Supplementary Information

Generalizing property prediction of ionic liquids from limited labeled data: a one-stop framework empowered by transfer learning

Guzhong Chen,[a,b] Zhen Song,[a,*] Zhiwen Qi[a,*], Kai Sundmacher[b,c]

[a] *State Key Laboratory of Chemical Engineering, School of Chemical Engineering, East China University of Science and Technology, 130 Meilong Road, Shanghai 200237, China*

[b] *Process Systems Engineering, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, D-39106 Magdeburg, Germany*

[c] *Process Systems Engineering, Otto-von-Guericke University Magdeburg, Universitätsplatz 2, D-39106 Magdeburg, Germany*

Corresponding authors: songz@ecust.edu.cn (Z. S.); zwqi@ecust.edu.cn (Z. Q.)

1 Supplementary Note1:

1.1 Encoder–decoder architecture

As an instance of the encoder-decoder architecture, the overall architecture of the transformer is presented in Figure S1. The encoder maps an input sequence of symbol representations $(x_1, \ldots, x_n)$ to a sequence of continuous representations $\mathbf{z} = (z_1, \ldots, z_n)$. Given $\mathbf{z}$, the decoder then generates an output sequence $(y_1, \ldots, y_m)$ of symbols, one element at a time. At each step, the model is auto-regressive, consuming the previously generated symbols as additional input when generating the next. In the encoder, the multi-head attention layers attend the input sequence and encode it into a hidden representation carrying the essential information, namely encoder state. The decoder consists of two types of multi-head attention layers: the first is masked and attends only the preceding outputs of the decoder, while the second multi-head attention layer attends encoder states as well as the output of the first decoder attention layer. It basically combines the information of the source sequence with the target sequence that has been produced so far. The SMILES Transformer model in this work utilized three Transformer blocks for both encoder and decoder, that is to say, N is 3 in Figure S1.

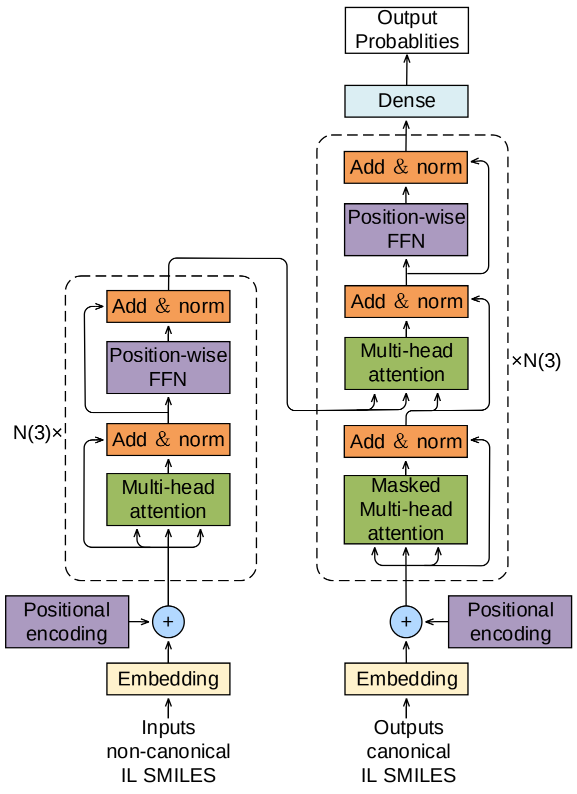

Figure S1. Architecture of the SMILES Transformer model used in this work. The left-half corresponds to the encoder while the right-half corresponds to the decoder.

1.2 Multi-head Attention

As the most important part of the Transformer architecture, the attention mechanism allows the model to focus on different tokens in the sequence at different stages of the network, enabling it to discover multiple relationships between groups of tokens. The attention function used here is called "Scaled-Dot Product Attention"[1] and can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. The input consists of queries and keys of dimension d_k, and values of dimension d_v. The dot products of the query with all keys are computed, and then divided by $\sqrt{d_k}$. A softmax function is then applied to obtain the weights on the values. In practice, the attention function is computed on a set of queries simultaneously, packed together into a matrix *Q*. The keys and values are also packed together into matrices *K* and *V*. The matrix of outputs is:

$$\text{Attention }(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

The attention score as computed above determines the importance that should be given to different parts of an input sequence in the current context. In order to allow the model to jointly factor in information from different representation subspaces at different positions, multi-headed attention is used. Multiple attention scores are first calculated in parallel and then concatenated and projected using a linear transformation as:

$$\text{MultiHead }(Q, K, V) = \text{Concat }(\text{head}_1, \ldots, \text{head}_\text{h})W^O$$
$$\text{where head}_\text{i} = \text{ Attention }\left(QW_i^Q, KW_i^K, VW_i^V\right)$$

where the projections are parameter matrix $W_i^Q \in \mathbb{R}^{d_\text{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_\text{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_\text{model} \times d_v}$ and $W^O \in \mathbb{R}^{hd_v \times d_\text{model}}$.

