

# Unimol+XGBoost 配方预测-初测结果

为了充分验证我们的领先性，我们和2024年新发表在arxiv的配方模型——MolSets进行对标。我们完全遵照MolSets对数据的拆分流程和测试方法进行比较。

## 对比MolSet

### 数据集的复现

MolSet论文中，用的数据基于ACS cenral 2023，但没有直接使用其中的整份数据集，而是“子集”；对温度的处理是，将分子组成相同但温度不同的entrees集合起来，根据阿累尼乌斯方程拟合成一个TARGET值。

### 拟合复现

代码如下：

```
1 # Arrhenius formula fitting and prediction functions
2 def fit_and_predict_conductivity(group_data, temp_to_predict):
3     # Transformations
4     group_data['1/T'] = 1 / group_data['temperature']
5     group_data['log_sigma'] = group_data['conductivity']
6
7     # Linear regression features and target
8     X = group_data[['1/T']]
9     y = group_data['log_sigma']
10
11     # Fit
12     model = LinearRegression()
13     model.fit(X, y)
14
15     # Predict
16     inverse_temp = 1 / temp_to_predict
17     predicted_log_sigma = model.predict([[inverse_temp]])
18     predicted_log_sigma_col = predicted_log_sigma[0]
19
20     return predicted_log_sigma_col
```

结果如下：

开始 插入 绘图 页面布局 >> 告诉我 批注 共享									
可能的数据丢失 如果将此工作簿以逗号分隔 (.csv) 格式保... 另存为...									
F248									
	A	B	C	D	E	F	G		
227	[Cu]C(=O)C	38000	[Li+].[O-][C]	0.29070246	0.290702463	-6.3759441			
228	[Cu]OC(=O)	65000	[Li+].[O-][C]	0.29070246	0.290702463	-6.0634104			
229	[Cu]SCCSCC	8894.4827	O=S(=O)([N]	0.29231605	0.292316045	-6.4217974			
230	[Cu]C(=O)CN(CCOCCOCC	O=S([N]-[S]	0.29350394	0.293503937	-5.8370101				
231	[Cu]P(OCCOCCOCC)(OCC	O=S(=O)([C]	0.29511703	0.29511703	-4.5800972				
232	[Cu]OCC[Au]	500	N#CcIn(C	0.29906924	0.29906924	-3.7871402			
233	[Cu]OCCCB(	2000	O=S(=O)([C]	0.29917726	0.299177264	-6.6570385			
234	[Cu]CC(CCS	56286000	O=S([N]-[S]	0.29928919	0.299289188	-11.210457			
235	[Cu]OCCOC	29500	O=S(=O)([N]	0.29	0.29	29500.0	-3.44		
236	[Cu]OCC[Au]	5000	O=S(=O)([N]	0.30268634	0.302686341	-3.2441251			
237	[Cu][B-]I(O	551.002974	O=ClO[B-]	0.30659804	0.306598041	-6.1396127			
238	[Cu]OCC[Au]	5000000	Fc1c(F)d(F)c	0.30716996	0.307169964	-10.179658			
239	[Cu]OC(=O)	19991	O=S(=O)([N]	0.30964546	0.309645456	-7.4109202			
240	[Cu]OCCOC	29500	O=S(=O)([N]	0.31	0.31	29500.0	-3.07		
241	[Cu]OC(F)(F	1819.72581	O=S(=O)([N]	0.32422467	0.324224671	-5.0259491			
242	[Cu][Si](C)(C)OCCOCCO	O=Cl(=O)([I]	0.32486011	0.324860112	-4.0920515				
243	[Cu]OC(=O)	299817.397	F[B-](F)(F)	0.32537957	0.325379567	-6.9477987			
244	[Cu]OC(=O)	299817.397	F[P-](F)(F)	0.32542854	0.325428543	-10.6916			
245	[Cu]OC(=O)	299817.397	F[Sb-](F)(F)	0.32639828	0.326398276	-6.9945459			
246	[Cu]OC(=O)	300000	O=S(=O)([C]	0.32650602	0.326506024	-9.6943733			
247	[Cu]OC(=O)	299817.397	F[Sb-](F)(F)	0.32650929	0.326509289	-9.0264563			
248	[Cu]SCCSCC	12552.7604	O=S(=O)([N]	0.32662139	0.326621387	-6.1491726			
249	[Cu]OC(=O)	299817.397	F[B-](F)(F)	0.32683906	0.326839064	-7.7039999			
250	[Cu]OC(=O)	299817.397	F[P-](F)(F)	0.32783818	0.327838182	-11.448065			
251	[Cu]OC(=O)	299817.397	F[Sb-](F)(F)	0.32856303	0.328563033	-7.5671964			
252	[Cu]CC(C)O	250000	O=S(=O)([N]	0.33217	0.33217_250	-5.0375999			
253	[Cu][Si](C)(C)OCCOCCO	O=Cl(=O)([I]	0.33436151	0.334361505	-4.1001795				
254	[Cu]OCCOCCOC(COCC=	O=S([N]-[S]	0.3345988	0.334598798	-4.5742805				
255	[Cu]SCCSCC	11624.7017	O=S(=O)([N]	0.33549248	0.335492479	-6.2049387			
256	[Cu]CC(C)[F	50000	O=S(=O)([C]	0.33736029	0.337360291	-11.551294			
257	[Cu]CN(C)C	10000	O=S([N]-[S]	0.33840948	0.338409475	-5.461216			
258	[Cu]C(OCCC=C)OC[Au]	O=S([N]-[S]	0.33865031	0.338650307	-5.0790751				
259	[Cu]OCC[Au]		F[P-](F)(F)	0.34090909	0.340909091	-6.2950052			
260	[Cu]C(=O)O	24390	O=S(=O)([N]	0.34468324	0.34468324	-6.0293657			
261	[Cu]C(=O)O	49151	O=S(=O)([N]	0.34468324	0.34468324	-6.9003887			
262	[Cu]OCCOCC	50000	O=S(=O)([N]	0.34468324	0.34468324	-6.1755783			
263	[Cu]C(Cl=C	51210	O=S(=O)([N]	0.34468324	0.34468324	-5.702972			
264	[Cu]C(Cl=C	55510	O=S(=O)([N]	0.34468324	0.34468324	-5.7114658			

