StatMechGlass: Python based software for composition–structure prediction in oxide glasses using statistical mechanics

Bødker, Mikkel Sandfeld; Wilkinson, Collin J.; Mauro, John C.; Smedskjær, Morten Mattrup

Published in:
SoftwareX

DOI (link to publication from Publisher):
10.1016/j.softx.2021.100913

Publication date:
2022

Document Version
Publisher's PDF, also known as Version of record

SoftwareX 17 (2022) 100913

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

StatMechGlass: Python based software for composition–structure prediction in oxide glasses using statistical mechanics

Mikkel S. Bødker [a], Collin J. Wilkinson [b], John C. Mauro [b], Morten M. Smedskjaer [a,*]

[a] Department of Chemistry and Bioscience, Aalborg University, 9220 Aalborg, Denmark
[b] Department of Materials Science and Engineering, The Pennsylvania State University, University Park, PA 16802, USA

ARTICLE INFO

Article history:
Received 8 January 2021
Received in revised form 6 October 2021
Accepted 19 November 2021

Keywords:
Statistical mechanics
Glass structure
Modelling
Python

ABSTRACT

Knowledge of the distribution of short range-order structural units in oxide glasses is important for deciphering their composition–property relations. However, measurements of the fractions of such units are often difficult and time consuming, especially for multicomponent glasses. Here, we introduce StatMechGlass, a Python-based software for calculating the short range-order structure distribution in oxide glasses based on statistical mechanics. By accounting for the enthalpic and entropic contributions to the network interactions in glass-forming melts, the atomic-scale structures of the resulting glasses can be calculated. As input, the software requires accurate interaction enthalpy values that can be supplied by the user or obtained directly by the software from experimental structure data. StatMechGlass thus enables the prediction of composition–structure relations for any oxide glass composition. When coupled with existing composition–property databases of experimental data, it enables the construction of composition–structure–property databases and models. StatMechGlass is open source and designed in a modular fashion for easy tailoring for specific needs.

Code metadata

Current code version	*v1.5*
Permanent link to code/repository used of this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-21-00004
Legal Code License	*GPL-3.0 License*
Code versioning system used	*Git*
Software code languages, tools, and services used	*Python*
Compilation requirements, operating environments & dependencies	*Numpy, sklearn, scipy, matplotlib*
Support email for questions	mos@bio.aau.dk

1. Motivation and significance

Oxide glasses are non-crystalline materials with a range of applications ranging from buildings to medicine and electronics [1–4]. They are made from network-forming elements such as silicon, phosphorous and boron that are connected through covalently bonded oxygen atoms to form a three-dimensional network [5–7]. Electropositive network-modifying elements such as sodium, potassium and calcium alter the network by forming ionic bonds to oxygen, thus weakening the rigidity of the structure and altering the properties of the resulting glass [8]. In commercial glasses, these network-modifying cations are usually added to lower the melting temperature during the glass formation process, hence reducing the manufacturing cost. The type and fraction of both network-modifying and forming units can be tailored to produce an oxide glass with certain desirable macroscopic properties [9,10]. The process of optimizing the glass composition for a specific application has historically been carried out through tedious trial-and-error experimentation [11].

There is thus a need to predict the composition–property relations. Machine learning based models have recently been used for this purpose and show some promise [12,13]. However, these models are typically limited to interpolation considering the complex and non-linear composition–property relations in oxide glasses. On the other hand, structure–property models have shown some ability to extrapolate predictions, as the embedded short-range order structures have a more direct connection with properties than the chemical composition [14–16]. Quantification

* Corresponding author.
E-mail address: mos@bio.aau.dk (Morten M. Smedskjaer).

https://doi.org/10.1016/j.softx.2021.100913

of the fraction of structural units is the main limitation for applying structure–property models [17]. The disordered nature of oxide glasses complicates the structural analysis to obtain accurate data, with solid state nuclear magnetic resonance (ssNMR) spectroscopy being the most frequently used technique [18]. Though possible, obtaining high-quality structure data from ssNMR measurements can take hours to days of measuring time and the data processing is often non-trivial. Additionally, the element of interest must contain naturally abundant NMR-active isotopes.

