A Unified Predictive and Generative Solution for Liquid Electrolyte Formulation

Zhenze Yang[1], Yifan Wu[1,**], Xu Han[2], Ziqing Zhang[2], Haoen Lai[1,**], Zhenliang Mu[2], Tianze Zheng[2], Siyuan Liu[2], Zhichen Pu[2], Zhi Wang[1], Zhiao Yu[1], Sheng Gong[1,*], Wen Yan[1,*]

[1]ByteDance Research, Bellevue, 98004, WA, USA.
[2]ByteDance Research, Beijing, 100098, China.
[**]work done as intern at ByteDance Research.
[*]corresponding: sheng.gong@bytedance.com, wen.yan@bytedance.com.

Abstract

Liquid electrolytes are critical components of next-generation energy storage systems, enabling fast ion transport, minimizing interfacial resistance, and ensuring electrochemical stability for long-term battery performance. However, measuring electrolyte properties and designing formulations remain experimentally and computationally expensive. In this work, we present a unified framework for designing liquid electrolyte formulation, integrating a forward predictive model with an inverse generative approach. Leveraging both computational and experimental data collected from literature and extensive molecular simulations, we train a predictive model capable of accurately estimating electrolyte properties from ionic conductivity to solvation structure. Our physics-informed architecture preserves permutation invariance and incorporates empirical dependencies on temperature and salt concentration, making it broadly applicable to property prediction tasks across molecular mixtures. Furthermore, we introduce—to the best of our knowledge—the first generative machine learning framework for molecular mixture design, demonstrated on electrolyte systems. This framework supports multi-condition–constrained generation, addressing the inherently multi-objective nature of materials design. This unified framework advances data-driven electrolyte design and can be readily extended to other complex chemical systems beyond electrolytes.

Keywords: Electrolyte design, Invariant neural network, Generative modeling, Molecular mixture

Introduction

Liquid electrolyte is one of the most important components in both the current commercial lithium ion batteries [1, 2] and the next-generation lithium-metal batteries [3]. Different solvent molecules and salts have been proposed as components of liquid electrolytes with corresponding advantages, such as cyclic carbonates with strong solvating power to Li^+, linear carbonates, esters and ethers with relatively low viscosity and moderate solvating power [1, 2], and various fluorinated molecules for LiF-dominated solid-electrolyte-interphase (SEI) [4, 5]. In this context, the formulation of electrolytes plays a critical role in optimizing battery performance [6], yet developing novel formulations remains challenging due to the vast chemical space of electrolyte molecules and the intractable number of possible molecular combinations. Despite extensive research, experimental measurements for key properties such as ionic conductivity and coulombic efficiency remain sparse, with approximately 10,000 conductivity data points available across various public datasets [7–9] and only a few hundred for coulombic efficiency [10]. These limited data cover only a small fraction of the compositional space of potential electrolytes, significantly hindering the efficient exploration of novel formulations.

To supplement experimental measurements, MD simulations using classic force fields such as OPLS series [11, 12] have been widely employed to estimate liquid electrolyte properties. Despite the low computational cost, these classical force fields often struggle to achieve satisfactory prediction accuracy for properties of liquid electrolytes [13], due to the simplicity of the functional forms of classic force fields. While machine learning force fields (MLFFs) [13–15] might offer improved accuracy over classic force fields, the computational cost of MLFF remains significantly higher, making MLFF-based large-scale screening impractical.

Recent advancements in data-driven machine learning (ML) approaches have shown promise in directly predicting electrolyte properties from molecular mixtures. Several studies have successfully employed deep learning (DL) architectures, such as graph neural networks (GNNs) and attention-based models, to achieve reasonable accuracy, demonstrating that ML models can extract meaningful patterns from limited experimental datasets. For instance, Zhang et al. [16] proposed the "MolSet" model for predicting the conductivity of lithium battery electrolytes, while Zeng et al. [17] leveraged multi-level information to capture various electrolyte formulation properties. However, these models still face two major limitations: limited formulation coverage and lack of physical constraints. For instance, the "MolSet" work is limited to handling mixtures containing up to four different molecule types and primarily focuses on polymer electrolytes, and that from Zeng et al. [17] does not include physical constraints, which might lead to occurrences of unphysical results for property prediction. Besides experimental data, Chew et al. [18] also utilized high-throughput molecular simulations together with ML models for property predictions of chemical mixtures. However, a significant gap remains between the results of molecular simulations and experimentally measured bulk properties.

In addition to the latest progress in ML-based property prediction models for liquid electrolytes—often regarded as a "forward problem"—an equally practical and impactful challenge lies in the "inverse problem": how to efficiently explore and design

novel electrolyte formulations that satisfy specific property requirements. This task becomes particularly challenging given the vast design space of electrolyte formulations. For instance, given a formulation where eight components are selected from a pool of 100 candidate molecules, and the molar ratios of each molecule range from 0 to 1 with an interval of 0.05, the number of possible combinations reaches approximately $C_{100}^{8} \times C_{27}^{7} \approx 1.65 \times 10^{17}$ (combinations of molar ratios can be reformulated as a non-integer composition problem). The diversity of molecular species, combined with the variability in molar ratios, results in an expansive design space for electrolyte formulations. In the current commercial electrolytes, electrolyte formulations with more than five, and in some cases, up to ten different component molecules [19, 20] are commonly used to meet the diverse performance requirements of a commercial battery cell. Electrolytes composed of molecular mixtures often exhibit enhanced properties, such as higher conductivity and coulombic efficiency, compared to single-molecule systems [6]. Moreover, the concept of high-entropy electrolytes, which are composed of a diverse set of solvents and salts, is gaining increasing attention for their potentially high ionic conductivity, cycling stability, and rate capacity [21–23]. These examples highlight the vast design space of electrolyte formulations, emphasizing the need for efficient exploration methods. However, most existing data-driven studies for liquid electrolyte focus on local optimization rather than global exploration. For instance, Zhu et al. [24] optimized only ternary mixtures, while Zeng et al. [17] primarily screened single solvent molecules to boost conductivity of electrolytes. As a result, these models, while effective in solving their respective designated tasks, do not naturally scale to the full complexity of multi-component electrolyte formulation.

To address these challenges, generative models offer a compelling alternative for systematically navigating the electrolyte design space. Unlike brute-force screening methods or local optimization approaches, generative approaches can propose novel electrolyte formulations or molecular mixtures by directly learning from the data distribution, making them well-suited for high-dimensional search spaces [25, 26]. **However, to the best of our knowledge, no prior work has applied generative modeling to molecular mixture generation, letting alone specific property-guided electrolyte generation.** This is in stark contrast to other material design fields, such as molecular discovery [27–29] and crystal structure generation [30–32], where generative models have been widely adopted. This gap exists because of several fundamental challenges:

- **Large and Structured Design Space** — Molecular mixture design involves a vast chemical space due to the diversity of molecular species and the wide range of possible stoichiometric mixing ratios. This combinatorial complexity significantly enlarges the design space compared to single-component systems.
- **Complex Interactions** — Even given a fixed set of chemicals and a specific mixing ratio, accurately simulating mixture properties remains highly challenging due to intricate intermolecular and intramolecular interactions. As a result, data-driven predictions often rely heavily on experimental measurements.
- **Data Scarcity** — Despite their importance, open-sourced experimental data for molecular mixtures are extremely limited due to the high cost and time required

for synthesis and characterization, as well as the commercial interest of molecular mixtures. Computational data is also expensive to obtain at scale.

- **Representation and Inductive Bias** — Effective methods like ML models for molecular mixtures need to incorporate appropriate inductive biases and representations that reflect the underlying physics, such as permutation invariance with respect to molecular ordering, to address the challenges of scale, complexity, and data scarcity.

These challenges highlight the need for a novel approach that integrates physics-aware representations [33], data-driven conditional generative modeling, and efficient evaluation strategies. In this work, we develop a unified framework for electrolyte formulation design, integrating a forward predictive model and an inverse generative approach. The main contributions of this paper are summarized as follows:

- We extensively collect literature data for both single molecules (240,000+) and molecular mixtures (10,000+) with labeled properties, enabling broad coverage of the electrolyte design space. By further integrating over 100,000 molecular mixture data points generated from molecular dynamics (MD) simulations, we are able to train an accurate ML model for not only conductivity prediction, but also solvation structure estimation which might relate to interfacial stability of Li metal batteries.
- We propose a physics-informed architecture that preserves permutation invariance and incorporates empirical dependencies on temperature and salt concentration for accurate property prediction. This approach serves as a universal framework applicable to a wide range of prediction and design tasks involving molecular mixtures, extending beyond electrolyte formulations.
- We develop the first generative framework for molecular mixture design, using electrolyte systems as a representative example. We introduce a multi-condition–constrained generative approach, providing a promising solution for multi-objective materials design.

Results

Overall workflow

The overall workflow of this study is illustrated in Fig. 1a and comprises two key components: a forward predictive model and an inverse generative process. The predictive model estimates two target properties—ionic conductivity and coordination ratio of anion around Li^+ (abbreviated as anion ratio below)—based on various electrolyte formulations. In contrast, the generative process designs new electrolyte formulations conditioned on desired property values. Both components follow three-stage training and evaluation procedures, described in detail below.

In the forward prediction process, we first employ a GNN model to learn a universal molecular embedding (also referred to as a "fingerprint" or "descriptor") for each molecule within the electrolyte system. This embedding is learned through a multi-task pretraining strategy, using a large single-molecule dataset (over 240,000 entries) to predict various molecular properties such as melting point (Process (1) in Fig. 1b). Next, the molecular embeddings of electrolyte components are aggregated—weighted

in a learnable way by their molar ratios—into a single electrolyte-level embedding using a permutation-invariant neural network. To broaden coverage of the formulation space, this "Invariant Aggregation" model is pretrained on over 100,000 electrolyte systems calculated via MD simulations (Process (2) in Fig. 1b). Finally, the model is fine-tuned on more than 10,000 experimental electrolyte data points (62 solvents and 17 Li salts) and incorporates empirical relations with physical priors to enhance conductivity prediction accuracy (Process (3) in Fig. 1b).

The generation process can be viewed as the inverse of the prediction task. We begin by training a conditional diffusion model to generate electrolyte embeddings given specified property targets—namely, anion ratio and conductivity (Process (1) in Fig. 1c). These generated electrolyte embeddings are then converted back to a set of molecular embeddings and corresponding molar ratios using a trainable decoder model (Process (2) in Fig. 1c). Finally, the generated molecular embeddings are matched to specific molecules in our electrolyte database based on the distance of embeddings (Process (3) in Fig. 1c). The selected molecules and their corresponding molar ratios together define the final generated electrolyte formulation.

The detailed model architectures within the whole workflow are visualized in Fig. S1 and data representations of molecules and electrolytes are summarized in Table S1. More information will be discussed in the following sections and Supplementary Information.

Electrolyte property prediction

In this section, we discuss in detail about the methodologies and results of the forward predictive model in our workflow.

Molecular pretraining

To obtain a comprehensive and efficient representation of diverse molecules within the electrolyte formulations, we pretrained a GNN model to learn a universal "fingerprint" by predicting various molecular properties with more than 200,000 molecular data. These properties include 11 distinct entries: melting point (T_m), boiling point (T_b), (liquid) refractive index (n_D/n_D^{liquid}), pK_a, pK_b, dielectric constant (ε), surface tension (γ_s), density (ρ), viscosity (η) and vapor pressure (P_{vap}). The GNN model takes atomic and bond features derived from the chemical SMILES representation [34] as input and outputs a molecular embedding using a modified Edge-augmented Graph Transformer (EGT) model [35, 36]. This embedding is then processed through separate readout blocks with multiple multilayer perceptrons (MLPs), each dedicated to one respective molecular properties.

The results of molecular pretraining are shown in Fig. 2a. We compare our GNN architecture (labeled as "Atom/bond feature + EGT") with two baseline methods: "Morgan fingerprint + NN" and "Atom/bond feature + GAT" (see the Methods section for details). Our GNN model consistently outperforms both baselines in the prediction of all molecular properties and achieves high R^2 scores across these tasks (see also in Fig. S2). These findings suggest that the molecular embeddings learned

during pretraining encode rich chemical information, enabling their effective application in downstream tasks on electrolytes, which are mixtures of the constituent molecules.

Physics-informed architecture

With a comprehensive descriptor for each individual molecule, the next step is to understand how they collectively influence the properties of liquid electrolyte systems. To this end, we developed an invariant aggregation block, followed by an empirical equation block, to integrate molecular information and predict electrolyte properties with physical prior.

More specifically, our model incorporates two key physical constraints: permutation invariance and the dependence of conductivity on temperature and concentration. Unlike predictive or generative tasks involving a single molecule, an electrolyte formulation is a mixture of multiple molecules, and its properties should remain invariant under permutations of its components. To achieve this, we employ a self-attention-based model that performs a learnable, weighted aggregation of molecular embeddings, guided by the molar ratios of the constituent molecules in the formulation (see Methods and Supporting Information for more details). The aggregation block produces an electrolyte-level embedding, which is then used to predict the parameters of our empirical equation that models the dependence of conductivity on temperature and concentration.

In terms of empirical relation, dependence of electrolyte conductivity on temperature and salt concentration has been extensively studied, with a large number of theoretical or empirical models [37–41]. Some of these general trends are well-known. For instance, conductivity typically increases initially and then decreases with increasing salt concentration due to the trade-off between the number of charge carriers and the increasing viscosity. In addition, conductivity increases with rising temperature due to enhanced ion mobility. To incorporate these domain knowledge, prevent unphysical prediction (Fig. S3) and enhance generalizability, we employed an empirical equation (Eq. 4) for the temperature and concentration dependence of conductivity based on previous studies [40, 41] and our evaluation (Fig. S4, see Methods section and Supporting Information for the derivation, mathematical form and ablation study of the empirical equation).

With the empirical equation, model inference is no longer required each time the temperature or concentration change. Instead, the entire temperature and concentration curve for an electrolyte system can be obtained with one-time inference of our model, which dramatically improves the efficiency of conductivity prediction. The peak conductivity and its corresponding salt concentration can also be directly computed using the empirical relation. In addition, we also incorporated viscosity in the empirical relation as conductivity generally increases with decreasing viscosity. We utilized an inverse relation between conductivity and viscosity described by the well-known Walden's rule [42]. The incorporation of viscosity into the empirical relation further enhances the generalizability of our model using the viscosity as a prior for conductivity prediction (see Table S2, data split based on viscosity).

Pretraining with computational data

To enable broader coverage of the electrolyte design space and capture solvation characteristics, we pretrained our electrolyte-level model using data generated from MD simulations (denoted as "computational pretraining"). Specifically, we conducted over 100,000 all-atom MD simulations using the OPLS force field [11, 12] to obtain key electrolyte properties, including ionic conductivity and anion ratio. High ionic conductivity enables rapid ion transport, which minimizes internal resistance, reduces energy loss, and supports high power output and fast charging. However, a well-known trade-off exists between ionic conductivity and interfacial stability of liquid electrolytes used in lithium-based batteries. This trade-off arises because organic solvents that provide high ionic conductivity—such as carbonates or ethers—tend to have low electrochemical and chemical stability, especially when in contact with reactive electrode materials like lithium metal or high-voltage cathodes [43, 44]. Recent works addressed this challenge in the lithium-metal battery by reducing the free solvent molecules within the Li^+ solvation structure (known as "weak solvation"), leading to a predominantly inorganic SEI for better Li cyclability [45–49]. Inspired by these studies, we use the anion ratio—which correlates positively with Li cyclability—as an additional objective alongside conductivity. Here, we define the anion ratio as the number of anions divided by the total number of anions and solvent molecules in the Li ion's first solvation shell. A larger anion ratio results in weaker solvation of solvents, thereby generally enhancing the battery's interfacial stability.

Given the extensive MD data, computational pretraining not only allows us to obtain relevant information to Li cyclability, but also provides broader coverage of the electrolyte formulation design space—particularly for multi-component systems. Experimental studies in the literature typically focus on liquid electrolytes with only a few solvents and a single type of Li salt. In addition, incorporating a new molecule into experiments is generally more costly—and sometimes impractical—compared to MD simulations. Consequently, computational pretraining serves as a valuable tool to improve predictive accuracy when exploring novel chemical candidates for electrolyte design as we show in Table S2.

To exploit diverse information from MD simulations, pretraining is conducted in a multi-task learning manner, simultaneously predicting anion ratio and ionic conductivity. The conductivities are obtained from two different methods: Mistry's method [50] (denoted as "mistry") and Nernst-Einstein (denoted as "NE") method (see Methods section and Supplementary Information for more details). Although there are some simplifications in these two methods (NE ignores the correlation between ions and Mistry's method calculates chemical potential under dilute assumption), we observe positive correlation between MD-calculated and experimental conductivities and consequently believe that MD data remains useful for the pretraining task (Fig. S5). The prediction results of these three properties are plotted in Fig. S6. We visualized the correlation between anion ratio and conductivity derived from MD data in Fig. 2b, which displays a clear trade-off between these two properties.

Fine-tuning with experimental data

Following computational pretraining on extensive MD simulation data, we further fine-tune our model using over 10,000 experimentally measured conductivity values collected from the scientific literature (denoted as "experimental fine-tuning"). During fine-tuning, the molecular embeddings learned from pretraining are kept fixed, while the electrolyte embeddings are updated. Since experimental data for anion ratio are not available, we continue to use MD-derived anion ratios during this stage. For conductivity prediction, our model estimates the electrolyte embedding-dependent parameters of the empirical relation. Conversely, because no empirical relation exists for anion ratio, we concatenate the electrolyte embedding with temperature and concentration and employ a readout layer to predict the property. As shown in Fig. 2c, the fine-tuned model accurately predicts both experimental ionic conductivity and computational anion ratio, achieving $R^2 = 0.985$ for conductivity and $R^2 = 0.953$ for anion ratio.

We further examine the temperature and concentration dependence of conductivity predicted by the model. As Fig. 2d reveals, the model precisely reproduces both trends across a wide range of electrolyte systems. The temperature dependence typically follows a linear relationship between log-scale conductivity and $\frac{1}{T-T_0}$, where T_0 is a system-dependent parameter generally associated with the glass transition temperature of the electrolyte [39]. In contrast, the concentration dependence exhibits a characteristic "volcano" plot, with conductivity first increasing and then decreasing as concentration changes. These trends are rigorously enforced through the incorporation of our empirical relation. More examples of temperature and concentration dependence of conductivity can be found in Fig. S7. These results demonstrate that our adapted empirical relations are applicable across different electrolyte systems and that our model accurately captures the effects of experimental conditions.

Generative electrolyte design

In this section, we discuss in detail about the methodologies and results of the inverse generative model in our workflow.

