Reconstructing phase diagram of a non-Markovian random walk by deep machine learning

A.A. Lobashev[1], S.K Nechaev[2,3], M.V. Tamm[1,4]

[1] *Department of Applied Mathematics, MIEM, National Research University Higher School of Economics, 123458 Moscow, Russia*

[2] *Interdisciplinary Scientific Center Poncelet (CNRS UMI 2615), 119002 Moscow, Russia*

[3] *P.N. Lebedev Physical Institute RAS, 119991 Moscow, Russia*

[4] *Faculty of Physics, Lomonosov Moscow State University, 119992 Moscow, Russia*

(Dated: November 24, 2020)

Here will be an abstract

PACS numbers: ?, ?

Methods of machine learning are useful for finding regularities in big datasets of different nature [**?**]. Recently, the application of these methods to various problems of statistical physics has been rapidly developing. One class of problems where its application seems to be fruitful is the determination of characteristics of an underlying random walk process based on individual trajectories [**?**]. Another is the determination of a phase diagram of an equilibrium statistical physical system based on the samples of microstates at various values of parameters[**?**].

From the machine learning point of view in both cases it is instructive to think of a studied model/process as rule (function, operator), which gets a relatively small set of macroscopic parameters as an input and returns a vector in a very multidimensional phase space, which we call *microscopic state*. Importantly, this rule is probabilistic, so one can think of a microstate $\mu \in \mathbb{R}^M, M \gg$ as a function of the set of macroscopic parameters p and some vector of random variables $\boldsymbol{\xi}$, $\mu = f(p, \boldsymbol{\xi})$.

The single trajectory analysis corresponds to an inverse problem, its aim is to infer the macroscopic parameters given some set of known microscopic states, i.e., to infer p given some collection of μ's. Phase classification or construction of phase diagrams is in a sense a problem of clustering: the microscopic states corresponding to the same phase are, in some respect, similar to each other, while those corresponding to different phases are different from each other. The basic idea is, therefore, is that the space $\mathbb{P}$ of parameters p can be separated into a set of mutially non-intersecting domains P_i (phases) so that the function $f(p, \boldsymbol{\xi})$ can be approximated by $\bar{f}(I(p), \boldsymbol{\xi})$ where $I(p)$ is an indicator function returning the number of the phase p belongs to:

$$I(p) = i \text{ if and only if } p \in P_i \tag{1}$$

and the problem is to find a separation of the space $\mathbb{P}$ into phases which minimizes the difference between $f(p, \boldsymbol{\xi})$ and $\bar{f}(I(p), \boldsymbol{\xi})$. Putting it in other words, we expect that if one takes two points $p_{1,2} \in \mathbb{P}$, constructs a collection of microstates corresponding to each of them, and trains a classificator to distinguish between these microstates, then the classificator's precision will depend on whether p_1 and p_2 belong to the same phase: for the points belonging to different phases the precision will converge to unity with growing size of the learning sample, while for the points belonging to the same phase it will remain roughly 1/2.

Then, one might suggest a natural way to construct a phase diagram as follows [1]. Separate the space $\mathbb{P}$ into random domains (the number of domains should be larger than the expected number of phases). Train classificator to distinguish microstates corresponding to different domains, then move domain walls randomly, train the classificator once again, and check if the new classificator has a better precision, if yes, accept the new domain wall positions. Continue this process until the domain structure allowing for a maximal possible precision of the classificator is reached. The resulting set of domains should reproduce the phase diagram. However, this algorithm is very demanding numerically, since one must retrain classificator from scratch every time the domain walls are moved.

In this paper we use the machine learning methods to study the morphology diagram of a novel class of non-Markovian random walks, which is a generalization of reinforced random walks studied in [2, 5**?**]. We call this novel class "random walks with volume and surface reinforcement".