Computational methods can be used to overcome the limitations of ssNMR measurements to determine the glass structure. Molecular dynamics (MD) simulation is the most frequently used method for obtaining oxide glass structures by computation [19]. However, as the structural distribution in oxide glasses is highly dependent on the potential energies associated with the breaking and formation of covalent and ionic bonds in the glass-forming melt, classical MD simulations are not always capable of replicating structures as this method is based only on empirical interatomic potentials [20,21]. To overcome this limitation, *ab initio* MD simulation applies quantum mechanical parameters, but the high degree of complexity limits the computation to a few hundreds to thousands of atoms and extremely short timescale [22,23]. Alternatively, the use of a reactive force field offers an intermediate MD approach to simplify the computation to a few reaction energy parameters [24]. While this method has been valuable in studying, e.g., glass surface reactions, it still suffers from relatively high cooling rates and low structural accuracy of bulk glasses.

Statistical mechanics has been used for some glass systems to overcome the experimental challenges and obtain more accurate structure predictions than MD simulations in general [25–29]. In this work, we introduce StatMechGlass, which is a Python based software for predicting short-range order structures in oxide glasses from their compositions. The software utilizes an established statistical mechanics-based model for obtaining reaction enthalpies between network formers and modifiers by fitting to experimental structural data for simple reference compositions. The enthalpies obtained from the simple glass structures can be used to predict the structural distribution of increasingly complex glass compositions. As such, the software is a tool for building a large database of structural data. On the GitHub page, the enthalpy database will continuously be updated and external users are encouraged to contribute to the database using the GitHub pull request functionality.

In the statistical mechanics-based composition–structure model, the distribution of short-range order structures in the glass is described statistically, following a hypergeometric distribution [27]. A glass of a given composition consists of a distribution of network forming sites, and the probability for a modifier cation to interact with each network-forming site can be calculated based on the relative enthalpic (heat of formation) and entropic (temperature and population size) contributions that are specific for each network modifier–former pair. Specifically, the probability (p) of each modifier–former interaction can be calculated as a function of composition using a Wallenius type non-central hypergeometric distribution [27]:

$$p_{i,\omega} = \frac{(g_i - n_{i,\omega-1})w_i}{\sum_{i=1}^{\Omega}\sum_{j=0}^{\omega-1}(g_i - n_{i,j})w_i}, \tag{1}$$

where Ω is the total number of species, ω is an increasing absolute fraction of network modifier, g_i is the degeneracy of network forming species i, $n_{i,j}$ is the fraction of network forming species i already reacted after j attempts, and w_i is the weighting factor specific for network forming species i. The numerator in Eq. (1) is the fraction of species i before the current modifier fraction, multiplied by the i^{th} specific weighting factor. The denominator of Eq. (1) is the sum of network forming fractions before the current modifier fraction, multiplied by their respective specific weighting factors. The Boltzmann weighting factor is given as [30]:

$$w_i = e^{-\frac{H_i}{kT_f}}, \tag{2}$$

where H_i is the enthalpy of the interaction between the modifier ion and network forming species i, k is Boltzmann's constant, and T_f is the fictive temperature of the glass. Here, the fictive temperature is assumed to be the temperature at which the structure of the glass-forming liquid becomes frozen into the glassy state. In glasses annealed at their glass transition temperature (T_g), we have $T_f = T_g$. This model may be extended for any number of network formers, but the H_i parameters are specific for the given modifier–former pair. As such, assuming a known value of T_f for a given composition, the only free parameters in the statistical mechanics-based model are the H_i values.

Next, the model is extended to glass systems with multiple modifiers, with each modifier having a specific enthalpic bonding preference for each network former. A multicomponent glass therefore offers a rich bonding environment and each enthalpic and entropic contribution must be considered to calculate the bonding probabilities. To account for the modifier specific enthalpy of interaction, the parameter $H_i^{\alpha_n}$ is introduced, where α_n refers to n specific modifiers α, with a total N number of modifiers,

$$p_{i,\omega} = \sum_{n=1}^{N}[\alpha_n]\frac{(g_i - n_{i,\omega-1})e^{-\frac{H_i^{\alpha_n}}{kT_f}}}{\sum_{i=1}^{\Omega}\sum_{j=0}^{\omega-1}(g_i - n_{i,j})e^{-\frac{H_i^{\alpha_n}}{kT_f}}}, \tag{3}$$

where $[\alpha_n]$ is the relative concentration of modifier α_n and $\sum_{n=1}^{N}[\alpha_n]$ equals unity.