Conditional generation

The forward predictive model enables rapid evaluation of electrolyte properties, making it an efficient tool for screening electrolyte formulations. However, as the candidate space grows exponentially as the number of molecule species in the mixture formulation, brute-force sequential screening becomes intractable to design electrolyte systems with target properties. As an alternative, we leverage a diffusion model, to conditionally generate electrolyte formulations that satisfy specific property targets.

We utilize a synthetic dataset generated by our predictive model to train our conditional diffusion model. The conditional diffusion model takes conductivity and anion ratio as constraints and generates electrolyte embeddings from random Gaussian noises (see Methods). The generated electrolyte embeddings are converted back to molecular embeddings using a decoder module (Fig. S1e). Each molecular embedding is then

matched to molecules in our electrolyte database based on a minimum distance criterion (see Supplementary Information for more details). Given a fixed set of molecular species, an electrolyte formulation can be represented as a normalized vector, referred to as the "Bag of Molecules" (BoM) vector. Each dimension of the "BoM" vector corresponds to the molar ratio of the molecule at that position. The concept is similar as the "Bag of Words" representation in Natural Language Processing (NLP). With the formulation representation, we further utilized our predictive model to evaluate the properties of generated electrolyte formulations.

The results of conditional generation for electrolyte formulations are plotted in Fig. 3. As shown by Fig. 3a, the generative model enables the design of electrolyte formulations with extrapolated properties in both conductivity and anion ratio. By conditioning the diffusion model on high conductivity or anion ratio values, the resulting data distribution is significantly shifted toward the desired property targets. In addition, we highlight that the model can generate electrolyte formulations that simultaneously satisfy both property targets. For example, under the condition of 10.0 mS/cm conductivity and an anion ratio of 0.3, the model's output is shown in Fig. 3b, and a representative formulation appears in Fig. 3c. We further evaluate conditional generation across conductivity constraints ranging from 5 to 30 mS/cm and anion ratios from 0.1 to 0.7; these results are visualized in Fig. S8, demonstrating the model's versatility. More generated samples under various conditions can be found in Fig. S9 through Fig. S13. These results demonstrate the strong capability of the generative model, which can largely enhance the efficiency of designing electrolytes with target properties.

Performance evaluation

To further quantify the performance of generation, we proposed three different evaluation metrics including one accuracy score (mean absolute percentage error, MAPE hereafter) and two diversity scores. The MAPE score is calculated by summarizing the relative error from both conductivity and anion ratio. In contrast, the two diversity metrics assess the similarity among generated formulations to help prevent mode collapse, a common issue in generative modeling. These two diversity scores highlight different aspects: formulation diversity measures the average pairwise distance between closest formulations within the generated set, while molecular diversity evaluates the entropy of molecular occurrence across generated samples (Mathematical definitions of diversity scores are discussed in Supporting Information).

With these evaluation metrics, we show that in Fig. 3d, both accuracy and diversity are higher when the generation conditions are close to the original labeled data distribution. In our synthetic dataset, the mean conductivity is approximately 5.57 mS/cm and the mean anion ratio is 0.269. Accordingly, generation conditioned on conductivity = 5.0 mS/cm and anion ratio = 0.3 yields low error and high diversity. Conversely, when the target properties lie at the tails of the data distribution (e.g. conductivity = 30.0 mS/cm and anion ratio = 0.7), the model tends to produce electrolyte formulations with larger deviations from the target properties and reduced diversity. In addition, there is a general trade-off between the generation accuracy and diversity shown by Fig. 3d.

Electrolyte generation under base formulation constraints

In addition to the two target properties of interest, incorporating a base formulation as a constraint is also one of the common practices in electrolyte formulation design. A base formulation poses constraints on the molar ratios of certain molecules. The motivations for this are as follows:

- Incorporating base formulation constraints helps reduce the design space of electrolyte formulations, thereby improving the efficiency of the design process.
- Practical battery design often requires electrolytes to meet a variety of additional constraints. By using a base formulation, we can integrate prior design knowledge beyond the specific conditions imposed on the diffusion model. For example, the molar ratio of ethylene carbonate (EC) in the solvent mixture is typically maintained above 20% to enhance the solubility of Li salts, while fluoroethylene carbonate (FEC) is usually kept below 10% in industrial applications due to cost considerations.
- This approach becomes especially important when certain property requirements—such as electrochemical stability window, interfacial compatibility, or thermal stability—are expensive and labor-intensive to measure. Consequently, the available data for training ML models can be limited. Applying base formulation constraints can help mitigate this issue by guiding the generation process within a more informed and feasible region of the design space.

Therefore, we here realize the conditional generation with the base formulation constraints by utilizing a classifier-guided diffusion (CGD) approach [51](Details can be found in Methods and Supporting Information). Specifically, during the sampling process, we use our decoder module as a classifier to convert noisy electrolyte embeddings into "BoM" vectors, which are then evaluated to determine whether the generated formulations satisfy the base formulation constraints. The gradient from the classifier is subsequently used to guide the denoising process toward the desired base formulation constraints. A schematic of this classifier-guided sampling is displayed in Fig. 3e. This approach offers several advantages over conventional classifier-guided or classifier-free diffusion methods. Above all, the decoder is computationally inexpensive, so incorporating gradient calculations during sampling does not really increase the overall sampling time. Additionally, there is no need to retrain the diffusion model or classifier each time a new constraint is introduced. The decoder module only needs to be trained once, and adapting to different base formulation constraints simply requires redefining the classification criteria on the decoded formulations. This significantly improves the efficiency and flexibility of the generation process.

We tested our CGD approach on two different base formulation constraints: (1) EC $>$ 20%; (2) EC $>$ 20% & DMC (dimethyl carbonate) $>$ 20% & EMC (ethyl methyl carbonate) $>$ 20%. For conditional generation, the target conductivity was set to 10 mS/cm and the anion ratio to 0.25, and more than 10,000 samples are generated for evaluation. As shown in Fig. 3f, our CGD method significantly improves the success rate of electrolyte formulation generation compared to the case without classifier guidance. For case (1), the success rate is improved by a factor of 30, while for case (2), it is

enhanced by at least three orders of magnitude (without CGD method, there is no single formulation meets the condition among more than 10,000 generated samples). The success rate accounts not only for satisfying the base formulation constraints, but also for generating electrolytes with conductivity and anion ratio values close to the specified targets. These results demonstrate that CGD can effectively guide the generative process toward realistic, multi-constraint electrolyte formulations with high fidelity.

Discussion

In summary, this study presents a unified framework that integrates both predictive and generative tools for electrolyte formulation. By leveraging a combination of computational and experimental data—spanning single molecules to molecular mixtures—the predictive model enables accurate property estimation from the atomic scale to the formulation level. Incorporating physics-informed priors, the model's prediction not only preserves permutation invariance, but also observes the typical temperature and concentration dependencies of ionic conductivity observed in electrolyte systems. Furthermore, through generative modeling, we significantly enhance the efficiency of electrolyte design and enable multi-objective generation, allowing simultaneous control over key formulation properties and constraints. This capability is particularly valuable, as practical electrolytes often consist of complex mixtures of multiple component molecules and must meet a range of performance criteria. Our approach provides a scalable and flexible framework to navigate this multi-dimensional design space more effectively.

However, there remains substantial future work to address the current limitations of the workflow. For instance, given the limited data availability—such as only a few hundred data points for Coulombic efficiency-we use anion ratio as an indirect proxy for evaluating the interfacial stability of electrolytes. This property is not as direct an indicator of battery cyclability as Coulombic efficiency. However, as data of battery performance becomes increasingly accessible, this framework can be readily extended to incorporate additional performance metrics, such as electrochemical stability window, Li-ion transference number, and thermal stability [52, 53], offering more direct insights into electrolyte behaviors within battery systems.

In addition, while the current workflow operates within a fixed molecular space, our model has the capacity to extend to a much broader chemical space. This is made possible by the decoder architecture, which recovers molecular embeddings before molecule matching. These embeddings can either be matched against a larger chemical database to identify promising candidates, or used as input to a molecular decoder trained to reconstruct molecular structures in the form of SMILES strings or 2D molecular graphs for molecular discovery [54, 55].

Last but not least, the methodology developed in this work for property prediction and conditional generation of liquid electrolytes with target properties is potentially transferable to other systems composed of mixtures of chemical species. These systems include but not limited to fossil fuels [56], biomolecular aggregates [57], high-entropy alloys [58], polymer blends [59], ionic liquids [60], deep eutectic solvents [61], and multi-component catalysts [62]. Given that current research efforts in the overall AI4Science

field largely focus on the prediction and generation of mono-material systems—such as protein monomers, single crystals, or individual molecules—we believe there is substantial opportunity to adapt our workflow to a wide range of complex, multi-component material systems in the future.

Methods

Molecular pretraining

The single-molecule dataset used for molecular pretraining was compiled from various public sources (Table S3) and carefully screened to eliminate ambiguous or incorrect data. The final dataset comprises a total of 189,528 unique molecules with 241,414 entries under different temperatures. Among 11 molecular properties we collect, density, refractive index, dielectric constant, surface tension, viscosity and vapor pressure are temperature-dependent (More details can be found in the Supplementary Information). For temperature-dependent properties, each unique temperature-property pair is treated as an independent data point for training.

For predicting molecular properties, we implement a GNN architecture, closely following the graph block design of the ByteFF model [35]. This architecture leverages both atom features (element type, ring connectivity, minimum ring size, and formal charge) and bond features (bond order and ring membership) as molecular input representations. To effectively create a molecular descriptor, a modified EGT is incorporated within the GNN, enabling the model to learn node and edge embeddings from atomic and bonding information. The final molecular embedding is constructed by concatenating the node and edge embeddings generated by the EGT block. Further in multi-property prediction, the universal molecular embedding is processed through separate readout blocks, establishing a multi-task learning framework that enables simultaneous prediction of multiple molecular properties. For those properties that are temperature-dependent, we adopt different empirical relations and use readout layers to predict parameters in these equations to obtain corresponding properties (see Supporting Information for more details). To benchmark the performance of this GNN-based approach and obtain an effective molecular embedding, we compare the performance with two baseline models including: (1) a simple feed forward neural network using Morgan fingerprint [63] as input ("Morgan fingerprint + NN") or (2) a Graph Attention Network (GAT) [64] as an alternative to EGT using node/atom features as input ("Atom/bond feature + GAT"). As shown in Table S4, the selected EGT model outperforms the two baseline models, achieving the highest accuracy across nearly all property prediction tasks.

Permutation-invariant electrolyte representation

To achieve permutation invariance, we design an electrolyte representation for both property prediction and electrolyte design using a multi-head self-attention-based aggregation mechanism [65] (see Fig. S1c). The aggregation mechanism is adapted from Zhang et al. [66]'s work with two key modifications. First, before passing the

molecular embedding through the attention blocks, it is scaled by the molar ratio to ensure that molecules with a ratio of zero do not contribute to the mixture embedding. Second, multi-head attention is incorporated to enhance the model's expressiveness.

More specifically, the permutation invariance is achieved with the following aggregation mechanism:

$$\boldsymbol{X} = [\boldsymbol{X_1}, \boldsymbol{X_2}, ...]^T, \tag{1}$$

$$\boldsymbol{r} = [r_1, r_2, ...]^T, \tag{2}$$

$$\text{Aggr}(\boldsymbol{X}, \boldsymbol{r}) = \boldsymbol{r}^T \cdot \text{MultiHeadAttention}(\boldsymbol{r} \odot \boldsymbol{X}) \tag{3}$$

where $\boldsymbol{X}$ is the tensor that stores molecular embedding, and $\boldsymbol{r}$ is a vector with molar ratio information. "$\cdot$" is inner product and "$\odot$" is row-wise multiplication. The aggregation mechanism is permutationally invariant, as the inner product operation gathers information independent of the ordering of molecules.

Empirical temperature and concentration dependence of conductivity

We incorporate prior domain knowledge into conductivity prediction through an empirical relation. The relation used in this work is described by the following equation:

$$\sigma(T, c) = \frac{A}{\eta} c^{n_1} e^{-\frac{B \times c^{n_2} + D}{T - T_0}} \tag{4}$$

where A, n_1, B, n_2, D and T_0 are all learnable parameters from electrolyte embeddings but independent from temperature (T) and Li salt concentration (c, Li molar ratio). η is the estimated viscosity of the solvent mixture based on predicted viscosities from molecule pretraining (see Supplementary Information). The inclusion of viscosity in the prediction of conductivity is inspired by the well-known Walden's rule [67], which describes the inverse relation between conductivity and viscosity. The empirical relation in Eq. 4 is further validated using data from both the Advanced Electrolyte Model (AEM) [24, 68, 69] and experiments [70]. As Fig. S4 and Table S5 indicate, the chosen empirical relation fits well across various electrolyte systems, and the low loss from our predictive model further demonstrates the generality of this relation.

Pretraining with computational data

The computational dataset for pretraining the predictive model was randomly sampled within the chemical species of all solvent molecules in the experimental dataset, together with the electrolyte formulations in the experimental dataset. During the sampling, the maximum number of solvents was set to six, and $LiPF_6$ and LiFSI were chosen as the salts given their common usage in commercialized electrolytes. In total, there are 104,657 formulations with their MD simulated anion coordination ratio and ionic conductivities calculated by the Nernst-Einstein method and a method described

by Mistry et al. [50] in the computational dataset. Details of the MD simulations are provided in the Supporting Information.

As briefly mentioned in the main text, the pretraining with computational data was performed to predict both conductivities—derived from the NE and Mistry's methods—as well as the anion ratio. Conductivities are predicted using an empirical relation, while the anion ratio is obtained directly through an MLP readout block. The readout block receives the electrolyte embedding, combined with temperature and concentration, as input and produces the corresponding prediction. During computational pretraining, the molecular embeddings obtained from molecular pretraining are kept frozen. Only the aggregation block used to generate the electrolyte embedding, along with the readout blocks for property prediction, remain trainable.

Fine-tuning with experimental data

Given the inaccuracy of MD-derived conductivity, we further fine-tune our predictive model with experimental conductivity values. The experimental data used for fine-tuning contains 10,407 entries with 62 types of solvents and 17 types of Li salts. The data were collected from online databases and public publications (about 50 publications in total). In combination with 104,657 anion ratio samples obtained from MD simulations, the model is fine-tuned to jointly predict experimental conductivity and computational anion ratio. As in computational pretraining, the molecular embeddings remain fixed, while the electrolyte embeddings are trainable during fine-tuning. More technical details of fine-tuning can be found in Supporting Information.

Conditional generation

The conditional diffusion model we utilize in this work is based on Denoising Diffusion Probabilistic Models (DDPM) [71] to generates invariant electrolyte embeddings given target conductivity and anion ratio. DDPMs are a class of generative models that synthesize data by learning to reverse a diffusion process, where noise is progressively added to data and then removed step-by-step. This approach has proven highly effective in generating high-quality, diverse samples across various domains. To realize conditional generation, additional labels are processed as embeddings besides time embeddings to modulate the noisy data during both diffusion and denoising steps (see Supporting Information for more details).

In the next step, the generated electrolyte embeddings are decoded into a set of molecular embeddings using a Set Transformer–inspired architecture [72], which preserves permutation invariance throughout the decoding process. Although not explored in this work, these molecular embeddings could potentially be further converted into 2D molecular graphs or SMILES strings by training an additional molecular decoder, enabling the discovery of new electrolyte molecules. For simplicity, we currently match the generated embeddings to existing molecules in our database. As a result, each electrolyte formulation can be represented as a "BoM" vector (see Supporting Information for technical details). Each value within the vector represents the corresponding molar ratio of the molecule at the position.

To train a conditional diffusion model, we first used our predictive model to synthetically create 30,000 different electrolyte formulations with conductivity and anion ratio labels. The synthetic dataset offers a more balanced coverage of the electrolyte design space compared to the experimental dataset and provides improved accuracy in conductivity evaluation compared to the computational dataset. All 62 solvents and 2 Li salts ($LiPF_6$ and LiTFSI) from experiments are randomly sampled to compose different formulations. The molar ratios of solvents and salts sum up to 1 with fixed salt concentration (10 mol %) and temperature (25°C). We select 2 Li salts as they are most commonly used in today's commercialized lithium-ion battery. As a result, our generative modeling primarily focuses on the solvent design space, which is typically the main target in electrolyte design. Details about the synthetic dataset are included in the Supporting Information.

Classifier-guided diffusion

To generate electrolyte formulations with base molecules and constrained molar ratios, we employ a classifier-guided diffusion (CGD) method [51]. The essential idea is that our electrolyte decoder already generates the "BoM" vector, which can be directly used to assess whether the generated formulation contains certain base molecules. This allows us to construct a classifier function based on the decoder for any given base-molecule scenario without the need to retrain the classifier for each condition. As a result, this approach significantly speeds up the adaptation of our model to new cases compared to approaches like classifier-free guidance. The CGD method guides the sampling towards target base formulation constraint by computing the gradient of the classifier's log-probability with respect to noisy data at time t (see Supporting Information for more technical details). Consequently, the gradient-based sampling approach guides the generation process toward satisfying the target constraint.

In order to better predict the "BoM" vector from noisy data during sampling, we retrain the decoder on noisy electrolyte embeddings for the CGD generation. These embeddings are generated as the noisy data obtained in the forward diffusion process. An additional hyperparameter, gradient scale, is multiplied to the gradient to control the degree of guidance (see Supporting Information for the usage and selection of gradient scale). We utilize the maximum success rate under different gradient scales for visualization in Fig. 3f.

Performance evaluation for formulation generation

To evaluate the performance of our generative process, we proposed three different evaluation metrics: (1) MAPE which evaluates the deviation of generated properties from the targets; (2) "formulation diversity" which calculates the average minimal pairwise difference between generated formulation; (3) "molecular diversity" which computes the entropy given the occurrence frequency of each molecule within the generated formulations. Together, these evaluation metrics offer insights into the accuracy of the generated formulations relative to the target, as well as their diversity in terms of molar ratio variation and molecular species composition. Mathematical expressions of all three metrics can be found in Supporting Information.

All evaluations of the generative model's performance are conducted on 10,240 generated electrolyte formulations for each case, produced in 40 batches with a batch size of 256. For evaluating the success rate of base formulation–constrained generation, we define a generated formulation as a successful generation if it satisfies the following criteria: (1) it meets the specified base formulation constraints of our target (e.g., the molar ratio of EC exceeds 20%); and (2) the deviations in both conductivity and anion ratio of generated formulations are smaller than the corresponding standard deviations of those properties in the labeled dataset. The final success rate is calculated by dividing the number of generated formulations that meet these criteria by the total number of generated samples.

Supplementary information. Additional explanations and details regarding the datasets, model architectures, training and evalution approaches, and results can be found in the supplementary information.