Let us first remind that reinforced random walk is a discrete-time random walk on a lattice (graph) defined in a way that the walker distinguishes between visited and unvisited cites [2–5] (compare also with greedy forager problem [6], so that, given a choice, the probability of going to an already visited site is $\exp(a)$ times larger than probability of going to a new cite, with a being a parameter of the model. It is known, that for $d > 1$ there is a morphological transition in this model from Brownian-motion-like trajectories at small a to supercollapsed trajectories at large a. The visited region for a long N-step walk in the supercollapsed state is a compact ball linear size growing as $N^{1/(d+1)}$. Presumably, the limiting shape of this ball is spherical, although it is not proven, the thickness of its boundary is $O(1)$ with $N \to \infty$. On a 3-dimensional simple cubic lattice the transition from Brownian to supercollapsed regime occurs at $a = 1.831 \pm 0.002$. Note also that the $a \to -\infty$ limit of this model is the so-called true self-avoiding walk [7].

Here we consider a generalization of this model, which distinguishes not only steps inside and outside of the visited area, but also steps along its surface. More formally, consider a random walk on some graph $\mathcal{G}$ defined by its sets of vertices and edges $\mathcal{G} = (\mathbb{V}, \mathbb{E})$. Let V_N be a set of vertices visited up to step N of the walk, and v_N – the position of the walker at step N. For each site v_i, which is neigbour of v_N, define a weight

$$w(v_i) = I(e_{N,i} \in \mathbb{E}) \begin{cases} \exp a, & \text{if } v_i \in V_N; \\ 1, & \text{if } v_i \notin V_N \text{ and } v_N \text{ is the only} \\ & \text{neighbor of } v_i \text{ from } V_N; \\ \exp b, & \text{if } v_i \notin V_N \text{ and there are} \\ & \text{other neighbors of } v_i \text{ in } V_N, \end{cases} \tag{2}$$

where $e_{N,i}$ denotes the edge connecting v_N and v_i, and I is the indicator function. Now the probability of a walker going from v_N to v_i on step $N+1$ is proportional to $w(v_i)$:

$$P(v_{N+1} = v_i) = \frac{w(v_i)}{\sum_k w(v_k)} \tag{3}$$

Importantly, the walks with volume and surface reinforcement defined in this way are substantially non-Markovian, with probabilities of each new step depending on the whole history of the walk. Depending on a and b the trajectories of the walks exhibit several distinct morphologies, including those of Brownian random walk (negative a and b), supercollapsed ball ($b = 0$, large positive a), and structures akin to crumpled globule state of polymer chains[8–10] (large positive b, large negative a, compare with the model suggested in supplementary materials to[11]). Enumerating all the different types of trajectories, and separating the plane (a, b) into regions corresponding to different types (i.e., constructing a phase diagram of the model) is an interesting and challenging question. It is made more difficult by the fact that, contrary to the examples above, walk with volume and surface reinforcement is a non-equilibrium object, so there is no hope to use the standard machinery of equilibrium statistical mechanics. As we show below, the deep learning techniques allow to effectively classify phases and construct sketch of a phase diagram of this model based on classification of relatively short (typically much less then 10^5 steps) trajectories.

The main idea of our approach is as follows. Take a pair of parameters (a, b) and generate a sufficiently long trajectory of a reinforced random walk with these parameters (we stop the trajectory genration when it has visited $N = 10^4$ distinct points, see [12] for details). Now, take an image of this trajectory (a photo taken from a random angle) and predict, based on it, the possible range of parameters for which it was generated. For that purpose we simultaniously train two neural networks: one takes a photo of the trajectory and transforms it into a non-negative function defined over (a, b) plane; another takes the pair of initial parameters (a, b) and produces a smoothed Gaussian function centered at (a, b). The networks are trained in a way to make their outputs as close as possible. Fig.1 shows the outline of the classificating scheme. The idea is that, since two trajectories belonging to the same morphological phase are impossible to distinguish, the best neural network can do in order to predict (a, b) is to highlight the whole range of possible parameters for which a given trajectory could have been obtained. The label smoothing network (LSN), which is needed in order to prevent divergence in th loss function is a 4-layered perseptron with 32 neurons in each layer. The detailed architecture of the image-analyzing network U-Net is given in [12] and the overall loss function for a given set of trajectory $\mathbf{X}$ and parameters a, b reads