As shown recently [30], the model of Eq. (3) can be fitted to experimental structural ssNMR data to obtain relative H_i values for simple binary glasses, which in turn can be used to predict interaction probabilities in mixed network former glasses containing the same components. As no absolute enthalpy values are known *a priori*, all H_i values are relative to their parent network former i. As such, the weighting factor for a former–former interaction must be known when applying Eq. (3) on oxide glasses with multiple network formers. This single parameter can be obtained by fitting to structural data from a single glass system containing the two formers, with only the former–former interaction parameter as the fitting parameter [30]. For this purpose, the basin-hopping parameter optimization is chosen as the fitting procedure. This method utilizes an algorithm to change the initial parameter guesses before performing the optimization to overcome the local minimum issue [31]. Upon determining the $H_i^{\alpha_n}$ values, the composition dependent distribution of structural units in any glass can be predicted using Eq. (3). We note that the interaction probabilities p_i must be calculated numerically and the distribution of structural units must be known. Exact knowledge of both thermal history (T_f) and chemical composition are essential when applying the statistical mechanics model as even small variations in these values can lead to large deviations in the predicted structural speciation.

2. Software description

The model of Eq. (3) is incorporated in StatMechGlass, which is a Python package developed using Numpy, Scipy and Sklearn [31–33]. The package contains a main module and several submodules. The main module is to be installed to the user's own script

using the Python import command. The package comes with predefined modifier–former interaction enthalpies and former–former interaction parameters and may be used to predict glass structure from composition without any input files. In case the glass system of interest requires other interaction parameters, additional parameters can easily be input in comma-separated values (CSV) file format. Additionally, the software has a feature for building the modifier–former interaction enthalpies from glass structure data input, also supplied in CSV file format for easy use in other software.

2.1. Software architecture

The software consists of a main module, a data directory, a parameters directory and a submodules directory. Below is a short description of each vital part of the package:

- stat_mech_glass.py:
  This is the main module of the software and brings the essential functions. Once imported, the functions are callable from a Python interpreter and will communicate with the data, parameters, and module directories and scripts/files within them.
- /Data:
  This is the directory for user inputted composition–structure data. The package can use two types of data, either binary oxide glass data or ternary oxide glass data. When fitting the model on composition–structure data for binary and ternary oxide glasses, all parameters necessary for calculating the structure of glasses with any number of components are obtainable. The model could be fitted on multicomponent glasses but by keeping the input data relatively simple, the uncertainties of the obtained parameters are reduced. The subdirectories in the /Data directories are named according to the network formers of the data. For the binary oxide glasses, their names are completely written out (i.e., "SiO2", "B2O3" etc.). For the ternary oxide glass data, the subdirectories are named according to the main atoms of the forming species (i.e., "SiB", "AlSi" etc.). Within the binary oxide glass directories, structural data files are given according to the network modifier. For example, "Na.csv" within the /Data/SiO2 directory contains structural data for a Na_2O-SiO_2 glass system. An additional "M_Tg.csv" file to provide the thermal history is required when fitting binary oxide glass data, where "M" corresponds to the network modifier. For example, both "Na.csv" that contains structural data and "Na_Tg.csv" that contains T_g data are required within the /Data/SiO2 directory to obtain interaction enthalpies for a Na_2O-SiO_2 glass system. The binary oxide glass structure data files must contain the following columns: Modifier mol %, former structure *1*, former structure *2*,… former structure *N* (where *N* is the total number of possible atomic structures). For example, the "Na.csv" file within the /Data/SiO2 directory must contain the following columns: Na_2O %, Si^4 %, Si^3 %, Si^2 %, Si^1 %, Si^0 %, where the superscript refers to the number of bridging oxygens bonded to the central atom. The "Na_Tg.csv" file must contain modifier mol % and T_g values (in K): Na_2O %, T_g. Similarly, structural data for the ternary oxide glasses are given according to the network modifier. For example, "Na.csv" within the /Data/BSi directory contains structural data for a Na_2O-B_2O_3-SiO_2 glass system. In the ternary oxide glass case, T_g values must be included for each glass composition in the "M.csv" data file. The ternary oxide glass structure data files must contain the following columns: Modifier mol %, former *1* mol %, former *2* mol %, T_g, former *1* structures, former *2* structures. For example, the "Na.csv" file within the /Data/BSi directory must contain following columns: Na_2O %, B_2O_3 %, SiO_2 %, T_g, B^4, Si^4 %, Si^3 %, Si^2 %, Si^1 %, Si^0 %.
- /Parameters:
  In this directory, interaction enthalpies are stored. The subdirectories are named according to network forming species in the case of binary oxide glass parameters. /Parameters/SiO2/Na.csv contains the interaction enthalpies in the Na_2O-SiO_2 glass system, etc. Additionally, mixed former interaction parameters are located in the /Parameters/MF directory and are named according to the former–former interaction. That is, "SiB.csv" is the B_2O_3 correction in relation to the SiO_2 parameters, while "BSi.csv" is the SiO_2 correction in relation to the B_2O_3 parameters. The "BSi.csv" and "SiB.csv" parameters are inversely related.
- /stat_mech_module:
  The modules for calculating the evolution of short-range order (SRO) structural units are located in this directory. The main module will call these scripts using the appropriate enthalpy parameters and step sizes. These scripts are located in separate modules to simplify the main module and for easy expansion of the package with additional oxide components.