Declarations

Conflicts of interest

ByteDance Inc. holds intellectual property rights pertinent to the research presented herein.

Author contribution

Conceptualization: Zhenze.Y., S.G. and W.Y.; Methodology: Zhenze.Y., S.G., Y.W., W.Y., X.H., Z.Z. and H.L.; Investigation: Zhenze.Y. Y.W., X.H., Z.Z., H.L., Z.M., T.Z., S.L., Z.P., Z.W., Zhiao.Y., S.G., W.Y.; Supervision: S.G. and W.Y.; Writing: Zhenze.Y., S.G. and W.Y.

Data and code availability

The computational, experimental and literature datasets, both predictive and generative models to reproduce the results in the paper, as well as the source codes, are provided in the referenced link (https://github.com/bytedance/bamboo).

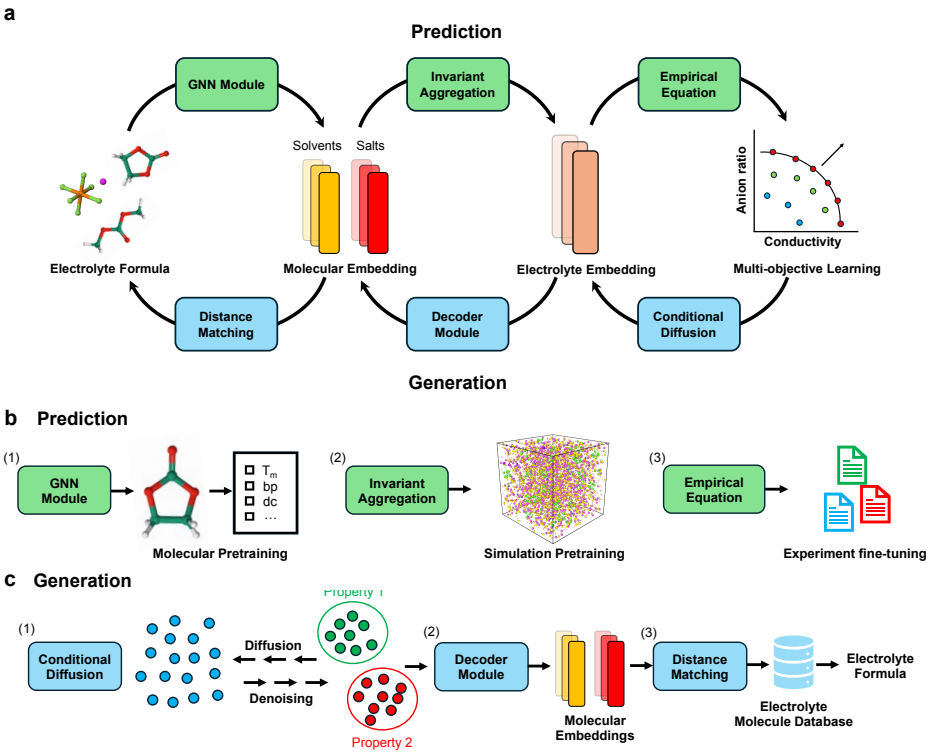

Fig. 1 A predictive and generative electrolyte design workflow reported in this work. **a** Forward (prediction) and inverse (generation) processes of electrolyte formulation are designed as three-stage workflows, using molecular embeddings (representations of individual component molecules in an electrolyte formulation) and electrolyte embeddings (permutation-invariant representations of entire electrolyte formulation). **b** Three stages of predictive model for conductivity and anion ratio predictions: (1) A GNN model is trained on single-molecule dataset on multi-property prediction, generating universal molecular embeddings. (2) MD data of around 100,000 different electrolyte formulations are utilized to further construct an informative electrolyte embedding from molecular embeddings. (3) An empirical relation is integrated into the model architecture and fine-tuned with 10,000+ experimental literature conductivity data points. **c** Three stages of generative model given property conditions: (1) A conditional diffusion model generates electrolyte embeddings based on specified properties. (2) The generated electrolyte embeddings are converted back to molecular embeddings with a decoder. (3) Finally, molecular embeddings are matched with our chemical database to obtain the electrolyte formulation.

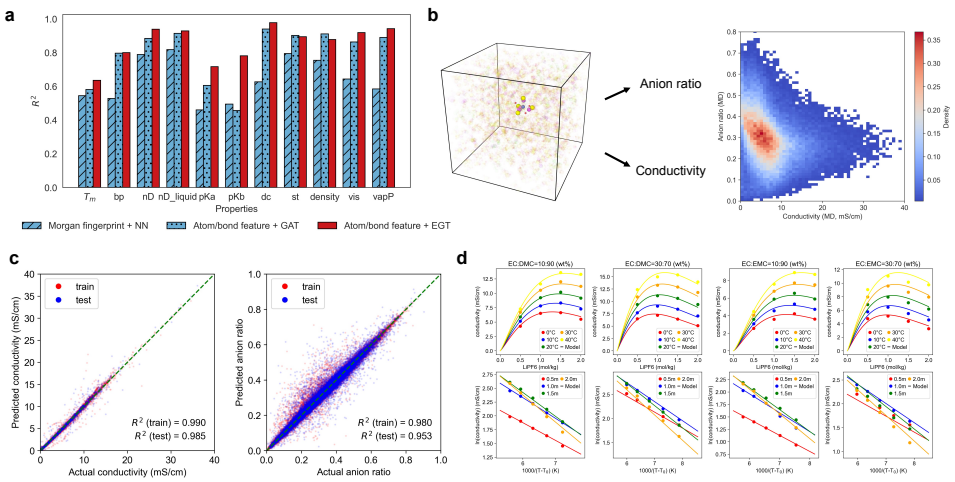

Fig. 2 Prediction performance a Comparison of model performance across various molecular properties during molecular pretraining. Our GNN model utilizes atom and bond features as input and is based on an EGT model (red bar, "Atom/bond feature + EGT"). Other two models are considered as baseline results (blue bars, "Morgan fingerprint + NN" and "Atom/bond feature + GAT"). **b** Anion ratio and ionic conductivity obtained from MD simulations using an OPLS force field. The density of data below 0.005 is omitted in the figure. Here, conductivity calculated using Mistry's method is used given that it aligns generally better with experiments (Fig. S5). **c** Predictions of our model versus ground truth after experimental fine-tuning for both anion ratio and conductivity. **d** Example predictions of temperature and concentration dependence of conductivity across various electrolyte systems with empirical equation. T_0 is a learnable temperature parameters in the empirical equation, which is generally related to glass transition temperature of electrolytes.

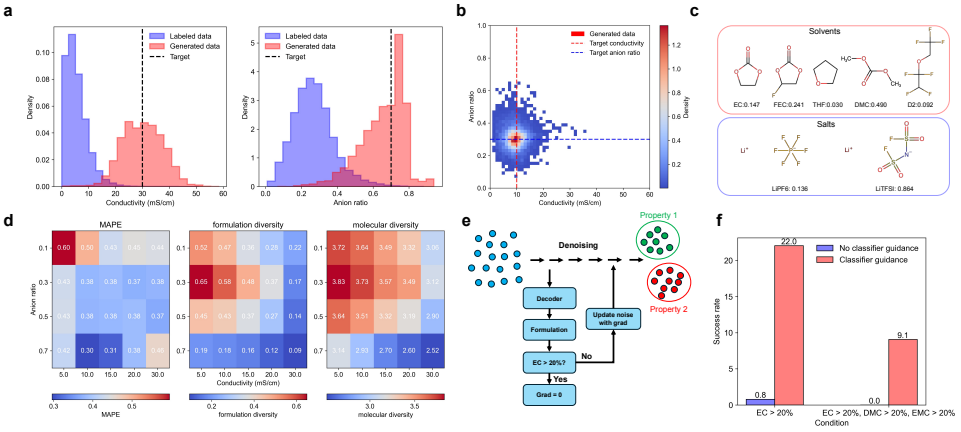

Fig. 3 Generation performances a Extrapolation of target properties using generative model. To extrapolate conductivity, the anion ratio is fixed at the dataset's mean value. Similarly, to extrapolate the anion ratio, conductivity is set to its mean. **b** Example distribution of generated electrolyte formulations given target conductivity (10.0 mS/cm) and anion ratio (0.3). More results of different conditions can be found in Fig. S8. **c** One example of generated electrolyte formulation from panel Fig. 3b. More examples are listed in Fig. S9 - Fig. S13. **d** Evaluation of generation using three metrics including MAPE and two diversity scores. **e** Schematic of classifier-guided conditional diffusion to ensure that the generated formulation satisfies the base formulation constraints. **f** Performance comparison between conditional generation and classifier-guided generation in satisfying the base formulation constraints.

Supporting Information

Supporting Information

Contents

A Data representation

The detailed data representations, along with their corresponding notations and shapes used in this work, are provided in Table S1.

A.1 Molecular representation

For molecular representation, we use a 2D graph to model each individual molecule, where (mapped) SMILES strings are utilized to extract atomic and bond features. The atomic features include: element type, ring connectivity, minimum ring size, and formal charge. The bond features are bond order and whether the bond belongs to a ring. Our GNN model takes these features as input and create a 1D vector for molecular embedding of dimension 64 (denoted as "$\boldsymbol{h}_m$" in the following context). For performance benchmarking, we also used a 512-bit Morgan fingerprint generated by RDKit [73] as the molecular descriptor to train a baseline model.

A.2 Electrolyte representation

The electrolyte formulation is represented by either an electrolyte embedding (denoted as "$\boldsymbol{h}_e$" in the following text) or a "Bag-of-Molecules" (BoM) vector (denoted as "$\boldsymbol{v}_{\text{BoM}}$" in the following text) when dealing with fixed chemical space in this work. The electrolyte embedding is a 384-dimensional vector derived from molecular embeddings using permutation-invariant aggregation operations. Notably, individual molecular contributions within the formulation can not be explicitly separated from the electrolyte embedding. The "BoM" vector follows a similar concept of "Bag-of-Words" from NLP. It has a dimensionality equal to the number of distinct molecules in our molecular "vocabulary", where each position corresponds to a specific molecule, and the value represents its molar ratio (in NLP, it is the occurrence frequency of each word). For instance, in our generation scenario, the vector is structured such that solvents occupy the first 62 dimensions, while the last 2 dimensions correspond to salts. In addition, molar ratios are handled separately for solvents and salts—specifically, the sum of the molar ratios for solvents is 1, and the same holds for salts.

B Molecular pretraining

B.1 Single-molecule dataset

The single-molecule dataset contains molecular property information gathered from 11 public datasets as shown in Table S3. The SMILES of molecules are collected from these databases or determined based on their names using CIRpy [74] which is a python interface for resolving chemical identifiers such as names, CAS registry numbers and SMILES strings. We then screened the data using a combination of following principles to remove incorrect, inconsistent or ambiguous entries:

1. Remove molecules with "bad" SMILES. A SMILES is a good SMILES if it satisfies:
   - "." does not appear in SMILES (individual molecule).
   - elements covered by H, C, N, O, F, P, S, Cl, Br, I.

- formal charge for F, Cl, Br, I must be negative.
- halogen connectivity must be 1.
- hybridization in s, sp, sp^2, sp^3.

2. Removing data from different sources that exhibit high inconsistencies.
   - large standard deviation for same properties.
   - different temperature dependence.
3. Removing ambiguous data such as "$T_m > 300K$".

After data preprocessing and screening, the final dataset consists of 241,414 entries with 11 molecular properties: $\boldsymbol{T_m}$: melting point, $\boldsymbol{T_b}$: boiling point, $\boldsymbol{n_D}/\boldsymbol{n_D^{\text{liquid}}}$: (liquid) refractive index, $\boldsymbol{pK_a}$: acid dissociation constant, $\boldsymbol{pK_b}$: base dissociation constant, $\boldsymbol{\varepsilon}$: dielectric constant, $\boldsymbol{\gamma_s}$: surface tension, $\boldsymbol{\rho}$: mass density, $\boldsymbol{\eta}$: viscosity and $\boldsymbol{P}_{\text{vap}}$: vapor pressure. Details regarding the final dataset, along with the corresponding data sources and the number of entries for each property label, are provided in Table S3.

B.2 Multi-task learning

Multi-task learning is employed to generate a universal molecular fingerprint for downstream formulation-level learning. The shared molecular fingerprint is then passed through separate readout blocks to predict different molecular properties. Of the 11 properties, density, refractive index, and dielectric constant are temperature-dependent but only contains data at a single temperature per molecule, whereas viscosity, surface tension, and vapor pressure are characterized over multiple temperatures. For the single-temperature properties, we simply concatenate the temperature with the molecular embedding for prediction. In contrast, to accurately capture the temperature dependence of viscosity, surface tension, and vapor pressure and to align our outputs with empirical curves, we incorporate empirical equation blocks for these properties and use MLPs to learn the corresponding parameters within the empirical equation. More specifically, the surface tension generally decreases linearly as temperature increases (Eötvös rule) [75], the relation between viscosity and temperature observes Vogel–Fulcher–Tammann (VFT) equation [39] and vapor pressure follows the Clausius-Clapeyron (CC) equation [76]. The corresponding empirical relations are simplified as follows:

$$A^{\gamma_s} = \text{MLP}(\boldsymbol{h}_m, \text{act=softplus}) \tag{B1}$$
$$B^{\gamma_s} = \text{MLP}(\boldsymbol{h}_m, \text{act=softplus}) \tag{B2}$$
$$\gamma_s = -A^{\gamma_s} \times T + B^{\gamma_s} \tag{B3}$$

$$\ln(\eta_0) = \text{MLP}(\boldsymbol{h}_m, \text{act=softplus}) \tag{B4}$$
$$A^{\eta} = \text{MLP}(\boldsymbol{h}_m, \text{act=softplus}) \tag{B5}$$
$$T_0^{\eta} = \text{MLP}(\boldsymbol{h}_m, \text{act=softplus}) \tag{B6}$$
$$\ln(\eta) = \ln(\eta_0) + \frac{A^{\eta}}{T - T_0^{\eta}} \tag{B7}$$

$$A^{P_{\text{vap}}} = \text{MLP}(\boldsymbol{h}_m, \text{act=softplus}) \tag{B8}$$

$$B^{P_{\text{vap}}} = \text{MLP}(\boldsymbol{h}_m, \text{act=softplus}) \tag{B9}$$

$$\ln(P_{\text{vap}}) = A^{P_{\text{vap}}} - \frac{B^{P_{\text{vap}}}}{T} \tag{B10}$$

where T is the temperature, As and Bs are learnable parameters (all positive) based on the molecular embedding ($\boldsymbol{h}_m$). The superscript denotes the relevant property, while the subscript indicates whether it is the reference state of that property. The "softplus" activation function is used for all the MLP blocks listed above to ensure positive learnable parameters and prevent gradient vanishing issues. For those properties without empirical relation, we use "sigmoid" function for their readout layers.

All the molecular properties are normalized to values between 0 and 1 based on their minimum and maximum values within the dataset. The total loss of multi-task learning is simply a weighted sum of Mean Squared Error (MSE) loss of each individual property. Here the weights are set to be 1 for all properties. The total loss is therefore as follow:

$$\begin{aligned}\mathcal{L}_{\text{tot}} = \mathcal{L}_{T_m} + \mathcal{L}_{T_b} + \mathcal{L}_{n_D} + \mathcal{L}_{n_D^{\text{liquid}}} + \\ \mathcal{L}_{pK_a} + \mathcal{L}_{pK_b} + \mathcal{L}_{\varepsilon} + \mathcal{L}_{\gamma_s} + \mathcal{L}_{\rho} + \mathcal{L}_{\eta} + \mathcal{L}_{P_{\text{vap}}}\end{aligned} \tag{B11}$$

The whole dataset is split into train (70%) and test (30%) sets for training and evaluation. As mentioned in the main text, we employ the graph blocks of the ByteFF model [35] to process atomic and bond information, generating a molecular embedding or fingerprint. The atom and edge features are first transformed into node and edge embeddings through an MLP block with two hidden layers of 32 neurons and GELU activation. These embeddings are then processed by three EGT layers, with a hidden dimension of 32 and 4 attention heads. The dimensions of both node and edge embeddings generated by the graph block are set to 32, resulting in a molecular embedding of dimension 64. The readout MLP contain three hidden layers with 256, 128 and 16 neurons respectively. MAE is used as the loss function for each property. When a label is unavailable for a specific molecule, a masking mechanism is applied to exclude the corresponding loss from the total loss function and the loss of each properties is averaged based on the number of labels. The model is trained with a batch size of 512, and early stopping with a patience of 50 epochs is applied to monitor and optimize training. A comparison between predicted properties and ground truth data is presented in Fig. S2.

B.3 Comparison with baseline models

To benchmark the performance of multi-task learning and ensure the effectiveness of the molecular embeddings, we developed and trained two additional baseline models: one is a simple fully connected neural network using the Morgan fingerprint as input features (labeled as "Morgan fingerprint + NN" in Table S4), and the other is a GNN model in which the EGT layers are replaced with GAT layers (labeled as "Atom/bond feature + GAT" in Table S4). We selected a 512-bit Morgan fingerprint as input and

passed it through an MLP block with two hidden layers (256 and 128 neurons) to match the dimensionality of the molecular fingerprint generated by the EGT model. In terms of GAT layers, we selected the same hyperparameters (hidden dimension = 32, number of attention heads = 4, attention channel = 8, number of layers = 3) as the EGT layers to ensure a fair comparison. The generated molecular embeddings from baseline models are passed through the same MLP and empirical equation blocks to predict molecular properties as described in Section B.2. All three models have approximately 800,000 trainable parameters and are trained using the same strategy for performance comparison.

The performances of these models, evaluated using R^2 and MAE, are displayed in Table S4. As the table shows, the EGT-based model outperforms the two baseline models in all property prediction tasks.

C Electrolyte property prediction

C.1 Computational dataset

To generate the computational dataset, MD simulations were conducted using LAMMPS [77], with the OPLS-AA force field [11, 78, 79]. The atomic charge was set as 0.8 times the CHELPG partial charges [80] calculated by the GPU4PySCF software [81]. The system size and simulation length of the MD simulations were set to be 10,000 atoms and 10 ns with the time step of 2 fs, respectively. To focus on the effect of coverage of chemical space instead of temperature, MD simulations are conducted at constant temperature 298 K (25 °C). Anion ratio was calculated by counting the number of molecules and ions within 2.8 Å of Li ions and dividing the number of anions by the total number of counted species.