存在一点点误差是因为此处用的273.15和298.15，换用273和298之后结果如下：

可以认为论文中对温度的处理具有科学性；有效性则需要进一步论证。

## 模型参数

```
1 xg_reg = xgb.XGBRegressor(
2     n_estimators=400, objective='reg:squarederror', colsample_bytree=0.3,
3     learning_rate=0.05, max_depth=6, alpha=3,
4     # tree_method='gpu_hist' if device.type == 'cuda' else 'auto'
5 )
```

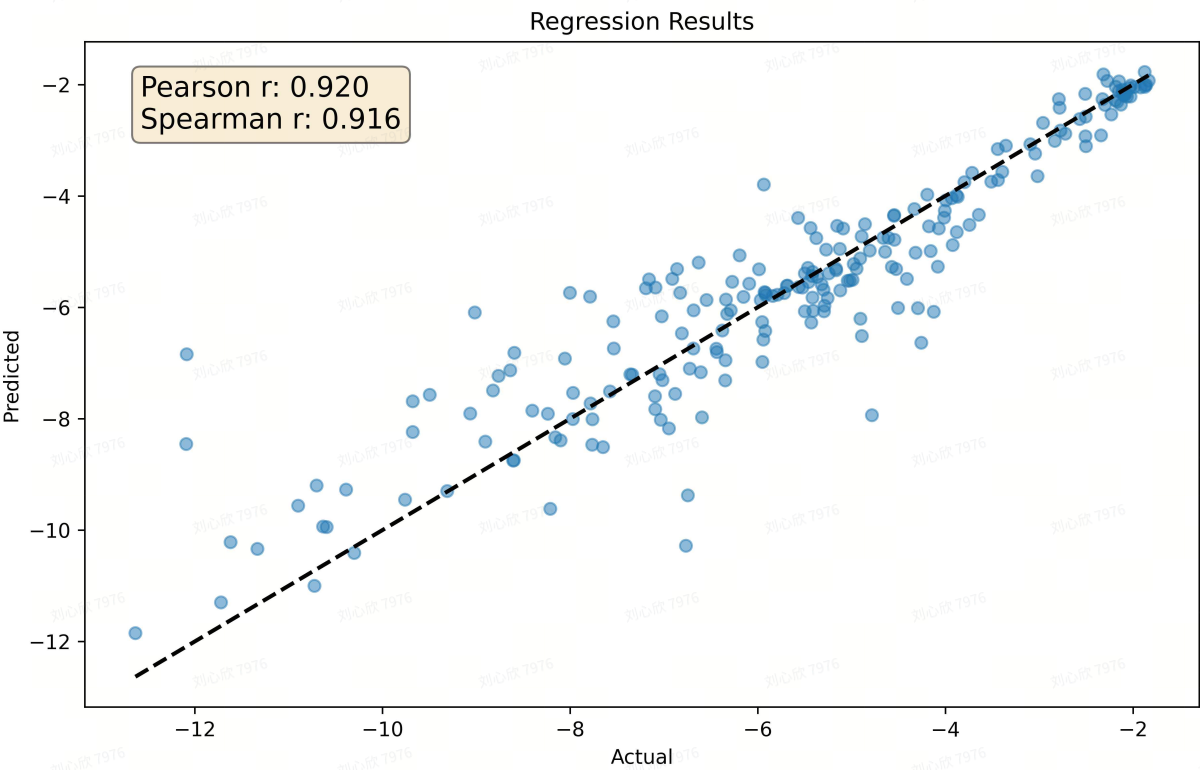
## 数据集还原

论文中没有直接给出SMILES，尽管数据基于ACS cenral 2023。还原数据集沿用了更早些QSPR论文里我们的格式：

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	SMILES_1	mw_1	SMILES_2	mw_2	SMILES_3	mw_3	SMILES_4	mw_4	SMILES_S1	molality_1	comp_type	mol_type	TARGET
2	CC1CCCCO	86.07316	O=C1OCC	88.01604	COC	90.12	CC1COC(=	102.0317	F[As-](F)(F)	1.378305	35	small	-1.81829
3	CC1CCCCO	86.07316	O=C1OCC	88.01604	COC	90.12	CC1COC(=	102.0317	F[As-](F)(F)	1.30887	35	small	-1.85078
4	O=C1OCC	88.01604	CC1CCCCO	86.07316	CC1COC(=	102.0317	COC	46.04186	F[As-](F)(F)	1.287324	36	small	-1.87347
5	CC1CCCCO	86.07316	O=C1OCC	88.01604	COC	90.12	CC1COC(=	102.0317	F[As-](F)(F)	1.296111	35	small	-1.89452