2.2. Software functionality

The software has four main functions to be called by the user:

- smg_binary_par:
  This function is used to establish interaction enthalpies by fitting to binary oxide glass structural data. First, place the structure data and the T_g data in the correct /Data subdirectory as described in the Software Architecture section. Then, execute the function with the appropriate input:

  smg_binary_par(former, modifier, it = 10)

  Example for the Na_2O-SiO_2 glass system:

  smg_binary_par("Si", "Na", it = 500)

  Here, "it" is the number of iterations used by the Basinhopping [31] parameter optimization algorithm. Based on our studies, 500 iterations are recommended to reach the global minimum for the parameter error space, but fewer will often suffice. The appropriate number of iterations can be obtained by tracking the lowest error found by the algorithm as a function of the iteration number.
- smg_ternary_par:
  This function is used when establishing the former–former interaction parameter in ternary modifier–former–former oxide glass systems. The script can be used to predict structures of oxide glasses containing more than two network formers, but when establishing the former–former relation parameters, only ternary glasses are required. That is, the SRO structure of a glass containing SiO_2, B_2O_3 and P_2O_5 can be predicted by combining SiO_2-B_2O_3 and SiO_2-P_2O_5 parameters. First, place the structure and T_g data in the correct /Data subdirectory as described in the Software Architecture section. Then, execute the function with the appropriate input:

  smg_ternary_par(formers, modifiers, it = 10)

  Example for the Na_2O-B_2O_3-SiO_2 glass system:

  smg_ternary_par(["B", "Si"], "Na", it = 100)

  Please address the smg_binary_par section for a description of the "it" variable.

- smg_structure:
  This function is used when predicting the SRO structural distribution in any oxide glass system with known enthalpy parameters:

  smg_structure(composition, T_g)

  Example for a $50Na_2O$-$25B_2O_3$-$25SiO_2$ glass:

  smg_structure({"Na" : 50, "B" : 25, "Si" : 25}, 700)

  Here, the composition is given using a Python dictionary and T_g is the unit of Kelvin. The total composition does not have to equal 100.
- smg_plot:
  This function is used for simple 2D plotting of the SRO structural distribution of any glass system with one free variable:

  smg_plot(composition, free component, T_g, save_plt = False)

  For example, plotting the fraction of structural units as a function of a composition variable for a $(2x)Na_2O$-$(50-x)B_2O_3$-$(50-x)SiO_2$ glass system and saving the plot as .png:

  Smg_plot({"B" : 50, "Si" : 50}, "Na", 700, save_plt = True)

  This will plot the fraction of borate and silicate structural units as a function of the sodium content. Note that the T_g for this function is set to be constant. If more specific plotting is required with composition dependent T_g values, the smg_structure function should be used to obtain the structures for manual plotting.

2.3. Common errors

Common errors when using the software are related to either the user input files or the usage of the functions. When inputting the files, all input must be numbers only and be corrected for unit conversions. Here, units are in mol% for glass components and structural units, where the sum of all structural units should be 100, while T_g should be given in the unit of Kelvin. When calling the functions, units should not be changed. Additional common errors when calling functions could include using incorrect terms as described in Section 2.2. For example, 50 mol% Na_2O should be defined as "Na":50 and not "Na2O":50.

3. Examples

example.py script is provided with one example for each of the four main functions. "Na.csv" and "Na_Tg.csv" Na_2O-SiO_2 silicate data files are provided in the Data/SiO2 directory containing SRO structural and T_g data, respectively. In the example script, the statistical mechanics-based model is fitted to the provided data to obtain the corresponding interaction enthalpies, which are automatically saved in the Parameters/SiO2 directory. When using the smg_binary_par function, the sum of squared errors is reported and an illustration of the fitting result is shown (see Fig. 1). Next in the "example.py" script, the Si-B interaction parameter can be established using the smg_ternary_par function. Then, the structure distribution in the $25Na_2O$-$25B_2O_3$-$25SiO_2$ glass composition can be calculated using the smg_structure function. Finally, the smg_plot function is used to plot the SRO structural distribution in the xNa_2O-$25B_2O_3$-$25SiO_2$ glass system as a function of x (see Fig. 2).