In this work, two approaches were used to calculate ionic conductivities from MD simulations, the Nernst-Einstein method and a method described by Mistry et al. [50] that built on Stefan-Maxwell diffusivities. The Nernst-Einstein equation is described below:

$$\sigma_{NE} = \frac{e^2}{V k_B T}(N_+ z_+^2 \bar{D}_+ + N_- z_-^2 \bar{D}_-), \tag{C12}$$

where e is the elementary charge, k_B is the Boltzmann constant, V and T are volume and temperature of the simulated system, and $\bar{D}_\pm$, $z_\pm$, and $N_\pm$ are self-diffusion coefficients, valence charges, and number of the positive and negative ions, respectively. The NE relation neglects ion pairing and the correlated motion of molecules in MD simulations, causing it to overestimate ionic conductivity. In contrast, Mistry's method [50] incorporates the relative motion of neutral molecules, cations, and anions. Consequently, Mistry's conductivity estimates for typical liquid electrolytes are consistently lower than those from the NE relation, yet they converge to NE values at extremely low salt concentrations.

C.2 Experimental dataset

The experimental dataset of electrolyte conductivity is obtained by gathering data from public datasets and academic publications. There are two public datasets we

collect data from: (1) Polymer Electrolyte Dataset [9]; (2) CALiSol-23 Dataset [8]. For the Polymer Electrolyte Dataset, we first filtered out all polymer electrolytes based on their SMILES representation, as polymer SMILES contain the symbol "$*$". For the remaining data, we manually reviewed the source publications, examining each paper individually to remove inconsistent data (e.g. solid electrolyte, duplicate but inconsistent entries, ...) and correct any errors (e.g. unnormalized molar/weight ratio, incorrect SMILES, ...). We also removed the conductivity at extremely low temperature (below phase transition temperature) and concentration (salt molar ratio < 0.0001). For CALiSol-23 Dataset, a similar data processing approach was applied. In addition to these two public datasets, we incorporate conductivity data from 21 more recent studies, with a focus on interfacial stability, particularly in fluorinated electrolyte systems [49, 82–101].

During the data preprocessing, all ratios were converted to molar ratios. For data reported in weight ratio, it is straightforward to use molar weight for conversion. For data reported in volume ratios, we used density information if available from the paper, to convert them to molar ratios. For cases where density was not provided, we either searched for density values or utilized predicted densities from molecular pretraining for individual solvent molecules to estimate the density of solvent mixtures. We assume ideal mixing for estimating volumes of mixtures, which may introduce some errors in data processing. However, this approach is the best solution we can think of given the available information. As a result, the density of a solvent mixture is essentially the volume-weighted average of the densities of its individual components. In addition, the addition of solid salts can also lead to volume changes. To account for this, we adopt an empirical observation to estimate the molar ratio of both solvents and salts: preparing an xM electrolyte solution requires adding approximately $x(1+0.05x)$ moles of salt per liter of solvent.

As a result, we curated an experimental dataset of electrolyte conductivity consisting of 10,407 entries, including 62 types of solvents and 17 Li salts. This dataset covers a wide array of elements (H, Li, B, C, N, O, F, P, S, Cl) and chemical species (carbonates, ethers, nitriles, fluorinated solvents, phosphates, ...), which can provide comprehensive conductivity information for electrolyte design.

C.3 Pretraining with computational data

In the pretraining stage using computational data, the multi-task loss is described as:

$$\mathcal{L}_{\text{tot}} = \mathcal{L}_{\text{conductivity(NE)}} + \mathcal{L}_{\text{conductivity(mistry)}} + \mathcal{L}_{\text{anion ratio}} \tag{C13}$$

Both conductivities are log-normalized based on the maximum and minimum conductivity values of experimental dataset. Anion ratio is inherently between 0 and 1. The batch size is set to 512. To predict the anion ratio, temperature and concentration are concatenated to the end of the electrolyte embedding. Both temperature and concentration are each repeated eight times before concatenation. As a result, the final feature vector has a dimension of 400. The readout block for anion ratio prediction is a simple MLP consisting of three hidden layers, followed by a sigmoid activation function. The hidden layers contain 512, 128, and 16 neurons, respectively. In addition, the

empirical relation block is also used for predictions of conductivities calculated from MD results.

C.4 Fine-tuning with experimental data

Experimental fine-tuning follows a similar multi-task learning approach as molecular and computational pretraining. The dataset comprises over 10,000 experimental conductivity measurements and more than 100,000 computational anion ratio samples, which are randomly split into training and test sets. Fine-tuning was initialized from the checkpoints obtained during computational pretraining, using conductivity predicted by Mistry's method as the starting point for experimental conductivity prediction. We freeze the molecular embedding and allow updates of the electrolyte embedding during the fine-tuning. The MAE losses for conductivity and anion ratio are averaged within each batch based on the number of samples with corresponding labels. The total loss is the sum of MAE losses of conductivity and anion ratios:

$$\mathcal{L}_{\text{tot}} = \mathcal{L}_{\text{conductivity}} + \mathcal{L}_{\text{anion ratio}} \tag{C14}$$

To perform a generalization test of the model as shown in Table S2, the training and test sets are split based on whether the electrolyte formulations contain solvents with specific elements such as F, S, or P for experimental fine-tuning. We aim to evaluate whether computational pretraining can enhance the model's ability to generalize to molecules not observed in experimental data. Therefore, the corresponding solvents are kept in computational pretraining. For generalization test on viscosity, we modified the empirical equation (with or without η in the Eq. D26) during both computational pretraining and experimental fine-tuning to keep the comparison consistent.

D Physics-informed architectures

D.1 Permutation invariance

Permutation invariance is realized using a self-attention mechanism with molar ratio-based scaling as described in Section 4. The multi-head attention uses classical query (Q), key (K) and value (V) to compute the attention output (see Fig. S1c):

$$Q = W_Q(\boldsymbol{r} \odot \boldsymbol{X}) + B_Q \tag{D15}$$
$$K = W_K(\boldsymbol{r} \odot \boldsymbol{X}) + B_K \tag{D16}$$
$$V = W_V(\boldsymbol{r} \odot \boldsymbol{X}) + B_V \tag{D17}$$
$$\boldsymbol{X'} = \text{softmax}(\frac{QK^T}{\sqrt{d_K}})V \tag{D18}$$

where W and B are learnable weights and biases in the self-attention mechanism. "$\odot$" stands for row-wise multiplication and $\boldsymbol{X}$ is the molecular embedding tensor and d_k is the dimensionality of K. To maintain a consistent shape for $\boldsymbol{X}$ in the attention mechanism, we pad the molecular embeddings of both solvents and salts to a maximum

of 12 molecules (12 molecules for solvents and 12 molecules for salts). The attention output is then aggregated using an inner product with molar ratio vector ($\boldsymbol{r}$) to ensure permutation invariance:

$$\boldsymbol{h_m} = \boldsymbol{r^T} \cdot \boldsymbol{X'} \tag{D19}$$

"·" is the inner product. We separate the molecular embeddings of solvents and salts, processing them independently through the aggregation block. Since salts are typically fewer in number than solvents in the electrolytes, we use 2 attention heads for salts and 4 for solvents. All the hidden dimensions are selected as 64 for each attention head. The numbers of attention head are 4 and 2 for solvents and salts respectively. As a result, the electrolyte embedding has a dimension of $64 \times (4 + 2) = 384$.

D.2 Empirical relations

The empirical equation is integrated into our predictive model to enable physics-informed conductivity prediction. We here discuss about the derivation of empirical relation.

In dilute solutions, the conductivity follows the equation proposed by Every et al. [102]:

$$\sigma(T, c) = \sum n_i q_i \mu_i \tag{D20}$$

where σ is ionic conductivity, n_i is the number of ions, q_i is the charge of the ion species i and μ_i is the mobility. In relatively medium and high salt concentration, the equation can be extended by defining an effective number of free ions. Based on Zhang et al. [41]'s work, the number of free ions n_i is a function of the electrolyte concentration c. As the concentration increases, there is an increasing number of ion associations. Consequently, the number of free ions will not increase linearly as the concentration arises. Their work assumed the following relation between the number of free ions and salt concentration [41] and we use it in this work:

$$n_i = Ac^{n_1} \tag{D21}$$

In terms of mobility μ, it is empirically described by an exponential relationship that depends on both temperature and concentration. For the temperature dependence, the VFT equation is widely used [39] and has been validated by extensive experimental studies [70, 103, 104]. As a result, the mobility observes the following equation:

$$\mu = \mu_0 e^{-\frac{E_a}{T - T_0}} \tag{D22}$$

where μ_0 is the mobility when temperature is infinite. T_0 is physically correlated to the glass transition temperature of the electrolytes and E_a is the activation energy for electrical conduction. In Fu et al. [40]'s work, the activation energy correlates to the

concentration with the below equation based on quasi-lattice theory:

$$E_a = (ac^{-0.5} + b)c + d \tag{D23}$$

where a, b and d are system-dependent parameters. The exponent -0.5 is chosen based on experimental observations of the [BMIM][TFSI]-PC/GBL mixture: as concentration increases, the activation energy increases, but the slope of the curve decreases. However, we found that in other systems like $LiClO_4$ in γ-butyrolactone [105], the slope in activation energy-concentration plot can also increases. To accommodate a more flexible dependence of E_a on concentration, we adopt the following modified expression:

$$E_a = Bc^{n_2} + D \tag{D24}$$

where B and D are adjustable parameters, and n_2 is constrained to be greater than zero only.

In terms of viscosity, we incorporate it following Walden's rule to provide prior intuition for predicting conductivity as the conductivity generally decreases as viscosity increases. If an electrolyte molecule is unseen by the model, its viscosity can serve as a prior to infer its potential impact on the overall electrolyte conductivity. To calculate the viscosity of mixed solvents in the electrolyte, we assume the mixture behaves as an ideal solution. Although this assumption may introduce inaccuracies, the empirical equation incorporates learnable parameters (A) that can correct any resulting deviations. As a result, incorporating such a term in the equation should not negatively impact the prediction. The viscosity of the mixture is then determined based on the viscosities of individual solvent molecules, which are predicted during molecular pretraining. The viscosity is described with Arrhenius blending rule [106, 107]:

$$ln(\eta) = \sum r_i ln(\eta_i) \tag{D25}$$

where r_i is the molar ratio of each solvent molecule and η_i is the predicted viscosity of each single molecule. By plugging in all components, the final ionic conductivity is defined with the following equation:

$$\sigma(T, c) = \frac{A}{\eta} c^{n_1} e^{-\frac{B \times c^{n_2} + D}{T - T_0}} \tag{D26}$$

where σ is the conductivity, c is salt concentration, T is the temperature, η is the viscosity, A, B, D, n_1, n_2 and T_0 are all learnable parameters, dependent on the electrolyte embedding ($\boldsymbol{h}_e$). During the training of the model, we actually use the logarithmic form of Eq. D26 for better training stability given that exponential term can easily explode or vanish:

$$\ln(\sigma(T, c)) = \ln(A) - \ln(\eta) + n_1 \ln(c) - \frac{B \times c^{n_2} + D}{T - T_0} \tag{D27}$$

$$\ln(A) = \text{MLP}(\boldsymbol{h}_e, \text{act=softplus}) \tag{D28}$$

$$B = \text{MLP}(\boldsymbol{h}_e, \text{act=softplus}) \tag{D29}$$
$$D = \text{MLP}(\boldsymbol{h}_e, \text{act=softplus}) \tag{D30}$$
$$n_1 = \text{MLP}(\boldsymbol{h}_e, \text{act=softplus}) \tag{D31}$$
$$n_2 = \text{MLP}(\boldsymbol{h}_e, \text{act=softplus}) \tag{D32}$$
$$T_0 = \text{MLP}(\boldsymbol{h}_e, \text{act=sigmoid}) \tag{D33}$$

where the activation functions of last layer of each MLP readout block are listed above, which constrains the range of learnable parameters within the empirical relation. The reason we use "sigmoid" function for T_0 is that we only consider cases when the working temperature T is higher than T_0 or glass transition temperature in this work.

The empirical equation is further validated using AEM [24] and experimental data [70]. Given six fitting parameters (except viscosity) in the empirical relation, we select only electrolyte systems that contain more than 30 data points spanning different temperatures and concentrations for validation of our propose empirical relation. To demonstrate, we select AEM data from four different binary mixtures and test experimental data for EC/EMC/$LiPF_6$ electrolyte mixture with varying component ratios (Fig. S4). For curve fitting, we utilized the "curve_fit" function in SciPy with a least-squares method [108]. All the learnable parameters are constrained to be greater than 0 and the maximum number of function evaluations during curve fitting is set to 10000.

An ablation study was further conducted to illustrate the necessity of parameters in our empirical equation. Given the mathematical form of the empirical equation, we removed n_1, n_2, and D respectively to evaluate their impact on the fitting accuracy. As Table S5 reveals, the error (MSE) increases considerably when any of these parameters are removed. These results demonstrate that each parameter is critical for accurately capturing the empirical relationship. While several empirical relations can describe how concentration and temperature influence conductivity, our equation performs well, achieving accurate predictions across more than 10,000 experimental measurements.

E Generative electrolyte design

E.1 Synthetic dataset

We use a synthetic dataset generated by our predictive model to train conditional diffusion model. The synthetic dataset provides more balanced coverage of the electrolyte design space, with a particular emphasis on multi-component systems that are really used in commercial battery systems. The dataset contains electrolyte formulations with various numbers of solvents listed as follows: (1) 1 solvent + 2 salts (62 data including all possible solvents); (2) 2 solvents + 2 salts (1891 data including all possible binary combinations of solvents); (3) 3 solvents + 2 salts (2805 data which is 10% of the rest of data); (4) 4 solvents + 2 salts (5609 data which is 20% of the rest of data); (5) 5 solvents + 2 salts (8414 data which is 30% of the rest of data); (6) 6 solvents + 2 salts (11219 data which is 40% of the rest of data). We selected six solvents and two salts as the maximum to maintain consistency with the computational dataset, and eight components are generally sufficient to formulate a commercially

viable electrolyte system. The molar ratios for solvents and salts are randomly sampled between 0 and 1 and normalized within solvents and salts, respectively. The overall salt concentration is fixed at 0.1 in terms of molar ratio, which is typically near the ideal concentration corresponding to peak conductivity, and the temperature is set to 25 °C as the room temperature. We include all possible unary and binary combinations of solvent molecules in the dataset and randomly sample multi-component systems based on their combinatorial count to ensure comprehensive coverage of the formulation space.

E.2 Conditional diffusion model

We utilize a diffusion model to generate electrolyte formulations given specific target properties. Electrolyte embeddings ($\boldsymbol{h}_e$) serve as the target output, reducing generation costs given their low dimensionality while ensuring permutation invariance. More specifically, the conditional generation is achieved based on the DDPM model [71]. In a DDPM, the forward process (diffusion process) defines a Markov chain that gradually adds Gaussian noise to the input data according to a variance schedule β_1, β_2, ..., β_T:

$$q(\boldsymbol{x}_{1:T}|x_0) = \prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \tag{E34}$$

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \boldsymbol{I}) \tag{E35}$$

where $\boldsymbol{x}_t$ is the data at time t, $\boldsymbol{I}$ is identity matrix/tensor. As time t increases, the data become more and more noisy. The key objective is to reverse this diffusion process, enabling the generation of data from random noise. The reverse process is defined as:

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t), \boldsymbol{\Sigma}_\theta(\boldsymbol{x}_t, t)) \tag{E36}$$

$$\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\right) \tag{E37}$$

$$\Sigma_\theta(\boldsymbol{x}_t, t) = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \boldsymbol{I} \tag{E38}$$

where $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, $\boldsymbol{\epsilon}_\theta$ is a functional approximator to predict added noise $\boldsymbol{\epsilon}$ during the diffusion process from $\boldsymbol{x}_t$. The functional approximator is a 1D U-Net model [109] we adopted from an open-source Github repository [110]. The 1D U-Net contains a downsampling encoder, a middle bottleneck block and an upsampling decoder with self-attention mechanism. We used a base hidden dimension of 64, and for each upsampling or downsampling layer, the hidden dimension is doubled or halved relative to the previous layer. The number of attention head is set to 4 and the dimension is 16 for self-attention blocks in the U-Net.

To achieve conditional generation, the input conditions are first processed through an MLP block to produce a high-dimensional conditional embedding vector. This embedding is then treated similarly as the time embedding in a diffusion model, where it modulates the input via Feature-wise Linear Modulation (FiLM) [111]. In the process, the model dynamically scales and shifts feature activations based on the time

and conditioning variables, enabling effective control over the generated output:

$$\boldsymbol{x} = (1 + \gamma(v, t))\boldsymbol{x} + \beta(v, t) \tag{E39}$$

$$\gamma(v, t), \beta(v, t) = \text{MLP}(v) + \text{MLP}(t) \tag{E40}$$

where v is the conditional variable, including conductivity and anion ratio in this work. Both the time and conditional embedding dimensions are set to 256.

In terms of other hyperparameters, the number of diffusion step is 1000. A cosine scheduler is implements for β_t. Training is performed with a batch size of 256, using a 70/30 train-test split to minimize the MSE loss of predicted noises. Early stopping with a patience of 20 epochs is applied to optimize the training process.

E.3 Electrolyte decoder

A decoder is developed to convert generated electrolyte embeddings back into their original electrolyte formulations. We first transfer the electrolyte embedding to a set of molecular embeddings of solvents and salts, and then match each molecular embedding to our chemical vocabulary to identify the corresponding molecules. The molar ratios of each molecule are also predicted from the electrolyte embedding. Based on the matching results and predicted molar ratios, we derive the "BoM" vector for each electrolyte formulation. There are two main reasons for retaining molecular embeddings during the decoding process. First, this approach can be extended for novel electrolyte molecule discovery by incorporating an additional molecular decoder capable of converting molecular embeddings back into molecular SMILES. Second, molecular embeddings possess valuable chemical information, enhancing the overall expressivity of the workflow.

To generate high-dimensional molecular embeddings for solvents and salts from the low-dimensional electrolyte embedding, we utilize a learnable seed vectors $\boldsymbol{S} \in \mathbb{R}^{N \times d}$ (N is the number of solvents and salts, d is the dimension of molecular embedding) [72] to extract molecular embedding at each position given the electrolyte embedding as the context. The steps are as follows:

$$\boldsymbol{c} = \text{MLP}(\boldsymbol{h}_e) \in \mathbb{R}^{1 \times d} \tag{E41}$$

$$\boldsymbol{C} = [\boldsymbol{c}, \boldsymbol{c}, \boldsymbol{c}, ...] \in \mathbb{R}^{N \times d} \tag{E42}$$

$$\boldsymbol{S}^{(0)} = \boldsymbol{S} \tag{E43}$$

$$Q^{(L)} = W_Q^{(L)} \boldsymbol{S}^{(L)} + B_Q^{(L)} \tag{E44}$$

$$K^{(L)} = W_K^{(L)} \boldsymbol{C} + B_K^{(L)} \tag{E45}$$

$$V^{(L)} = W_V^{(L)} \boldsymbol{C} + B_V^{(L)} \tag{E46}$$

$$\boldsymbol{S}^{(L+1)} = \text{softmax}(\frac{Q^{(L)}(K^{(L)})^T}{\sqrt{d_{K^{(L)}}}})V^{(L)} \in \mathbb{R}^{N \times d} \tag{E47}$$

$$\boldsymbol{S}^{(F)} = \boldsymbol{h}_m \tag{E48}$$

$$\boldsymbol{r} = \text{softmax}(\text{MLP}(\text{concat}(\boldsymbol{h}_m, \boldsymbol{C}))) \tag{E49}$$

where $\boldsymbol{C}$ is the context embedding based on the electrolyte embedding, $\boldsymbol{S}^{(L)}$ denotes the output at layer L, $\boldsymbol{S}^{(0)}$ is the input learnable query and $\boldsymbol{S}^{(F)}$ is the output of last layer which corresponds to the molecular embeddings. The solvents and salts are processed separately using the above procedure.