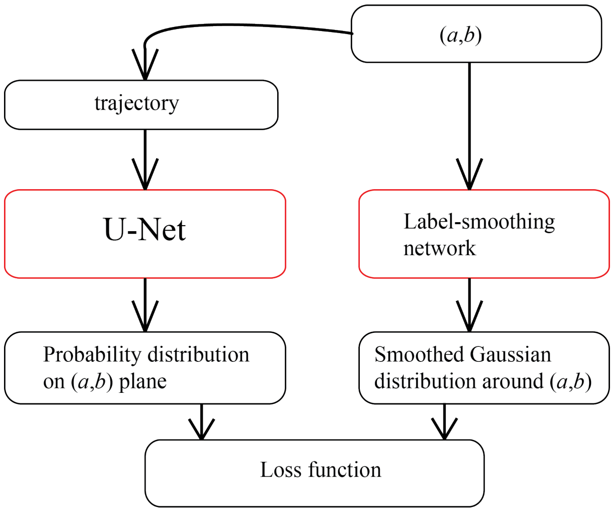
(a,b)
trajectory
U-Net
Label-smoothing network
Probability distribution on (a,b) plane
Smoothed Gaussian distribution around (a,b)
Loss function

Figure 1: General architecture of our machine learning approach.

$$\begin{aligned} \mathcal{L}(\mathbf{X}, a, b) &= \lambda \det(\mathrm{cov}(\mathrm{LSN}(a, b)))^2 + \\ &+ \sum_{\text{pixel-wise}} (\mathrm{UNet}(\mathbf{X}) - \mathrm{LSN}(a, b))^2 \end{aligned} \tag{4}$$

where $\mathrm{cov}(\mathrm{LSN}(a, b))$ is the covariance matrix of the Gaussian function, which is the output of the LSN. The first term in (6) is needed to prevent convergence of the output of both networks to a delocalized flat distribution. The detailed network training procedure is described in [12]. Fig.2 presents examples of the output of the networks. It is clearly seen that U-net indeed produces as its output some clearly defined regions in the parameter space.

Given the output of U-Net, one can define a distance between two trajectories $\mathbf{X}$ and $\mathbf{Y}$ as follows

$$d(\mathbf{X}, \mathbf{Y}) = \frac{\sum_{\text{pixel-wise}} (\mathrm{UNet}(\mathbf{X}) - \mathrm{UNet}(\mathbf{Y}))^2}{\sum_{\text{pixel-wise}} (\mathrm{UNet}(\mathbf{X}) + \mathrm{UNet}(\mathbf{Y}))^2} \tag{5}$$

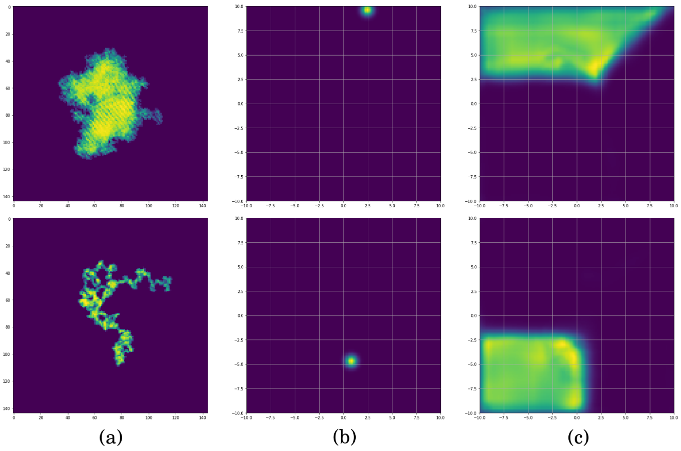

Figure 2: (a) Input of the first neural network: photo of a sample trajectory for $a = ..., b = ...$ (top), $a = ..., b = ...$ (bottom); (b) Output of a second neural network blurred point on the (a, b) plane; (c) the output of a second neural network: network's estimate of possible (a, b) region which might have grnerated the sample trajectory.