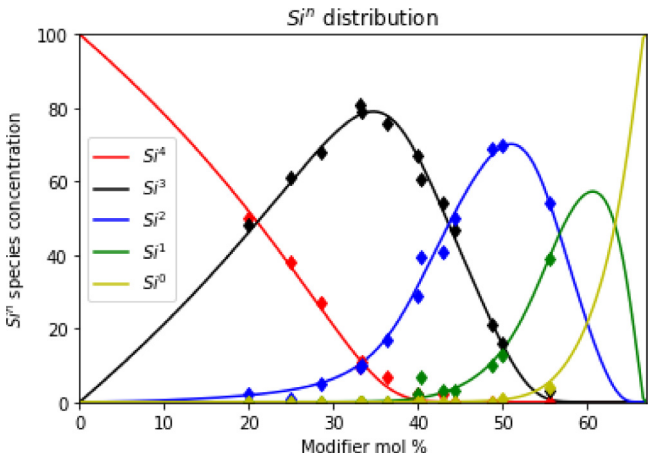

Fig. 1. Si^n distribution in binary Na_2O-SiO_2 glasses as a function of sodium modifier content as determined by using the smg_binary_par function in the software.

4. Impact

The StatMechGlass open source software provides the ability to predict and calculate the SRO structural distribution in multicomponent oxide glass systems by accounting for the enthalpic and entropic contributions to interspecies interactions using statistical mechanics. Prediction of SRO units enables a more efficient design of glasses with tailored properties, while simultaneously improving our understanding of composition–structure relations. The model has already been established and applied successfully to predict SRO glass structure distribution in various systems [28,29,34] and this software acts as a starting point for other researchers to apply the statistical mechanics-based model. The software can easily be scaled to other glass systems and pairs very well with other models, as it can produce structural data for desired glass compositions. Machine learning models have been able to predict some macroscopic properties of glasses based on composition–property training data. However, studies have found that the incorporating the missing link (i.e., structure) in the models improves the accuracy of the predictions [35,36]. No large database of SRO structures exists for oxide glasses, and as such, the present software can be used to convert a composition–property dataset to a composition–structure–property dataset. The modular design of the software allows for easy extension with additional oxide components or even to other material families exhibiting similar structural features as oxide glasses. Building upon the software by including new glass families requires that a new module for that family should be made. In this case, the same procedure as in the other modules can be used, but the family specific structure units and interactions need to be adjusted. Next, the new module in the "form_lookup" function should be included within the "stat_mech_glass" script. Moreover, the name of the family, the paths to the data specific to the family, and the functions loaded by the new module should be included in the same way as for the other families already present in the "form_lookup" function.

5. Conclusions

We report an open source software (StatMechGlass) for calculating the fraction of short-range order structural units in oxide glasses from composition and fictive temperature. StatMechGlass can build network modifier–former interaction enthalpies by fitting the underlying statistical mechanics model to available experimental composition–structure data for simple glass systems

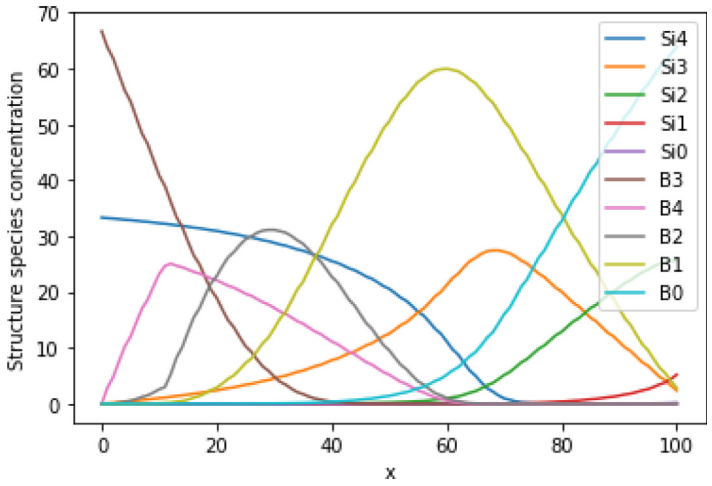

Fig. 2. Structure species distribution in the ternary xNa_2O-$25B_2O_3$-$25SiO_2$ glass system as a function of x as determined using the smg_plot function in the software.

with few components. The parameters established in the simple systems can then be used to calculate the structural distributions in multicomponent glasses by accounting for the enthalpic and entropic contributions to interspecies interactions. The modular design allows for an easy extension to other glass systems and we believe the results of the software have high potential for being coupled with machine learning models to offer an improved prediction of glass properties based on their composition and thermal history.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported by the Independent Research Fund Denmark (grant no. 7017-00019).