With the molecular embeddings, we further match each embedding to molecules within our chemical vocabulary. This matching is performed by computing the L2 distance between embedding vectors. Since a hard indexing operation would disrupt the computational graph and hinder back propagation, we employ a soft matching mechanism instead:

$$\boldsymbol{p} : p_{i,j} = \text{softmax}(-\lambda d_{i,j}) \tag{E50}$$

where $p_{i,j}$ is the probability of "Mol i" is same/similar as "Mol j", $d_{i,j}$ is the L2 distance between embedding vectors of these two molecules, λ is a parameter to control the sharpness of the matching mechanism. In other words, a larger λ makes the matching process more selective, increasing the likelihood of collapsing onto a single molecule. In this work, we set λ to 1000 to make the matching process closer to a hard indexing operation while avoiding numerical instability that may arise when λ is too large.

The final BoM vector is then obtained based on the inner product of matching probability and predicted molar ratios:

$$\boldsymbol{v}_{\text{BoM}} = \boldsymbol{p}^T \cdot \boldsymbol{r} \tag{E51}$$

The training of the decoder is performed by minimizing the MSE loss of predicted and actual $\boldsymbol{v}_{\text{BoM}}$.

E.4 Evaluation metrics

The three evaluation metrics we utilized to examine the performances of generation are MAPE, formulation diversity and molecular diversity. The definitions of these metrics are listed as follows:

$$\text{MAPE} = \sum (|\frac{y_c^{\text{pred}} - y_c^{\text{target}}}{y_c^{\text{target}}}| + |\frac{y_a^{\text{pred}} - y_a^{\text{target}}}{y_a^{\text{target}}}|) \tag{E52}$$

where y_c^{pred} and y_c^{target} are predicted and target conductivities respectively, y_a^{pred} and y_a^{target} are predicted and target anion ratios.

The formulation diversity is defined based on the average minimal pairwise L1 distance between $\boldsymbol{v}_{\text{BoM}}$ of generated formulations, showing the difference between generated formulations. In other words, it is the average distance from each generated formulation to its nearest neighbor:

$$\text{formulation diversity} = \frac{1}{N} \sum_{i=1}^{N} \min_{\substack{j=1 \\ j \neq i}}^{N} \sum_{k=1}^{d} |\boldsymbol{v}_{BoM,i}^{(k)} - \boldsymbol{v}_{BoM,j}^{(k)}| \tag{E53}$$

where d is the dimension of $\boldsymbol{v}_{\text{BoM}}$. N is the number of generated samples. We use the L1 distance to evaluate differences between electrolyte formulations, as it not only captures variations in the molar ratios of shared molecular species but also imposes a large penalty when there is no overlap in molecular components. For example, if the molar ratio of EC increases from 0.2 to 0.4 in a formulation, the L1 distance between the original and modified formulations is 0.2. In contrast, if 20% EC is replaced with 20% DMC, the L1 distance increases to 0.4. In addition, the L1 distance is inherently normalized, since the molar ratios sum to one.

The molecular diversity is defined based on the Shannon entropy/relative entropy of occurrence frequency of different component molecules calculated using Scipy [108]:

$$\text{molecular diversity} = -\sum_{i=1}^{N} p_i \log(p_i) \tag{E54}$$

where p_i is the occurrence frequency of component molecule i. The frequency is normalized during the calculation. This metric captures the diversity of electrolyte formulations considering the molecular species. For instance, in high-conductivity electrolyte systems, acetonitrile (AN) is widely seen which can be reflected by this score.

E.5 Classifier-guided diffusion

To generate electrolyte formulations that satisfy base formulation constraints, we employ a CGD-based method. The CGD method explicitly calculates gradients during the sampling process (reverse process) given predicted label y from the classifier with respect to the noisy data. Instead of sampling from the Gaussian distribution at each time step t as Eq. E36 shows, it shifts the mean value $\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t)$ towards the target class using the gradients:

$$\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, y, t) = \boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t) + s \sum \nabla_{\boldsymbol{x}_t} \log p_\phi(y|\boldsymbol{x}_t) \tag{E55}$$

where s is a gradient scale that adjusts the strength of classifier-guided generation. ϕ is the classifier function. A larger s results in more sharp guidance towards the target constraint.

To accommodate different base formulation scenarios, we formulate the classifier function as follows:

$$\boldsymbol{v}_{\text{BoM}} = \text{Decoder}(\boldsymbol{x}_t) \tag{E56}$$

$$\phi(\boldsymbol{x}_t, \boldsymbol{b}, \boldsymbol{\tau}, \boldsymbol{m}) = \sum_{i=1}^{n_m} -\text{ReLU}((\boldsymbol{b}[i] - \boldsymbol{v}_{\text{BoM}}[\boldsymbol{m}_i]) \times \boldsymbol{\tau}_i) \tag{E57}$$

where $\boldsymbol{m}$ is a list of molecule indices which require control on molar ratios, $\boldsymbol{b}$ is the corresponding bound values of molar ratio, $\boldsymbol{\tau}$ indicates ("-" or "+") whether the $\boldsymbol{\tau}$ stores the upper or lower bounds. n_m is the number of molecules that require controls

within the base formulation. Below is an example of representing a base formulation with $\boldsymbol{m}$, $\boldsymbol{b}$ and $\boldsymbol{\tau}$:

$$\boldsymbol{m} = [0, 2], \boldsymbol{b} = [0.2, 0.4], \boldsymbol{\tau} = [1, -1] \Leftrightarrow \text{“Mol 0”} > 20\% \text{ and “Mol 2”} < 40\% \quad \text{(E58)}$$

More specifically, in the above example, $\boldsymbol{m}$ stores the indices of two molecules in the base formulation, namely “Mol 0” and “Mol 2”, which are referenced based on our molecular dictionary. The vector $\boldsymbol{b}$ provides the molar ratio thresholds for these molecules, with 0.2 for “Mol 0” and 0.4 for “Mol 2”. Finally, in $\boldsymbol{\tau}$, a value of “1” indicates that the molar ratio of the corresponding molecule should be greater than its threshold (lower bound), while “-1” signifies that the molar ratio should be less than its threshold (upper bound).

This classifier function allows us to handle various base formulation requirements while also enabling precise control over the molar ratios of component molecules. Specifically, we can achieve this by setting both upper and lower bounds for a given molecule. Since small variations in molar ratios typically have minimal impact on key electrolyte properties, such as conductivity, we can apply a relatively soft constraint.

Finally, we utilize a negative ReLU function to ensure proper gradient behavior during sampling. Above all, the gradient should be zero when the condition is satisfied. In other words, one side of the activation function should be constant as in ReLU function. Second, when the condition is not met, the gradient should be positive; thus, a negative version of ReLU is used. Third, since the ReLU function maintains a constant gradient when the condition is unsatisfied, we do not need to adjust the scale of molar ratios to avoid gradient vanishing or exploding issues. The gradient strength can be easily adjusted via the gradient scale, offering flexibility in guidance control. The gradient scales $\boldsymbol{s}$ are chosen between 0, 1, 10, 100, 1000 and 10000. When gradient scale is 0, the classifier does not impose any constraint to the generation which falls back to conditional generation with only conductivity and anion ratio targets. We found that for both base formulation constraints we tested in this study — “EC $>$ 20%” and “EC $>$ 20%, DMC $>$ 20% and EMC $>$ 20%”, the success rates are highest when $\boldsymbol{s} = 1000$.

For the decoder used in the CGD method, we retrain it with noisy electrolyte embeddings to predict the corresponding BoM vectors. The noisy data are sampled directly from the forward process in the diffusion model during training. As expected, the decoder trained on noisy data exhibits a higher loss compared to the model trained on the final electrolyte embedding (Section E.3). However, this training strategy enhances the model's ability to establish a more robust linkage between noisy data during sampling and the final electrolyte formulation which is represented by $\boldsymbol{v}_{\text{BoM}}$.

Notation	Data representation	Shape
$\boldsymbol{X}$	padded molecular embeddings of solvents and salts within the electrolyte.	(24, 64)
$\boldsymbol{x}/\boldsymbol{x}_t$	noisy data of electrolyte embedding during diffusion and denoising process.	(384,)
$\boldsymbol{h}_m$	molecular embedding of one single molecule.	(64,)
$\boldsymbol{h}_e$	electrolyte embedding of one formulation.	(384,)
$\boldsymbol{v}_{\mathrm{BoM}}$	"Bag-of-Molecules" vector to represent one electrolyte formulation.	(vocab_size,)

Table S1 Data representation notation. The table presents the key data representations for both molecules and electrolyte formulations, along with their corresponding notations and common dimensions used in this work. "vocab_size" refers to the number of available solvent and salt molecules within our chemical database.

Data split	no F/with F		no S/with S		no P/with P		high viscosity/low viscosity	
Method	Computational pretraining	No pretraining	Computational pretraining	No pretraining	Computational pretraining	No pretraining	With viscosity	No viscosity
R^2	**0.901**	0.893	**0.797**	0.491	**0.611**	0.238	**0.559**	0.425

Table S2 Generalization tests. We evaluated how computational pretraining and the inclusion of viscosity in the empirical relation influence the generalization ability of our model. The label "no F / with F" indicate that the training set contains electrolytes with non-fluorinated solvents, while the test set contains electrolytes with fluorinated solvents. Similar notations are used for "no S/with S" and "no P/with P". The label "high viscosity / low viscosity" corresponds to a scenario in which low-viscosity solvents including ethyl acetate (EA), methyl acetate (MA), and AN are excluded from the training set and included only in the test set. The R^2 score is computed after performing a linear fit between the predicted and actual conductivity values.

Property	T_m	T_b	n_D	n_D^{liquid}	pK_a	pK_b
Resource	Tetko et al. [112] Bradley et al. [113] CAS Covid dataset [114] CRC handbook [115] OPERA dataset [116]	CRC handbook [115] OPERA dataset [116]	CRC handbook [115]	CRC handbook [115]	OPERA dataset [117]	OPERA dataset [117]
Property	ε	γ_s	ρ	η	P_{vap}	
Resource	Bouteloup and Mathieu [118]	Wohlfarth and Wohlfarth [119]	CRC handbook [115]	Goussard et al. [120] Chew et al. [121]	OPERA dataset [116] Gharagheizi et al. [122]	

Table S3 Single-molecule dataset. The table lists the data resources of different molecular properties.

T_m: melting point; T_b: boiling point; $n_D/n_D^{\mathrm{liquid}}$: (liquid) refractive index; ε: dielectric constant; γ_s: surface tension, η: viscosity; P_{vap}: vapor pressure.

Total number of data: T_m: 184921; T_b: 7464; n_D: 4252; n_D^{liquid}: 2592; pK_a: 2568; pK_b: 3283, ε: 1286; γ_s: 14317; ρ: 5181; η: 3405; P_{vap}: 32184.

Model	T_m	T_b	n_D	n_D^{liquid}	pK_a	pK_b
	R^2/MAE	R^2/MAE	R^2/MAE	R^2/MAE	R^2/MAE	R^2/MAE
Morgan fingerprint + NN	0.506/0.045	0.625/0.035	0.768/0.026	0.789/0.029	0.537/0.055	0.541/0.036
Atom/bond feature + GAT	0.629/0.039	0.799/0.020	0.825/0.014	0.892/0.014	0.651/0.044	0.750/0.027
Atom/bond feature + EGT	**0.649/0.038**	**0.888/0.015**	**0.878/0.012**	**0.932/0.013**	**0.819/0.033**	**0.852/0.023**

Model	ε	γ_s	ρ	η	P_{vap}
	R^2/MAE	R^2/MAE	R^2/MAE	R^2/MAE	R^2/MAE
Morgan fingerprint + NN	0.744/0.062	0.600/0.048	0.747/0.027	0.530/0.098	0.558/0.077
Atom/bond feature + GAT	0.982/0.019	0.900/0.019	0.842/0.014	0.932/0.030	0.927/0.022
Atom/bond feature + EGT	**0.991/0.014**	**0.958/0.013**	**0.880/0.013**	**0.949/0.027**	**0.967/0.015**

Table S4 Model performance comparison for molecular property predictions. The best performance of each property among the three models is highlighted in bold. All properties are normalized based on the minimum and maximum values within the dataset. For "ϵ", "η" and "P_{vap}", the properties are log-normalized due to their long-tail distribution or exponential dependence on temperature.

Empirical form Parameters	$Ac^{n_1}e^{-\frac{B\times c^{n_2}+D}{T-T_0}}$ (A, B, D, n_1, n_2, T_0)	$Ace^{-\frac{B\times c^{n_2}+D}{T-T_0}}$ (A, B, D, n_2, T_0)	$Ac^{n_1}e^{-\frac{B\times c+D}{T-T_0}}$ (A, B, D, n_1, T_0)	$Ac^{n_1}e^{-\frac{B\times c^{n_2}}{T-T_0}}$ (A, B, n_1, n_2, T_0)
AEM data [24]	**0.00504**	0.01283	0.01237	0.02458
Experimental data [70]	**0.01326**	0.01663	0.02108	0.09039

Table S5 Ablation study of empirical relations. We compare our empirical equations with modified versions where one parameter is removed using both AEM and experimental data. The values represent the mean MSEs ((mS/cm)^2) across different electrolyte formulations of each empirical relation after parameter fitting. Our selected equation exhibits better fitting accuracy compared to these modified versions.

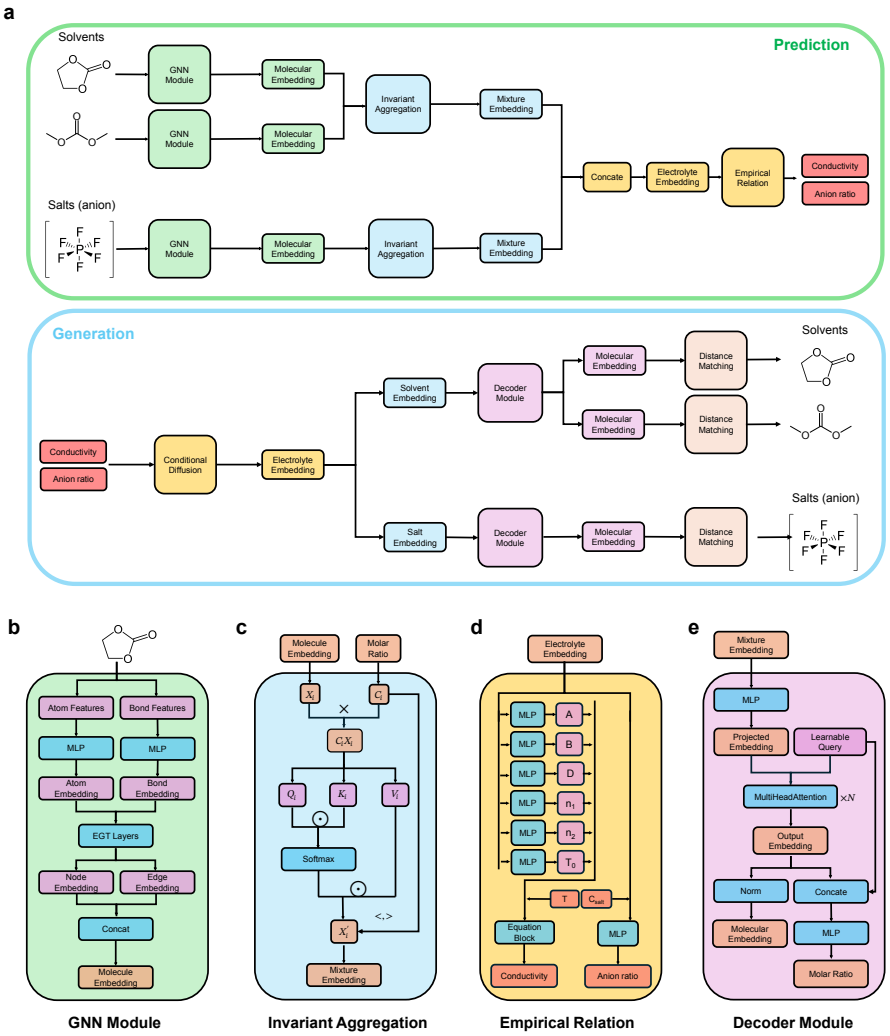

Fig. S1 Detailed model architectures within the workflow reported in this work. a Overall workflow of both predictive and generative process. Solvents and Li salts are considered separately in the workflow. **b** Module architecture of GNN model which takes SMILES of a single molecule as input and generates a universal molecular embedding by multi-task learning on 11 molecular properties. **c**. A self-attention-based aggregation block to merge multiple molecular embeddings into a mixture embedding and ensure permutation invariance. "$\times$" is row-wise multiplication, "$\odot$" stands for matrix multiplication and "$\langle , \rangle$" represents inner product. **d** The empirical relation block for conductivity and anion ratio prediction. For conductivity, there are six empirical learnable parameters (except viscosity) based on the electrolyte embedding, while anion ratio is predicted directly from the electrolyte embedding concatenated with temperature and concentration using a readout MLP layer. Activation layers are omitted from the schematic. **e** The decoder module which recovers molecular embeddings from mixture embeddings for both solvents and salts, which is further matched to molecules in our electrolyte molecule database.