1.3 Positional encoding

Unlike RNNs that recurrently process tokens of a sequence one by one, self-attention ditches sequential operations in favor of parallel computation. To use the sequence order information, absolute or relative positional information is injected by adding positional encoding to the input representations. Positional encodings can be either learned or fixed. In the following, a fixed positional encoding based on sine and cosine functions is described.

Suppose that the input representation $X \in \mathbb{R}^{n\times d}$ contains the d-dimensional embeddings for n tokens of a sequence. The positional encoding outputs X+P using a positional embedding matrix $P \in \mathbb{R}^{n\times d}$ of the same shape, whose element on the i^{th} row and the $(2j)^{\text{th}}$ and the $(2j+1)^{\text{th}}$ column is:

$$p_{i,2j} = \sin\left(\frac{i}{10000^{2j/d}}\right)$$
$$p_{i,2j+1} = \cos\left(\frac{i}{10000^{2j/d}}\right)$$

In the positional embedding matrix ***P***, rows correspond to positions within a sequence and columns represent different positional encoding dimensions.

2 Supplementary Note2

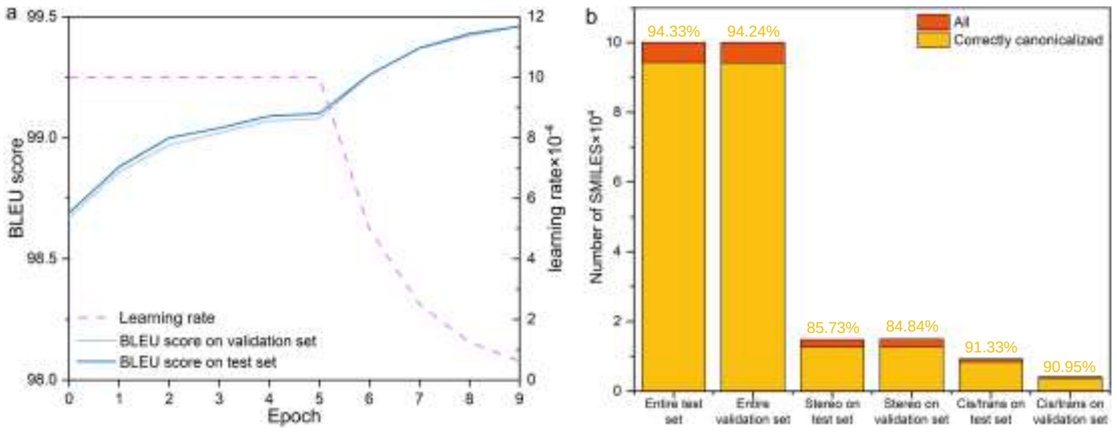

Figure S2. Performance of IL Transformer. (a) Learning curves for the validation and test sets, display the learning rate and the BLEU score with the training epochs. **(b)** Number of accurate canonical SMILES matches between the model-predicted and actual canonical SMILES for the test and validation sets. The IL Transformer displays remarkable performance, achieving stable validation and test BLEU scores of nearly 99.5 after the eighth epoch. Additionally, the model attains over 94% accurate canonicalized SMILES for both validation and test sets, including molecules with stereo- or cis/trans conformers, with translation accuracy higher than 84% and 90%, respectively.

3 Supplementary Note3

The characters as well as their indexes in the vocabulary are: [('<unk>', 0), ('<pad>', 1), ('<bos>', 2), ('<eos>', 3), ('c', 4), ('C', 5), ('(', 6), (')', 7), ('O', 8), ('1', 9), ('=', 10), ('N', 11), ('[', 12), (']', 13), ('2', 14), ('-', 15), ('+', 16), ('n', 17), ('3', 18), ('H', 19), ('@', 20), ('F', 21), ('S', 22), ('.', 23), ('l', 24), ('/', 25), ('4', 26), ('s', 27), ('B', 28), ('#', 29), ('r', 30), ('o', 31), ('\', 32), ('P', 33), ('I', 34), ('5', 35), ('i', 36), ('a', 37), ('K', 38), ('e', 39), ('Z', 40), ('L', 41), ('U', 42), ('Y', 43), ('6', 44), ('u', 45), ('R', 46), ('T', 47), ('M', 48), ('A', 49), ('g', 50), ('t', 51), ('b', 52), ('W', 53), ('d', 54), ('f', 55), ('V', 56), ('h', 57), ('7', 58), ('G', 59), ('p', 60), ('8', 61), ('m', 62), ('9', 63), ('E', 64), ('D', 65), ('%', 66), ('y', 67), ('0', 68), ('*', 69), ('X', 70), ('k', 71)]. The meaning of each character in SMILES can be found in the original literature[2]. The symbol <bos> and <eos> are used for indicating the start and end of a SMILES string, respectively. The symbol <unk> is reserved for characters that do not exist in the dictionary. The symbol <pad> is used for padding SMILES strings of length less than 100 (the input size of our model) to 100. The non-canonical SMILES and canonical SMILES share the same vocabulary in this work.

