier, E. Flenner, C. J. Fullerton, C. Scalliet, and M. Singh, Efficient swap algorithms for molecular dynamics simulations of equilibrium supercooled liquids, Journal of Statistical Mechanics: Theory and Experiment **2019**, 064004 (2019).

[5] G. Jung, G. Biroli, and L. Berthier, Predicting dynamic heterogeneity in glass-forming liquids by physics-inspired machine learning, Phys. Rev. Lett. **130**, 238202 (2023).

[6] B. Guiselin, C. Scalliet, and L. Berthier, Microscopic origin of excess wings in relaxation spectra of supercooled liquids, Nature Physics **18**, 468 (2022).

[7] V. Bapst, T. Keck, A. Grabska-Barwińska, C. Donner, E. D. Cubuk, S. S. Schoenholz, A. Obika, A. W. Nelson, T. Back, D. Hassabis, *et al.*, Unveiling the predictive power of static structure in glassy systems, Nature Physics **16**, 448 (2020).

[8] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, C. Gulcehre, F. Song, A. Ballard, J. Gilmer, G. Dahl, A. Vaswani, K. Allen, C. Nash, V. Langston, C. Dyer, N. Heess, D. Wierstra, P. Kohli, M. Botvinick, O. Vinyals, Y. Li, and R. Pascanu, Relational inductive biases, deep learning, and graph networks (2018), arXiv:1806.01261.

[9] R. M. Alkemade, F. Smallenburg, and L. Filion, Improving the prediction of glassy dynamics by pinpointing the local cage, The Journal of Chemical Physics **158**, 134512 (2023).

[10] E. Boattini, F. Smallenburg, and L. Filion, Averaging local structure to predict the dynamic propensity in supercooled liquids, Phys. Rev. Lett. **127**, 088007 (2021).

[11] F. S. Pezzicoli, G. Charpiat, and F. P. Landes, SE(3)-equivariant graph neural networks for learning glassy liquids representations, arXiv preprint arXiv:2211.03226 (2022).

[12] D. Coslovich, R. L. Jack, and J. Paret, Dimensionality reduction of local structure in glassy binary mixtures, J. Chem. Phys. **157**, 204503 (2022).

[13] J. Paret and D. Coslovich, partycls: A python package for structural clustering, JOSS **6**, 3723 (2021).

[14] H. Tong and H. Tanaka, Revealing hidden structural order controlling both fast and slow glassy dynamics in supercooled liquids, Phys. Rev. X **8**, 011041 (2018).

[15] P. J. Steinhardt, D. R. Nelson, and M. Ronchetti, Bond-orientational order in liquids and glasses, Phys. Rev. B **28**, 784 (1983).

[16] H. Tong and H. Tanaka, Structural order as a genuine control parameter of dynamics in simple glass formers, Nature communications **10**, 5596 (2019).

[17] D. Coslovich, M. Ozawa, and W. Kob, Dynamic and thermodynamic crossover scenarios in the kob-andersen mixture: Insights from multi-cpu and multi-gpu simulations, The European Physical Journal E **41**, 1 (2018).