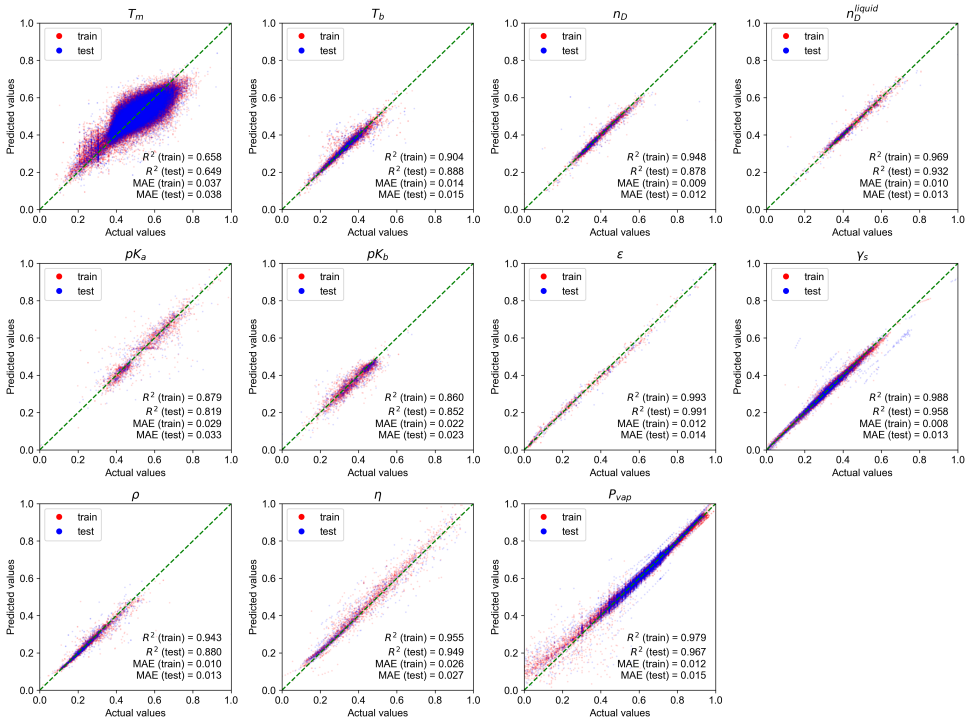

Fig. S2 Predicted molecular properties from molecular pretraining. The parity plots comparing molecular properties predicted from our GNN model and ground truth from experiments.

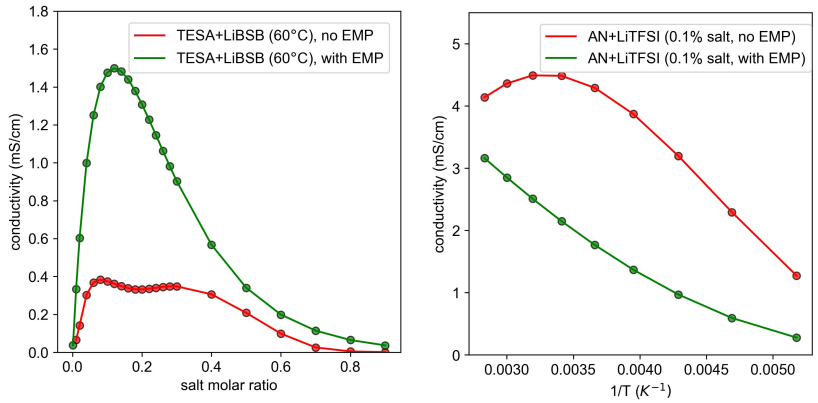

Fig. S3 Unphysical predictions without empirical relation. The figures show example outliers predicted by model without empirical relation in comparison with our model with empirical equations. "EMP" stands for "empirical relation". The left figure demonstrates a outlier case where conductivities show multiple peaks as concentration varies, while the right figure depicts an incorrect temperature dependence of conductivity as conductivity should increase monotonically with temperature.

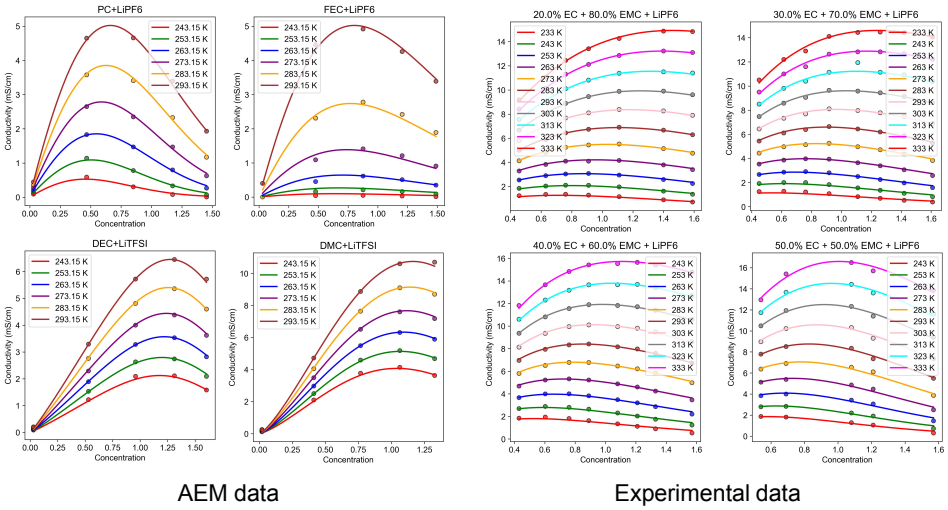

Fig. S4 Empirical equation validation. AEM and experimental data of various electrolyte formulation systems are utilized to validate our chosen empirical relation.

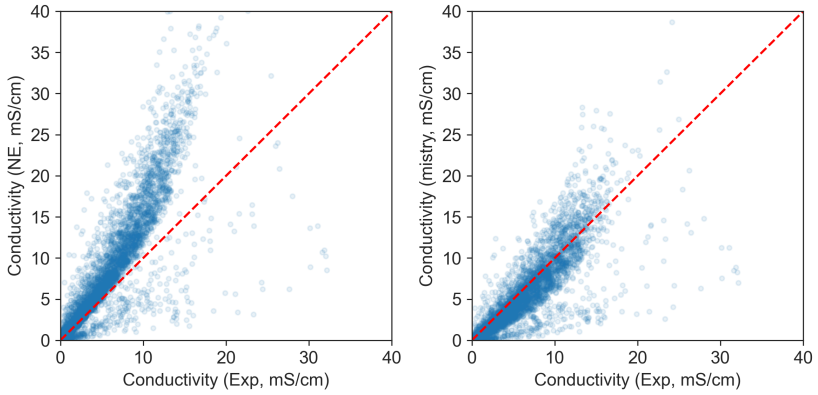

Fig. S5 Comparison of MD conductivities with experimental measurements. The left parity plot compares NE conductivity with experimental ground truth and the right parity plot compares mistry conductivity.

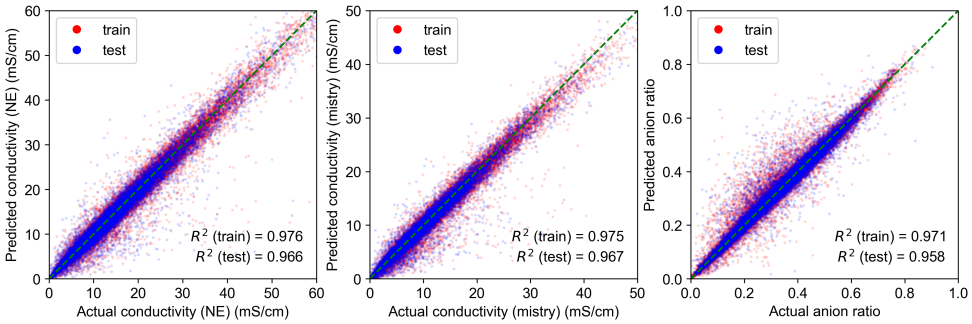

Fig. S6 Predicted electrolyte properties from computational pretraining. The parity plots comparing conductivities and anion ratio predicted from our model and ground truth from MD simulations.

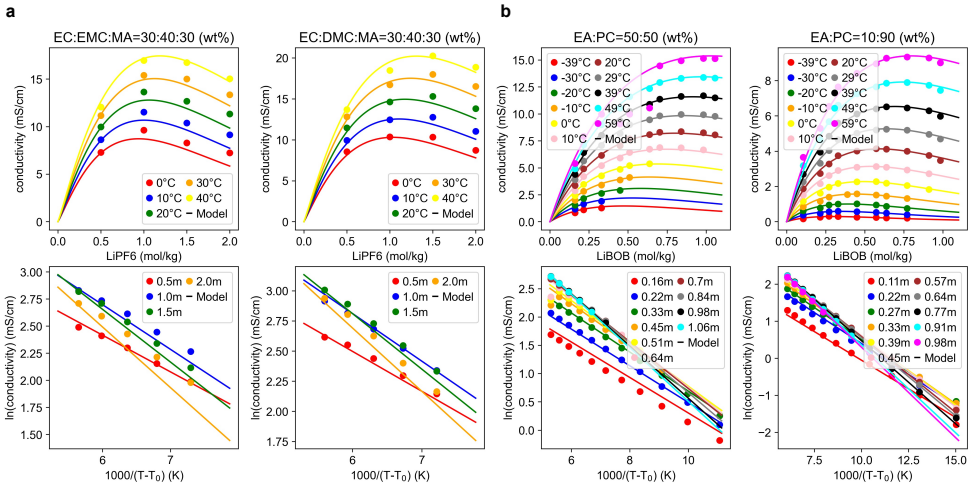

Fig. S7 More results of model predictions on temperature and concentration dependence of conductivity. a EC/DMC/MA/$LiPF_6$ and EC/EMC/MA/$LiPF_6$ systems. The dots are experimental data from Ref. [123] **b** EA/PC/LiBOB system. The experimental data are from Ref. [124].

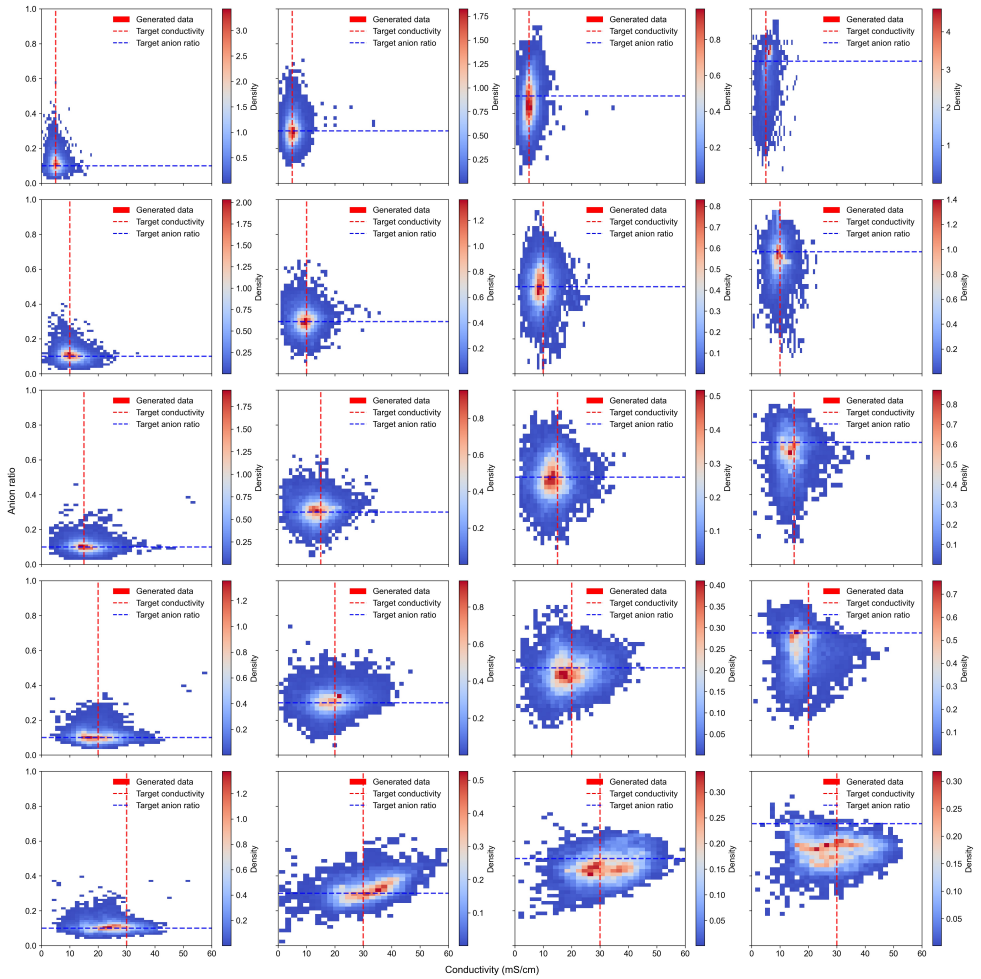

Fig. S8 More results of conditional generation. Different combinations of target conductivities (5.0, 10.0, 15.0, 20.0, 30.0 mS/cm) and anion ratios (0.1, 0.3, 0.5, 0.7) are tested.

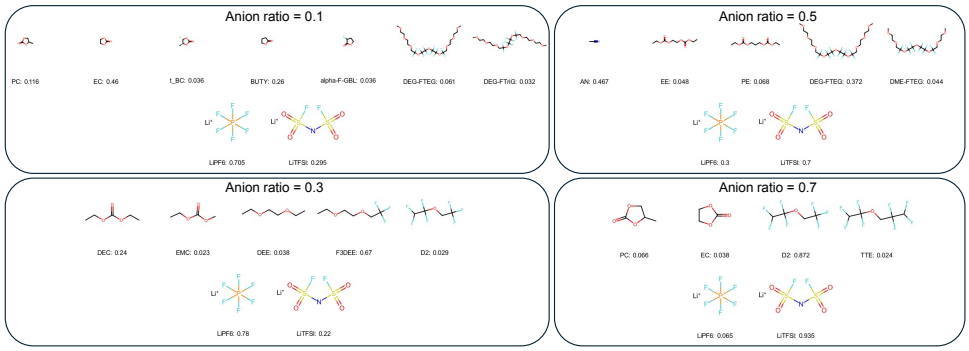

Fig. S9 Example of generated electrolyte formulation (conductivity = 5.0 mS/cm).

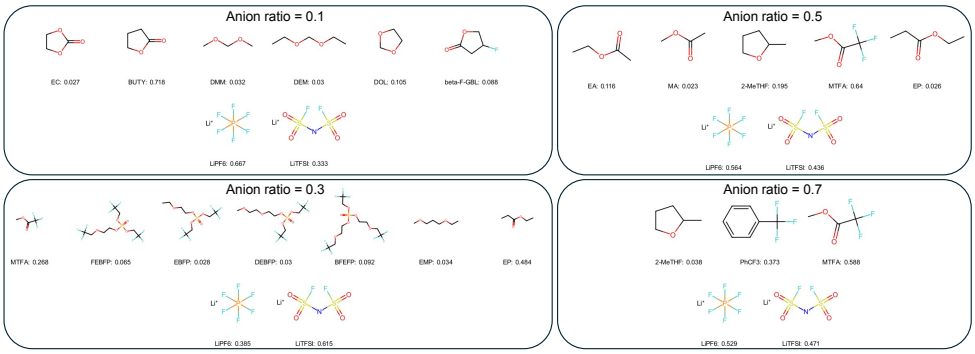

Fig. S10 Examples of generated electrolyte formulation (conductivity = 10.0 mS/cm).

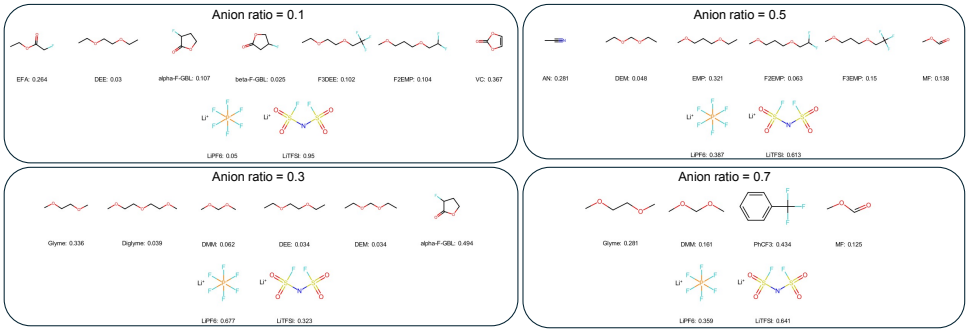

Fig. S11 Examples of generated electrolyte formulation (conductivity = 15.0 mS/cm).

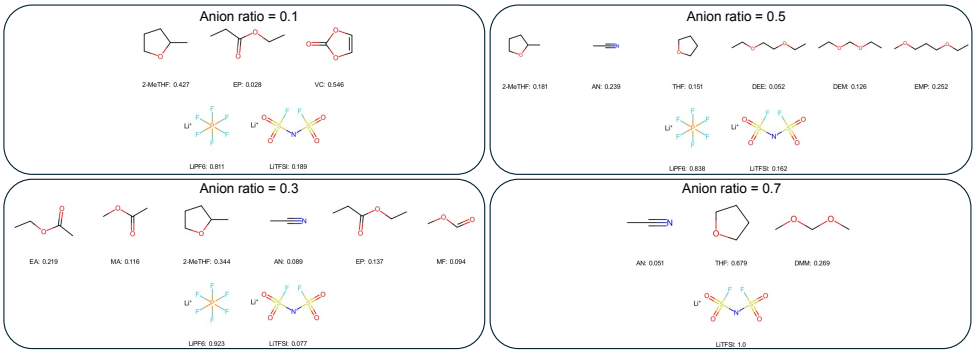

Fig. S12 Examples of generated electrolyte formulation (conductivity = 20.0 mS/cm).

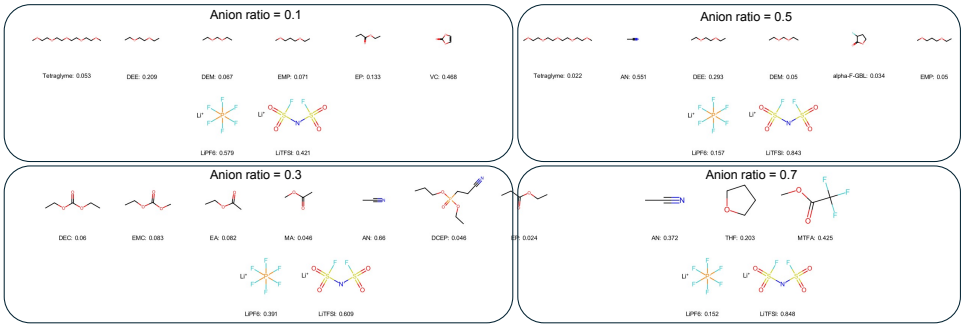

Fig. S13 Examples of generated electrolyte formulation (conductivity = 30.0 mS/cm).