Clearly, $d(\mathbf{X}, \mathbf{Y})$ takes values between 0 and 1. Distance close to 0 corresponds to two trajectories generating a very similar output of U-Net, that is to say, areas on the (a, b) plane, which strongly overlap. In turn, dustance close to 1 means that wherever output UNet$(\mathbf{X})$ differs from 0, the output UNet$(\mathbf{Y})$ is essentially zero and vice versa. Therefore, distance defined above can work as an indicator of whether trajectories $\mathbf{X}$ and $\mathbf{Y}$ correspond to the same phase $(d(\mathbf{X}, \mathbf{Y}) \approx 1)$ or to different phases $(d(\mathbf{X}, \mathbf{Y}) \approx 0)$.

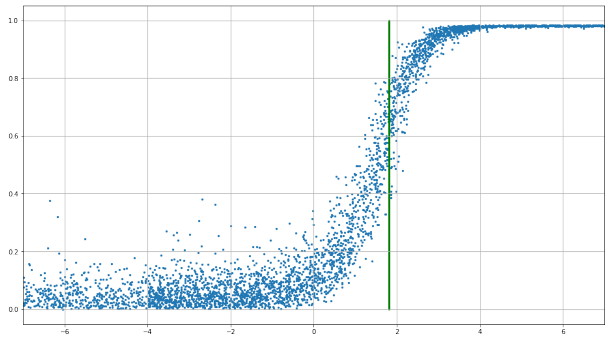

Figure 3: U-Net distance (5) from a Brownian motion trajectory $(a = -6, b = 0)$ to sample trajectories for $b = 0$. Vertical green line corresponds to the transition point $a \approx 1.831$ found by finite-size scaling in [3].

In order to check whether the resulting classification is reasonable, we look at the particular case of $b = 0$, which corresponds to volume-only reinforcement studied previously in [2–5]. Consider trajectories generated with $b = 0$ and various a and study their distance (5) from the trajectory generated at $a = -6, b = 0$, i.e. deep within the true self-avoiding walk [7] region. The results are shown in Fig.3. One clearly sees a well-defined transition from trajectories similar to true self-avoiding walk to some differet phase (the visited regions in this latter phase are known to be the asymptotically spherical and supercollapsed [2, 3]). The transition point as defined by the inflection point of the S-shaped curve in Fig.3 seems to be extremely close to $a_{cr} \approx 1.831$ found in [3] by finite-size scaling based on the statistics of much longer (up to 10^7 steps) trajectories.

The quality of the output may be further improved by a following trick. Take a batch of 512 trajectories for randomly chosen values of (a, b) and for each trajectory $\mathbf{X}_0(a_0, b_0)$ in the batch find 15 closest trajectories $\mathbf{X}_i$, $i = 1, \ldots 15$ in the sense of (5). Now, train network U-Net2, which has exactly the same architecture as U-Net to predict not just values of (a_0, b_0) but also those of all 15 (a_i, b_i) used to generating the trajectory $\mathbf{X}_i$. That is to say, to train the network U-Net2 we use a loss function

$$\mathcal{L}_2(\mathbf{X}) = \lambda \sum_{i=0..15} \det(\mathrm{cov}(\mathrm{LSN}(a_i, b_i)))^2 + \\ + \sum_{\text{pixel-wise}} \left(\mathrm{UNet2}(\mathbf{X}) - \sum_{i=0..15} \mathrm{LSN}(a_i, b_i) \right)^2 \qquad (6)$$

As seen in Figure .. of [12], the output of U-Net2 is similar to the output of U-Net but is much more uniform and has clearer boundaries. Thus, the network U-Net2 effectively separates the (a, b) plane into non-intersecting regions which we classify as phases.