4 Supplementary Note4

Table S1. 18 ILs meeting all the four constraints

IL ID	IL_SMILES	CP	MP	TOX	T298KP1bar_Xco2	T328KP1bar_Xco2	viscosity	TD
3219685	CCCNCC[N+](CC)(CC)CC.N#C[N-]C#N	539.04	296.92	3.48	0.07	0.02	76.85	212.54
3252213	CCC[N+](CCN(CC)CC)(CC)CC.N#C[N-]C#N	630.81	292.92	3.41	0.04	0.00	94.40	263.32
3257267	COCC[N+](CC)(CC)CC.N#C[N-]C#N	487.29	285.92	3.14	0.05	0.01	48.70	169.71
3257270	COCC[N+](CC)(CC)CC.[O-]c1ccccc1	436.80	297.33	3.22	0.01	0.00	76.32	167.67
3257293	CCC[N+](CCOC)(CC)CC.FC(S(=O)(=O)[N-]S(=O)(=O)C(F)(F)F)(F)F	635.71	282.41	3.48	0.06	0.03	80.91	188.01
3257305	CCC[N+](CCOC)(CC)CC.N#C[N-]C#N	541.49	262.31	3.11	0.05	0.00	59.35	185.38
3257308	CCC[N+](CCOC)(CC)CC.[O-]c1ccccc1	495.02	284.66	3.26	0.01	0.00	72.26	174.41
3257343	CCCC[N+](CCOC)(CC)CC.N#C[N-]C#N	577.75	262.85	3.21	0.04	0.00	74.32	183.15
3257346	CCCC[N+](CCOC)(CC)CC.[O-]c1ccccc1	514.05	291.19	3.34	0.01	0.00	84.46	170.19
3257521	CCOCC[N+](CCOCC)(CC)CC.FC(S(=O)(=O)[N-]S(=O)(=O)C(F)(F)F)(F)F	681.91	284.12	3.42	0.07	0.03	83.68	165.99
3257533	CCOCC[N+](CCOCC)(CC)CC.N#C[N-]C#N	623.95	294.67	3.40	0.04	0.00	74.57	171.23
3258319	COCCOCC[N+](CCOCC)(CC)CC.FC(S(=O)(=O)[N-]S(=O)(=O)C(F)(F)F)(F)F	716.25	296.40	3.49	0.05	0.02	81.42	157.30
3258395	COCCOCCOCC[N+](CCOCC)(CC)CC.FC(S(=O)(=O)[N-]S(=O)(=O)C(F)(F)F)(F)F	809.71	283.77	3.31	0.07	0.03	92.49	196.47
3258433	CCOCC[N+](CC)(CC)CC.FC(S(=O)(=O)[N-]S(=O)(=O)C(F)(F)F)(F)F	646.90	268.09	3.08	0.06	0.03	77.67	180.13
3258445	CCOCC[N+](CC)(CC)CC.N#C[N-]C#N	521.51	284.80	3.40	0.04	0.00	52.46	201.99
3258471	CCOCC[N+](CCC)(CC)CC.FC(S(=O)(=O)[N-]S(=O)(=O)C(F)(F)F)(F)F	633.81	288.64	3.13	0.09	0.04	78.62	181.55
3259535	CCCOCC[N+](CCOCCOCCOC)(CC)CC.FC(S(=O)(=O)[N-]S(=O)(=O)C(F)(F)F)(F)F	803.36	275.87	3.21	0.11	0.05	86.40	191.64
3415058	CNCC[N+](CCN(C)C)(CCC)CC.[n-]1ccnn1	567.15	291.36	3.18	0.01	0.00	88.32	204.68

Reference

1. Vaswani, A. *et al.* Attention is All you Need. in *Advances in Neural Information Processing Systems* vol. 30 (Curran Associates, Inc., 2017).
2. Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Model.* **28**, 31–36 (1988).