References

[1] Xu, K.: Nonaqueous liquid electrolytes for lithium-based rechargeable batteries. Chem. Rev. **104**(10), 4303–4418 (2004)

[2] Meng, Y.S., Srinivasan, V., Xu, K.: Designing better electrolytes. Science **378**(6624), 3750 (2022)

[3] Wang, H., Yu, Z., Kong, X., Kim, S.C., Boyle, D.T., Qin, J., Bao, Z., Cui, Y.: Liquid electrolyte: The nexus of practical lithium metal batteries. Joule **6**(3), 588–616 (2022)

[4] Li, Z., Chen, Y., Yun, X., Gao, P., Zheng, C., Xiao, P.: Critical review of fluorinated electrolytes for high-performance lithium metal batteries. Advanced Functional Materials **33**(32), 2300502 (2023)

[5] Yu, Z., Wang, H., Kong, X., Huang, W., Tsao, Y., Mackanic, D.G., Wang, K., Wang, X., Huang, W., Choudhury, S., *et al.*: Molecular design for electrolyte solvents enabling energy-dense and long-cycling lithium metal batteries. Nature Energy **5**(7), 526–533 (2020)

[6] Chen, W., Park, J.-S., Kwon, C., Plaza-Rivera, C.O., Hsu, C.-W., Phong, J.K., Kilgallon, L.J., Wang, D., Dai, T., Kim, S.Y., et al.: Hybrid solvating electrolytes for practical sodium-metal batteries. Joule (2025)

[7] Kumar, R., Vu, M.C., Ma, P., Amanchukwu, C.: Electrolytomics: A unified big data approach for electrolyte design and discovery. ChemRxiv (2024) https://doi.org/10.26434/chemrxiv-2024-vqtc7 . This content is a preprint and has not been peer-reviewed.

[8] Blasio, P., Elsborg, J., Vegge, T., Flores, E., Bhowmik, A.: Calisol-23: Experimental electrolyte conductivity data for various li-salts and solvent combinations. Scientific Data **11**, 750 (2024) https://doi.org/10.1038/s41597-024-03575-8

[9] Bradford, G., Lopez, J., Ruza, J., Stolberg, M.A., Osterude, R., Johnson, J.A., Gomez-Bombarelli, R., Shao-Horn, Y.: Chemistry-informed machine learning for polymer electrolyte discovery. ACS Central Science **9**, 206–216 (2023) https://doi.org/10.1021/acscentsci.2c01123 . doi: 10.1021/acscentsci.2c01123

[10] Kim, S.C., Oyakhire, S.T., Athanitis, C., Wang, J., Zhang, Z., Zhang, W., Boyle, D.T., Kim, M.S., Yu, Z., Gao, X., Sogade, T., Wu, E., Qin, J., Bao, Z., Bent, S.F., Cui, Y.: Data-driven electrolyte design for lithium metal anodes. Proceedings of the National Academy of Sciences **120**, 2214357120 (2023) https://doi.org/10.1073/pnas.2214357120

[11] Jorgensen, W.L., Maxwell, D.S., Tirado-Rives, J.: Development and testing of

the opls all-atom force field on conformational energetics and properties of organic liquids. Journal of the American Chemical Society **118**, 11225–11236 (1996) https://doi.org/10.1021/ja9621760 . doi: 10.1021/ja9621760

[12] Doherty, B., Zhong, X., Gathiaka, S., Li, B., Acevedo, O.: Revisiting opls force field parameters for ionic liquid simulations. Journal of Chemical Theory and Computation **13**, 6131–6145 (2017) https://doi.org/10.1021/acs.jctc.7b00520 . doi: 10.1021/acs.jctc.7b00520

[13] Gong, S., Zhang, Y., Mu, Z., Pu, Z., Wang, H., Yu, Z., Chen, M., Zheng, T., Wang, Z., Chen, L., Wu, X., Shi, S., Gao, W., Yan, W., Xiang, L.: BAMBOO: a predictive and transferable machine learning force field framework for liquid electrolyte development (2024). https://arxiv.org/abs/2404.07181

[14] Wang, F., Cheng, J.: Understanding the solvation structures of glyme-based electrolytes by machine learning molecular dynamics. Chinese Journal of Structural Chemistry **42**, 100061 (2023) https://doi.org/10.1016/j.cjsc.2023.100061

[15] Dajnowicz, S., Agarwal, G., Stevenson, J.M., Jacobson, L.D., Ramezanghorbani, F., Leswing, K., Friesner, R.A., Halls, M.D., Abel, R.: High-dimensional neural network potential for liquid electrolyte simulations. The Journal of Physical Chemistry B **126**, 6271–6280 (2022) https://doi.org/10.1021/acs.jpcb.2c03746 . doi: 10.1021/acs.jpcb.2c03746

[16] Zhang, H., Lai, T., Chen, J., Manthiram, A., Rondinelli, J.M., Chen, W.: Learning molecular mixture property using chemistry-aware graph neural network. PRX Energy **3**, 023006 (2024) https://doi.org/10.1103/PRXEnergy.3.023006

[17] Zeng, B., Chen, S., Liu, X., Chen, C., Deng, B., Wang, X., Gao, Z., Zhang, Y., E, W., Zhang, L.: Uni-ELF: A Multi-Level Representation Learning Framework for Electrolyte Formulation Design (2024). https://arxiv.org/abs/2407.06152

[18] Chew, A.K., Afzal, M.A.F., Kaplan, Z., Collins, E.M., Gattani, S., Misra, M., Chandrasekaran, A., Leswing, K., Halls, M.D.: Leveraging high-throughput molecular simulations and machine learning for the design of chemical mixtures. npj Computational Materials **11**, 72 (2025) https://doi.org/10.1038/s41524-025-01552-2

[19] Faustov, A., Kovacs, A., Brown, D.: Improved Electrolyte for Electrochemical Cell. WO2021260274A1, December 30 2021. https://patents.google.com/patent/WO2021260274A1/en

[20] Cheng, G., Zhu, Y., Strand, D., Hallac, B., Metz, B.M.: Electrolyte Formulations for Lithium Ion Batteries. US9490503B1, November 8 2016. https://patents.google.com/patent/US9490503B1/en

[21] Wang, Q., Zhao, C., Wang, J., Yao, Z., Wang, S., Kumar, S.G.H., Ganapathy,

S., Eustace, S., Bai, X., Li, B., Wagemaker, M.: High entropy liquid electrolytes for lithium batteries. Nature Communications **14**, 440 (2023) https://doi.org/10.1038/s41467-023-36075-1

[22] Kim, S.C., Wang, J., Xu, R., Zhang, P., Chen, Y., Huang, Z., Yang, Y., Yu, Z., Oyakhire, S.T., Zhang, W., Greenburg, L.C., Kim, M.S., Boyle, D.T., Sayavong, P., Ye, Y., Qin, J., Bao, Z., Cui, Y.: High-entropy electrolytes for practical lithium metal batteries. Nature Energy **8**, 814–826 (2023) https://doi.org/10.1038/s41560-023-01280-1

[23] Wang, Q., Wang, J., Heringa, J.R., Bai, X., Wagemaker, M.: High-entropy electrolytes for lithium-ion batteries. ACS Energy Letters **9**, 3796–3806 (2024) https://doi.org/10.1021/acsenergylett.4c01358 . doi: 10.1021/acsenergylett.4c01358

[24] Zhu, S., Ramsundar, B., Annevelink, E., Lin, H., Dave, A., Guan, P.-W., Gering, K., Viswanathan, V.: Differentiable modeling and optimization of non-aqueous li-based battery electrolyte solutions using geometric deep learning. Nature Communications **15**, 8649 (2024) https://doi.org/10.1038/s41467-024-51653-7

[25] Bond-Taylor, S., Leach, A., Long, Y., Willcocks, C.G.: Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(11), 7327–7347 (2022) https://doi.org/10.1109/tpami.2021.3116668

[26] Liu, Y., Yang, Z., Yu, Z., Liu, Z., Liu, D., Lin, H., Li, M., Ma, S., Avdeev, M., Shi, S.: Generative artificial intelligence and its applications in materials science: Current situation and future perspectives. Journal of Materiomics **9**, 798–816 (2023) https://doi.org/10.1016/j.jmat.2023.05.001

[27] Sanchez-Lengeling, B., Aspuru-Guzik, A.: Inverse molecular design using machine learning: Generative models for matter engineering. Science **361**, 360–365 (2018) https://doi.org/10.1126/science.aat2663

[28] Machine learning-aided generative molecular design. Nature Machine Intelligence **6**, 589–604 (2024) https://doi.org/10.1038/s42256-024-00843-5

[29] Bilodeau, C., Jin, W., Jaakkola, T., Barzilay, R., Jensen, K.F.: Generative models for molecular discovery: Recent advances and challenges. WIREs Computational Molecular Science **12**, 1608 (2022) https://doi.org/10.1002/wcms.1608

[30] Xie, T., Fu, X., Ganea, O.-E., Barzilay, R., Jaakkola, T.: Crystal Diffusion Variational Autoencoder for Periodic Material Generation (2022). https://arxiv.org/abs/2110.06197

[31] Zeni, C., Pinsler, R., Zügner, D., Fowler, A., Horton, M., Fu, X., Wang, Z.,

Shysheya, A., Crabbé, J., Ueda, S., Sordillo, R., Sun, L., Smith, J., Nguyen, B., Schulz, H., Lewis, S., Huang, C.-W., Lu, Z., Zhou, Y., Yang, H., Hao, H., Li, J., Yang, C., Li, W., Tomioka, R., Xie, T.: A generative model for inorganic materials design. Nature (2025) https://doi.org/10.1038/s41586-025-08628-5

[32] Anstine, D.M., Isayev, O.: Generative models as an emerging paradigm in the chemical sciences. Journal of the American Chemical Society **145**, 8736–8750 (2023) https://doi.org/10.1021/jacs.2c13467 . doi: 10.1021/jacs.2c13467

[33] Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., Yang, L.: Physics-informed machine learning. Nature Reviews Physics **3**, 422–440 (2021) https://doi.org/10.1038/s42254-021-00314-5

[34] Weininger, D.: Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. Journal of Chemical Information and Computer Sciences **28**, 31–36 (1988) https://doi.org/10.1021/ci00057a005 . doi: 10.1021/ci00057a005

[35] Zheng, T., Wang, A., Han, X., Xia, Y., Xu, X., Zhan, J., Liu, Y., Chen, Y., Wang, Z., Wu, X., Gong, S., Yan, W.: Data-driven parametrization of molecular mechanics force fields for expansive chemical space coverage. Chem. Sci. **16**, 2730–2740 (2025) https://doi.org/10.1039/D4SC06640E

[36] Hussain, M.S., Zaki, M.J., Subramanian, D.: Global self-attention as a replacement for graph convolution. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. KDD '22, pp. 655–665. Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3534678.3539296 . https://doi.org/10.1145/3534678.3539296

[37] Kontogeorgis, G.M., Maribo-Mogensen, B., Thomsen, K.: The debye-hückel theory and its importance in modeling electrolyte solutions. Fluid Phase Equilibria **462**, 130–152 (2018) https://doi.org/10.1016/j.fluid.2018.01.004

[38] Casteel, J.F., Amis, E.S.: Specific conductance of concentrated solutions of magnesium salts in water-ethanol system. Journal of Chemical & Engineering Data **17**, 55–59 (1972) https://doi.org/10.1021/je60052a029 . doi: 10.1021/je60052a029

[39] Ngai, K.: Relaxation and Diffusion in Complex Systems. Springer, New York, NY, USA (2011)

[40] Fu, Y., Cui, X., Zhang, Y., Feng, T., He, J., Zhang, X., Bai, X., Cheng, Q.: Measurement and correlation of the electrical conductivity of the ionic liquid [bmim][tfsi] in binary organic solvents. Journal of Chemical & Engineering Data **63**, 1180–1189 (2018) https://doi.org/10.1021/acs.jced.7b00646 . doi: 10.1021/acs.jced.7b00646

[41] Zhang, W., Chen, X., Wang, Y., Wu, L., Hu, Y.: Experimental and modeling of conductivity for electrolyte solution systems. ACS Omega **5**, 22465–22474 (2020) https://doi.org/10.1021/acsomega.0c03013 . doi: 10.1021/acsomega.0c03013

[42] Zhang, Q.: In: Zhang, S. (ed.) Electroconductivity of Ionic Liquids, pp. 358–364. Springer, ??? (2022). https://doi.org/10.1007/978-981-33-4221-7_110

[43] Xu, K.: Nonaqueous liquid electrolytes for lithium-based rechargeable batteries. Chemical Reviews **104**, 4303–4418 (2004) https://doi.org/10.1021/cr030203g . doi: 10.1021/cr030203g

[44] Tikekar, M.D., Choudhury, S., Tu, Z., Archer, L.A.: Design principles for electrolytes and interfaces for stable lithium-metal batteries. Nature Energy **1**, 16114 (2016) https://doi.org/10.1038/nenergy.2016.114

[45] Yu, Z., Wang, H., Kong, X., Huang, W., Tsao, Y., Mackanic, D.G., Wang, K., Wang, X., Huang, W., Choudhury, S., Zheng, Y., Amanchukwu, C.V., Hung, S.T., Ma, Y., Lomeli, E.G., Qin, J., Cui, Y., Bao, Z.: Molecular design for electrolyte solvents enabling energy-dense and long-cycling lithium metal batteries. Nature Energy **5**, 526–533 (2020) https://doi.org/10.1038/s41560-020-0634-5

[46] Li, Z., Rao, H., Atwi, R., Sivakumar, B.M., Gwalani, B., Gray, S., Han, K.S., Everett, T.A., Ajantiwalay, T.A., Murugesan, V., Rajput, N.N., Pol, V.G.: Non-polar ether-based electrolyte solutions for stable high-voltage non-aqueous lithium metal batteries. Nature Communications **14**, 868 (2023) https://doi.org/10.1038/s41467-023-36647-1

[47] Chen, Y., Liao, S.-L., Gong, H., Zhang, Z., Huang, Z., Kim, S.C., Zhang, E., Lyu, H., Yu, W., Lin, Y., Sayavong, P., Cui, Y., Qin, J., Bao, Z.: Hyperconjugation-controlled molecular conformation weakens lithium-ion solvation and stabilizes lithium metal anodes. Chem. Sci. **15**, 19805–19819 (2024) https://doi.org/10.1039/D4SC05319B

[48] Wu, L.-Q., Li, Z., Fan, Z.-Y., Li, K., Li, J., Huang, D., Li, A., Yang, Y., Xie, W., Zhao, Q.: Unveiling the role of fluorination in hexacyclic coordinated ether electrolytes for high-voltage lithium metal batteries. Journal of the American Chemical Society **146**, 5964–5976 (2024) https://doi.org/10.1021/jacs.3c11798 . doi: 10.1021/jacs.3c11798

[49] Emilsson, S., Albuquerque, M., Öberg, P., Brandell, D., Johansson, M.: Understanding ion transport in alkyl dicarbonates: An experimental and computational study. ACS Physical Chemistry Au **5**, 80–91 (2025) https://doi.org/10.1021/acsphyschemau.4c00078 . doi: 10.1021/acsphyschemau.4c00078

[50] Mistry, A., Yu, Z., Cheng, L., Srinivasan, V.: On relative importance of vehicular and structural motions in defining electrolyte transport. Journal of The Electrochemical Society **170**, 110536 (2023) https://doi.org/10.1149/1945-7111/

ad0c66

[51] Dhariwal, P., Nichol, A.: Diffusion Models Beat GANs on Image Synthesis (2021). https://arxiv.org/abs/2105.05233

[52] Wang, H., Yu, Z., Kong, X., Kim, S.C., Boyle, D.T., Qin, J., Bao, Z., Cui, Y.: Liquid electrolyte: The nexus of practical lithium metal batteries. Joule **6**, 588–616 (2022) https://doi.org/10.1016/j.joule.2021.12.018 . doi: 10.1016/j.joule.2021.12.018

[53] Teng, W., Wu, J., Liang, Q., Deng, J., Xu, Y., Liu, Q., Wang, B., Ma, T., Nan, D., Liu, J., Li, B., Weng, Q., Yu, X.: Designing advanced liquid electrolytes for alkali metal batteries: Principles, progress, and perspectives. ENERGY & ENVIRONMENTAL MATERIALS **6**, 12355 (2023) https://doi.org/10.1002/eem2.12355 . e12355 EEM-2021-0828.R1

[54] Gómez-Bombarelli, R., Wei, J.N., Duvenaud, D., Hernández-Lobato, J.M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T.D., Adams, R.P., Aspuru-Guzik, A.: Automatic chemical design using a data-driven continuous representation of molecules. ACS Central Science **4**, 268–276 (2018) https://doi.org/10.1021/acscentsci.7b00572 . doi: 10.1021/acscentsci.7b00572

[55] Bagal, V., Aggarwal, R., Vinod, P.K., Priyakumar, U.D.: Molgpt: Molecular generation using a transformer-decoder model. Journal of Chemical Information and Modeling **62**, 2064–2076 (2022) https://doi.org/10.1021/acs.jcim.1c00600 . doi: 10.1021/acs.jcim.1c00600

[56] Kuzhagaliyeva, N., Horváth, S., Williams, J., Nicolle, A., Sarathy, S.M.: Artificial intelligence-driven design of fuel mixtures. Communications Chemistry **5**(1), 111 (2022)

[57] Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A.J., Bambrick, J., Bodenstein, S.W., Evans, D.A., Hung, C.-C., O'Neill, M., Reiman, D., Tunyasuvunakool, K., Wu, Z., Žemgulytė, A., Arvaniti, E., Beattie, C., Bertolli, O., Bridgland, A., Cherepanov, A., Congreve, M., Cowen-Rivers, A.I., Cowie, A., Figurnov, M., Fuchs, F.B., Gladman, H., Jain, R., Khan, Y.A., Low, C.M.R., Perlin, K., Potapenko, A., Savy, P., Singh, S., Stecula, A., Thillaisundaram, A., Tong, C., Yakneen, S., Zhong, E.D., Zielinski, M., Žídek, A., Bapst, V., Kohli, P., Jaderberg, M., Hassabis, D., Jumper, J.M.: Accurate structure prediction of biomolecular interactions with alphafold 3. Nature **630**, 493–500 (2024) https://doi.org/10.1038/s41586-024-07487-w

[58] George, E.P., Raabe, D., Ritchie, R.O.: High-entropy alloys. Nature reviews materials **4**(8), 515–534 (2019)

[59] Ge, W., Silva, R.D., Fan, Y., Sisson, S.A., Stenzel, M.H.: Machine learning in

polymer research. Advanced Materials **37**, 2413695 (2025) https://doi.org/10.1002/adma.202413695

[60] Koutsoukos, S., Philippi, F., Malaret, F., Welton, T.: A review on machine learning algorithms for the ionic liquid chemical space. Chem. Sci. **12**, 6820–6843 (2021) https://doi.org/10.1039/D1SC01000J