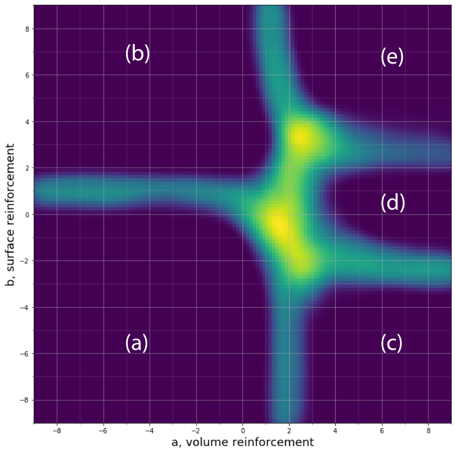

Figure 4: Phase diagram of the model in the (a, b) plane. Dark

Direct inspection of the output shows that there are 5 distinct outputs corresponding to 5 distinct morphologies. In order to present the resulting phase diagram we clustered the output of the neural nework into 5 clusters (see [12] for the details of the procedure). In Fig.4 we present the results of this clustering. Here dark blue regions correspond to such values of parameters for which corresponding trajectories are always classified into the same cluster, i.e. these parameters reliably belong inside a phase, rather than on a phase boundary. In turn, trajectories corresponding to the points in the light blue regions are, depending on the realization, classified into one of two clusters, meaning that they are in the vicinity of a transition line between two phases. Finally, trajectories corresponding to the yellow regions got classified into three or more different clusters, outlining the vicinity of triple points on the phase diagram.

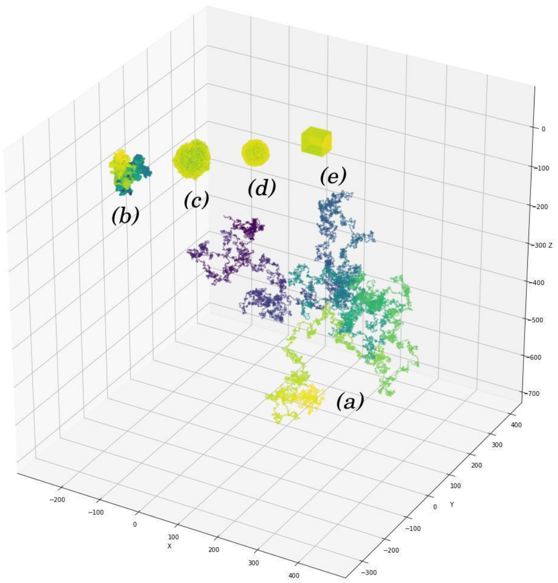

Figure 5: Examples of representative trajectories of the 5 phases found in a phase diagram.

In Fig.5 we show the examples of morphologies corresponding to the 5 regions in the phase diagram. Phase (a) is a simple Brownian motion or true self-avoiding walk with spatial extension of N-step walk proportional to $N^{1/2}$ and Gaussian distribution of the end-to-end distance. Phase (d) is the phase of supercollapsed ball first predicted in [2] and well-studied in the volume-reinforced ($b = 0$) case. The visited region of the walk is in this case asymptotically a smooth ball with radius growing as $N^{1/4}$ with growing number of steps.