References

[1] Maiorov VA. Opt Spectrosc 2018;124:594–608.
[2] Ali S, Farooq I, Iqbal K. Saudi Dent J 2014;26:1–5.
[3] Baino F. Ceram Int 2018;44:14953–66.
[4] Axinte E. Mater Des 2011;32:1717–32.
[5] Mysen BO, Richet P. Silicate glasses and melts properties and structure. Elsevier; 2005.
[6] Brow RK. J Non-Cryst Solids 2000;263:1–28.
[7] Wright AC. Phys Chem Glas Eur J Glas Sci Technol Part B 2010;51:1–39.
[8] Varshneya A, Mauro JC. Fundamentals of inorganic glasses. 3rd ed.. Elsevier; 2019.
[9] Mauro JC, Tandia A, Vargheese KD, Mauro YZ, Smedskjaer MM. Chem Mater 2016;28:4267–77.
[10] Smedskjaer MM, Mauro JC, Sen S, Yue Y. Chem Mater 2010;22:5358–65.
[11] Mauro JC. Front Mater 2014;1:1–5.
[12] Butler KT, Davies DW, Cartwright H, Isayev O, Walsh A. Nature 2018;559:547–55.
[13] Liu H, Fu Z, Yang K, Xu X, Bauchy M. J Non-Cryst Solids X 2019;4:100036.
[14] Mauro JC. Am Ceram Soc Bull 2011;90:31–7.
[15] Smedskjaer MM, Hermansen C, Youngman RE. MRS Bull 2017;42:29–33.
[16] Winterstein-Beckmann A, Möncke D, Palles D, Kamitsos EI, Wondraczek L. J Non-Cryst Solids 2013;376:165–74.
[17] Youngman R. Materials (Basel) 2018;11:476.
[18] Edén M. Annu Rep Sect C Phys Chem 2012;108:177.
[19] Massobrio C, Du J, Bernasconi M, Salmon PS. Molecular dynamics simulations of disordered materials. Cham: Springer International Publishing; 2015.
[20] Norman GE, Stegailov VV. Math Model Comput Simul 2013;5:305–333.
[21] Wan S, Sinclair RC, Coveney PV. Philos Trans R Soc A 2020;1–32.
[22] Ryuo E, Wakabayashi D, Koura A, Shimojo F. Phys Rev B 2017;96:2–6.
[23] Ohkubo T, Tsuchida E, Deguchi K, Ohki S, Shimizu T, Otomo T, et al. J Am Ceram Soc 2018;101:1122–34.
[24] Hahn SH, Van Duin ACT. J Phys Chem C 2019;123:15606–17.
[25] Mauro JC, Smedskjaer MM. J. Non-Cryst Solids 2014;396–397:41–53.
[26] Goyal S, Mauro JC. J Am Ceram Soc 2018;101:1906–15.
[27] Mauro JC. J Chem Phys 2013;138. 12A522.
[28] Bødker MS, Christensen R, Sørensen LG, Østergaard MB, Youngman RE, Mauro JC, et al. J Non-Cryst Solids 2020;544:120099.
[29] Bødker MS, Youngman RE, Mauro JC, Smedskjaer MM. J Phys Chem B 2020;124:10292–9.
[30] Bødker MS, Sørensen SS, Mauro JC, Smedskjaer MM. Front Mater 2019;6:175.
[31] Jones E, Oliphant T, Peterson P. Int J Comput Sci Eng 2009;4:296–305.
[32] Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, et al. Nature 2020;585:357–62.
[33] Pedregosa F, Varoquaux G, Gramfor A. J Mach Learn Res 2014;39:i–ii.
[34] Bødker MS, Mauro JC, Goyal S, Youngman RE, Smedskjaer MM. J Phys Chem B 2018;122:7609–15.
[35] Liu H, Zhang T, Anoop Krishnan NM, Smedskjaer MM, Ryan JV, Gin S, et al. Npj Mater Degrad 2019;3:32.
[36] Bapst V, Keck T, Grabska-Barwińska A, Donner C, Cubuk ED, Schoenholz SS, et al. Nat Phys 2020;16:448–54.