[61] Hansen, B.B., Spittle, S., Chen, B., Poe, D., Zhang, Y., Klein, J.M., Horton, A., Adhikari, L., Zelovich, T., Doherty, B.W., Gurkan, B., Maginn, E.J., Ragauskas, A., Dadmun, M., Zawodzinski, T.A., Baker, G.A., Tuckerman, M.E., Savinell, R.F., Sangoro, J.R.: Deep eutectic solvents: A review of fundamentals and applications. Chemical Reviews **121**, 1232–1285 (2021) https://doi.org/10.1021/acs.chemrev.0c00385 . doi: 10.1021/acs.chemrev.0c00385

[62] Machine learning for computational heterogeneous catalysis. ChemCatChem **11**, 3581–3601 (2019) https://doi.org/10.1002/cctc.201900595

[63] Morgan, H.L.: The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. Journal of Chemical Documentation **5**, 107–113 (1965) https://doi.org/10.1021/c160017a018 . doi: 10.1021/c160017a018

[64] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph Attention Networks. International Conference on Learning Representations (2018). accepted as poster

[65] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention Is All You Need (2023). https://arxiv.org/abs/1706.03762

[66] Zhang, H., Lai, T., Chen, J., Manthiram, A., Rondinelli, J.M., Chen, W.: Learning molecular mixture property using chemistry-aware graph neural network. PRX Energy **3**(2) (2024) https://doi.org/10.1103/prxenergy.3.023006

[67] Walden, P.: Über organische lösungs- und ionisierungsmittel. Zeitschrift für Physikalische Chemie **55U**(1), 207–249 (1906) https://doi.org/10.1515/zpch-1906-5511

[68] Gering, K.L.: Prediction of electrolyte viscosity for aqueous and non-aqueous systems: Results from a molecular model based on ion solvation and a chemical physics framework. Electrochimica Acta **51**, 3125–3138 (2006) https://doi.org/10.1016/j.electacta.2005.09.011

[69] Gering, K.L.: Prediction of electrolyte conductivity: Results from a generalized molecular model based on ion solvation and a chemical physics framework. Electrochimica Acta **225**, 175–189 (2017) https://doi.org/10.1016/j.electacta.2016.12.083

[70] Ding, M.S., Xu, K., Zhang, S.S., Amine, K., Henriksen, G.L., Jow, T.R.: Change of conductivity with salt content, solvent composition, and temperature for electrolytes of lipf6 in ethylene carbonate-ethyl methyl carbonate. Journal of The Electrochemical Society **148**(10), 1196 (2001) https://doi.org/10.1149/1.1403730

[71] Ho, J., Jain, A., Abbeel, P.: Denoising Diffusion Probabilistic Models (2020). https://arxiv.org/abs/2006.11239

[72] Lee, J., Lee, Y., Kim, J., Kosiorek, A.R., Choi, S., Teh, Y.W.: Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks (2019). https://arxiv.org/abs/1810.00825

[73] RDKit: Open-Source Cheminformatics. https://www.rdkit.org

[74] Peach, M.L., Nicklaus, M.C.: Chemoinformatics at the CADD Group of the National Cancer Institute, pp. 385–393. John Wiley & Sons, Ltd, Chichester, UK (2018). https://doi.org/10.1002/9783527806539.ch6k . https://onlinelibrary.wiley.com/doi/abs/10.1002/9783527806539.ch6k

[75] Mitra, S.S., Srivastava, S.N.: Dependence of surface tension on temperature. The Journal of Chemical Physics **22**, 1134–1135 (1954) https://doi.org/10.1063/1.1740282

[76] Brown, O.L.I.: The clausius-clapeyron equation. Journal of Chemical Education **28**, 428 (1951) https://doi.org/10.1021/ed028p428 . doi: 10.1021/ed028p428

[77] Thompson, A.P., Aktulga, H.M., Berger, R., Bolintineanu, D.S., Brown, W.M., Crozier, P.S., Veld, P.J., Kohlmeyer, A., Moore, S.G., Nguyen, T.D., Shan, R., Stevens, M.J., Tranchida, J., Trott, C., Plimpton, S.J.: LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. Comp. Phys. Comm. **271**, 108171 (2022) https://doi.org/10.1016/j.cpc.2021.108171

[78] Soetens, J.-C., Millot, C., Maigret, B.: Molecular dynamics simulation of li+ bf4- in ethylene carbonate, propylene carbonate, and dimethyl carbonate solvents. The Journal of Physical Chemistry A **102**(7), 1055–1061 (1998)

[79] Sambasivarao, S.V., Acevedo, O.: Development of opls-aa force field parameters for 68 unique ionic liquids. Journal of chemical theory and computation **5**(4), 1038–1050 (2009)

[80] Breneman, C.M., Wiberg, K.B.: Determining atom-centered monopoles from molecular electrostatic potentials. the need for high sampling density in formamide conformational analysis. J. Comput. Chem. **11**(3), 361–373 (1990) https://doi.org/10.1002/jcc.540110311

[81] Wu, X., Sun, Q., Pu, Z., Zheng, T., Ma, W., Yan, W., Xia, Y., Wu, Z., Huo, M., Li, X., Ren, W., Gong, S., Zhang, Y., Gao, W.: Enhancing gpu-acceleration in the python-based simulations of chemistry frameworks. WIREs Computational Molecular Science **15**(2), 70008 (2025) https://doi.org/10.1002/wcms.70008 https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wcms.70008. e70008 CMS-1146.R2

[82] Sasaki, Y.: Chapter 13 - physical and electrochemical properties and application to lithium batteries of fluorinated organic solvents. In: Nakajima, T., Groult, H. (eds.) Fluorinated Materials for Energy Conversion, pp. 285–304. Elsevier Science, Amsterdam (2005). https://doi.org/10.1016/B978-008044472-7/50041-2 . https://www.sciencedirect.com/science/article/pii/B9780080444727500412

[83] Yang, C., Zhou, X., Sun, R., Hu, W., Wang, M., Dong, X., Piao, N., Han, J., Chen, W., You, Y.: A safe electrolyte enriched with flame-retardant solvents for high-voltage licoo2——graphite pouch cells. ACS Energy Letters **9**, 5364–5372 (2024) https://doi.org/10.1021/acsenergylett.4c02466 . doi: 10.1021/acsenergylett.4c02466

[84] Yu, Z., Rudnicki, P.E., Zhang, Z., Huang, Z., Celik, H., Oyakhire, S.T., Chen, Y., Kong, X., Kim, S.C., Xiao, X., Wang, H., Zheng, Y., Kamat, G.A., Kim, M.S., Bent, S.F., Qin, J., Cui, Y., Bao, Z.: Rational solvent molecule tuning for high-performance lithium metal battery electrolytes. Nature Energy **7**, 94–106 (2022) https://doi.org/10.1038/s41560-021-00962-y

[85] Amanchukwu, C.V., Yu, Z., Kong, X., Qin, J., Cui, Y., Bao, Z.: A new class of ionically conducting fluorinated ether electrolytes with high electrochemical stability. Journal of the American Chemical Society **142**, 7393–7403 (2020) https://doi.org/10.1021/jacs.9b11056 . doi: 10.1021/jacs.9b11056

[86] Huang, R., Guo, X., Chen, B., Ma, M., Chen, Q., Zhang, C., Liu, Y., Kong, X., Fan, X., Wang, L., Ling, M., Pan, H.: Electrolyte design chart reframed by intermolecular interactions for high-performance li–ion batteries. JACS Au **4**, 1986–1996 (2024) https://doi.org/10.1021/jacsau.4c00196 . doi: 10.1021/jacsau.4c00196

[87] Chen, L., Lu, J., Wang, Y., He, P., Huang, S., Liu, Y., Wu, Y., Cao, G., Wang, L., He, X., Qiu, J., Zhang, H.: Double-salt electrolyte for li-ion batteries operated at elevated temperatures. Energy Storage Materials **49**, 493–501 (2022) https://doi.org/10.1016/j.ensm.2022.04.036

[88] Cui, Z., Liu, C., Manthiram, A.: Enabling stable operation of lithium-ion batteries under fast-operating conditions by tuning the electrolyte chemistry. Advanced Materials **36**, 2409272 (2024) https://doi.org/10.1002/adma.202409272

[89] Xu, Z., Zhang, X., Yang, J., Cui, X., Nuli, Y., Wang, J.: High-voltage and

intrinsically safe electrolytes for li metal batteries. Nature Communications **15**, 9856 (2024) https://doi.org/10.1038/s41467-024-51958-7

[90] Chen, Y., Liao, S.-L., Gong, H., Zhang, Z., Huang, Z., Kim, S.C., Zhang, E., Lyu, H., Yu, W., Lin, Y., Sayavong, P., Cui, Y., Qin, J., Bao, Z.: Hyperconjugation-controlled molecular conformation weakens lithium-ion solvation and stabilizes lithium metal anodes. Chem. Sci. **15**, 19805–19819 (2024) https://doi.org/10.1039/D4SC05319B

[91] Liang, P., Li, J., Dong, Y., Wang, Z., Ding, G., Liu, K., Xue, L., Cheng, F.: Modulating interfacial solvation via ion dipole interactions for low-temperature and high-voltage lithium batteries. Angewandte Chemie International Edition **64**, 202415853 (2025) https://doi.org/10.1002/anie.202415853

[92] Cui, Z., Wang, D., Guo, J., Nian, Q., Ruan, D., Fan, J., Ma, J., Li, L., Dong, Q., Luo, X., Wang, Z., Ou, X., Cao, R., Jiao, S., Ren, X.: Push–pull electrolyte design strategy enables high-voltage low-temperature lithium metal batteries. Journal of the American Chemical Society **146**, 27644–27654 (2024) https://doi.org/10.1021/jacs.4c09027 . doi: 10.1021/jacs.4c09027

[93] Fan, Z., Zhang, J., Wu, L., Yu, H., Li, J., Li, K., Zhao, Q.: Solvation structure dependent ion transport and desolvation mechanism for fast-charging li-ion batteries. Chem. Sci. **15**, 17161–17172 (2024) https://doi.org/10.1039/D4SC05464D

[94] Zhao, Y., Hu, Z., Zhao, Z., Chen, X., Zhang, S., Gao, J., Luo, J.: Strong solvent and dual lithium salts enable fast-charging lithium-ion batteries operating from 78 to 60 c. Journal of the American Chemical Society **145**, 22184–22193 (2023) https://doi.org/10.1021/jacs.3c08313 . doi: 10.1021/jacs.3c08313

[95] Tan, S., Borodin, O., Wang, N., Yen, D., Weiland, C., Hu, E.: Synergistic anion and solvent-derived interphases enable lithium-ion batteries under extreme conditions. Journal of the American Chemical Society **146**, 30104–30116 (2024) https://doi.org/10.1021/jacs.4c07806 . doi: 10.1021/jacs.4c07806

[96] Wu, S., Liu, X., Hao, Z., Sun, X., Hou, J., Shang, L., Wang, L., Zhang, K., Li, H., Yan, Z., Chen, J.: Uncovering the crucial role of chelating structures in cyanoalkyl-phosphate electrolytes for high-voltage lithium metal batteries. Journal of the American Chemical Society **146**, 28770–28782 (2024) https://doi.org/10.1021/jacs.4c07739 . doi: 10.1021/jacs.4c07739

[97] Wu, L.-Q., Li, Z., Fan, Z.-Y., Li, K., Li, J., Huang, D., Li, A., Yang, Y., Xie, W., Zhao, Q.: Unveiling the role of fluorination in hexacyclic coordinated ether electrolytes for high-voltage lithium metal batteries. Journal of the American Chemical Society **146**, 5964–5976 (2024) https://doi.org/10.1021/jacs.3c11798 . doi: 10.1021/jacs.3c11798

[98] Nambu, N., Sasaki, Y.: Physical and electrolytic properties of monofluorinated ethyl acetates and their application to lithium secondary batteries. Open Journal of Metal **5**(1), 1–9 (2015)

[99] Wang, J., Yamada, Y., Sodeyama, K., Chiang, C.H., Tateyama, Y., Yamada, A.: Superconcentrated electrolytes for a high-voltage lithium-ion battery. Nature Communications **7**, 12032 (2016) https://doi.org/10.1038/ncomms12032

[100] Dave, A., Mitchell, J., Burke, S., Lin, H., Whitacre, J., Viswanathan, V.: Autonomous optimization of non-aqueous li-ion battery electrolytes via robotic experimentation and machine learning coupling. Nature Communications **13**, 5454 (2022) https://doi.org/10.1038/s41467-022-32938-1

[101] Piao, N., Ji, X., Xu, H., Fan, X., Chen, L., Liu, S., Garaga, M.N., Greenbaum, S.G., Wang, L., Wang, C., He, X.: Countersolvent electrolytes for lithium-metal batteries. Advanced Energy Materials **10**, 1903568 (2020) https://doi.org/10.1002/aenm.201903568

[102] Every, H., Bishop, A.G., Forsyth, M., MacFarlane, D.R.: Ion diffusion in molten salt mixtures. Electrochimica Acta **45**, 1279–1284 (2000) https://doi.org/10.1016/S0013-4686(99)00332-1

[103] Ding, M.S., Xu, K.: Phase diagram, conductivity, and glass transition of litfsi–h2o binary electrolytes. The Journal of Physical Chemistry C **122**, 16624–16629 (2018) https://doi.org/10.1021/acs.jpcc.8b05193 . doi: 10.1021/acs.jpcc.8b05193

[104] HIRAYAMA, H., TACHIKAWA, N., YOSHII, K., WATANABE, M., KATAYAMA, Y.: Ionic conductivity and viscosity of solvate ionic liquids composed of glymes and excess lithium bis(trifluoromethylsulfonyl)amide. Electrochemistry **83**(10), 824–827 (2015) https://doi.org/10.5796/electrochemistry.83.824

[105] Chagnes, A., Carré, B., Willmann, P., Lemordant, D.: Ion transport theory of nonaqueous electrolytes. liclo4 in γ-butyrolactone: the quasi lattice approach. Electrochimica Acta **46**, 1783–1791 (2001) https://doi.org/10.1016/S0013-4686(00)00718-0

[106] GRUNBERG, L., NISSAN, A.H.: Mixture law for viscosity. Nature **164**, 799–800 (1949) https://doi.org/10.1038/164799b0

[107] Boehm, R.C., Hauck, F., Yang, Z., Wanstall, C.T., Heyne, J.S.: Error quantification of the arrhenius blending rule for viscosity of hydrocarbon mixtures. Frontiers in Energy Research **10** (2022) https://doi.org/10.3389/fenrg.2022.1074699

[108] Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, İ., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., SciPy 1.0 Contributors: SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods **17**, 261–272 (2020) https://doi.org/10.1038/s41592-019-0686-2

[109] Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation (2015). https://arxiv.org/abs/1505.04597

[110] Wang, P.: Denoising Diffusion Probabilistic Models in PyTorch. https://github.com/lucidrains/denoising-diffusion-pytorch. GitHub repository (2023)

[111] Perez, E., Strub, F., Vries, H., Dumoulin, V., Courville, A.: FiLM: Visual Reasoning with a General Conditioning Layer (2017). https://arxiv.org/abs/1709.07871

[112] Tetko, I.V., Lowe, D.M., Williams, A.J.: The development of models to predict melting and pyrolysis point data associated with several hundred thousand compounds mined from patents. Journal of Cheminformatics **8**, 2 (2016) https://doi.org/10.1186/s13321-016-0113-y

[113] Bradley, J.-C., Williams, A., Lang, A.: Jean-Claude Bradley Open Melting Point Dataset (2014) https://doi.org/10.6084/m9.figshare.1031637.v2

[114] https://www.cas.org/resources/gated-content/cas-covid-19-antiviral-candidate-sar-dataset

[115] Lide, D.R.: CRC Handbook of Chemistry and Physics vol. 85. CRC Press, Boca Raton, FL (2004)

[116] Mansouri, K., Grulke, C.M., Judson, R.S., Williams, A.J.: Opera models for predicting physicochemical properties and environmental fate endpoints. Journal of Cheminformatics **10**, 10 (2018) https://doi.org/10.1186/s13321-018-0263-1

[117] Open-source qsar models for pka prediction using multiple machine learning approaches. Journal of Cheminformatics **11**, 60 (2019) https://doi.org/10.1186/s13321-019-0384-1

[118] Bouteloup, R., Mathieu, D.: Predicting dielectric constants of pure liquids: fragment-based kirkwood–fröhlich model applicable over a wide range of polarity. Phys. Chem. Chem. Phys. **21**, 11043–11057 (2019) https://doi.org/10.1039/C9CP01704F

[119] Wohlfarth, C., Wohlfarth, B.: Pure Liquids: Data: Datasheet from Landolt-Börnstein - Group IV Physical Chemistry · Volume 16: "Surface Tension of Pure Liquids and Binary Liquid Mixtures" in SpringerMaterials (https://doi.org/10.1007/10560191_2). Springer. Copyright 1997 Springer-Verlag Berlin Heidelberg. https://doi.org/10.1007/10560191_2

[120] Goussard, V., Duprat, F., Ploix, J.-L., Dreyfus, G., Nardello-Rataj, V., Aubry, J.-M.: A new machine-learning tool for fast estimation of liquid viscosity. application to cosmetic oils. Journal of Chemical Information and Modeling **60**, 2012–2023 (2020) https://doi.org/10.1021/acs.jcim.0c00083 . doi: 10.1021/acs.jcim.0c00083

[121] Chew, A.K., Sender, M., Kaplan, Z., Chandrasekaran, A., Elk, J.C., Browning, A.R., Kwak, H.S., Halls, M.D., Afzal, M.A.F.: Advancing material property prediction: using physics-informed machine learning models for viscosity. Journal of Cheminformatics **16**, 31 (2024) https://doi.org/10.1186/s13321-024-00820-5

[122] Gharagheizi, F., Eslamimanesh, A., Ilani-Kashkouli, P., Mohammadi, A.H., Richon, D.: Determination of vapor pressure of chemical compounds: A group contribution model for an extremely large database. Industrial & Engineering Chemistry Research **51**, 7119–7125 (2012) https://doi.org/10.1021/ie3002099 . doi: 10.1021/ie3002099

[123] Logan, E.R., Tonita, E.M., Gering, K.L., Li, J., Ma, X., Beaulieu, L.Y., Dahn, J.R.: A study of the physical properties of li-ion battery electrolytes containing esters. Journal of The Electrochemical Society **165**(2), 21 (2018) https://doi.org/10.1149/2.0271802jes

[124] Ding, M.S., Jow, T.R.: Properties of pc-ea solvent and its solution of libob comparison of linear esters to linear carbonates for use in lithium batteries. Journal of The Electrochemical Society **152**(6), 1199 (2005) https://doi.org/10.1149/1.1914757