Phase (c), first conjectured in [13], is similar to phase d but for the fact that the supercollapsed ball in this case has a rough surface and is porous. As a result, the surface and the volume of the ball are growing as $N^{2\sigma}$ and $N^{3\nu}$, respectively, with $\sigma > \nu$. There is an easy scaling argument establishing connection between σ and ν. Indeed, in the $e^a \gg \max(1, e^b)$ limit the walk spends most of the time inside the already visited volume, whose surface almost always reflects it. As a result, the probability of finding walker in any point inside the visited volume is the same (more precisely, converges to uniform distribution after many reflections from the boundary). If the walker is currently at the surface of the visited volume he has some small probability to go outside and thus increase the visited volume by one new lattice cite. If it is not on the surface (i.e., in the inner part of the visited volume), then the probability of increasing volume on the next step is zero. Thus,

$$\frac{dV}{dN} = const \times \frac{S}{V}; \ \rightarrow \ 6\nu = 1 + 2\sigma, \tag{7}$$

where constant depends on the particular values of a and b. In the case of phase (d) the exponents σ and ν are the same, which leads to $\nu = 1/4$ [2]. For phase (c) the exact values of ν and σ are unknown, and it is not even clear if they are constant throughout the phase or change with a and b.

In phase (b) the walk is mostly sticking to the surface of the already visited region, without either penetrating it or going away. As a result, visited volume grows, at least approximately, proportionally to the number of steps but the visited area forms a rather irregularly shaped blob (see Fig.5:b) with a very developed surface, which is reminiscent of the shapes of polymer rings in a melt [14, 15] (see also supplementary materials of [11] where a very similar construction was used to generate approximate initial condition for a crumpled chromatin-like polymer).

To understand phase (e) note that on th the $a, b \gg 1$ limit it is beneficial for the visited volume to have extended flat faces, which grow without defects up to the area of order e^b. As a result, crystaline-like objects like one shown in Fig.5:e are formed. Clearly, for any given a, b they are unstable in the $N \to \infty$ limit, but the length of the walk needed for the defects to destroy this crystaline structure is exponentially large. However, the existance of this phase is a peculiarity of the simple cubic lattice, which is conducive to the flat face formation. In [12] we check that phase diagram of a similar walk on body-centered cubic lattice does not include this phase. Instead, there is a direct transition from crumpled-globule-like phase (b) to supercondensed ball phase (d).

Here will be a discussion

The authors are grateful to L. Nazarov and N. Kotelevskii who contributed to preliminary numerical analysis of the reinforced random walk model by means of direct Monte Carlo simulations. This work is supported by BASIS Foundation grants 17-... and 19-...

[1] Van Nieuwenburg, Evert PL, Ye-Hua Liu, and Sebastian D. Huber. "Learning phase transitions by confusion." Nature Physics 13.5 (2017): 435-439.

[2] Sapozhnikov

[3] A. Ordemann, G. Berkolaiko, S. Havlin , and A. Bunde, Swelling-Collapse Transition of Self-Attracting Walks, Physical Review E, V. 61, N 2, P. 1005-1007 (2000).

[4] A. Ordemann, E. Tomer, G. Berkolaiko, S. Havlin, and A. Bunde, Structural Properties of Self-Attracting Walks, Physical Review E, V. 64, 046117 (2001).

[5] Foster, Jacob G., Peter Grassberger, and Maya Paczuski. "Reinforced walks in two and three dimensions." New Journal of Physics 11.2 (2009): 023009.

[6] Bhat, S. Redner, O. Benichou, 2017.

[7] Parisi, Peliti, 1983.

[8] A.Yu. Grosberg, S. Nechaev, E. Shakhnovich, J. de Physique, 1988.

[9] A.Yu. Grosberg, S. Havlin, et al., 1993.

[10] L. Mirny, Chromosome Res., 2009.

[11] M.V. Tamm, L.I. Nazarov, A.A. Gavrilov, A.V. Chertovich, "Anomalous diffusion in fractal globules." Physical review letters 114.17 (2015): 178102.

[12] A. Lobashev, S. Nechaev, M. Tamm, supplementary materials to this paper.

[13] N.Y. Kotelevskii, S.K. Nechaev, M.V. Tamm, “A new class of reinforced random walks”, Book of Abstracts of “International Conference on Computer Simulation in Physics and beyond”, p. 138, Moscow, 2018.

[14] Everaers, Rosa, Grosberg Rings

[15] Smrek, Halverson, Kremer, Grosberg, Review