6	COC	46.04186	CC1CCCCO	86.07316	O=C1OCC	88.01604	CC1COC(=	102.0317	F[As-](F)(F)	1.527497	37	small	-2.02481
7	COC	46.04186	CC1CCCCO	86.07316	O=C1OCC	88.01604	CC1COC(=	102.0317	F[As-](F)(F)	1.429354	37	small	-1.92108
8	O=C1OCC	88.01604	CC1CCCCO	86.07316					F[As-](F)(F)	0.919927	38	small	-1.97028
9	CICCI	83.95336	CC1CCCCO	86.07316	O=C1OCC	88.01604	CC1COC(=	102.0317	F[As-](F)(F)	0.843996	39	small	-2.05351
10	CICCI	83.95336	CC1CCCCO	86.07316	O=C1OCC	88.01604	CC1COC(=	102.0317	F[As-](F)(F)	0.910995	39	small	-1.84209
11	CICCI	83.95336	CC1CCCCO	86.07316	O=C1OCC	88.01604	CC1COC(=	102.0317	F[As-](F)(F)	0.98955	39	small	-1.84806
12	CC1CCCCO	86.07316	O=C1OCC	88.01604	CC1COC(=	102.0317			F[As-](F)(F)	0.576945	40	small	-2.0281
13	O=C1OCC	88.01604	CC1CCCCO	86.07316	CC1COC(=	102.0317			F[As-](F)(F)	1.120814	41	small	-2.09901
14	CC1CCCCO	86.07316	O=C1OCC	88.01604	CC1COC(=	102.0317			F[As-](F)(F)	1.137112	40	small	-1.96305
15	CC1CCCCO	86.07316	O=C1OCC	88.01604	CC1COC(=	102.0317			F[As-](F)(F)	1.15389	40	small	-2.00208
16	CC1CCCCO	86.07316	O=C1OCC	88.01604	CC1COC(=	102.0317			F[As-](F)(F)	1.730836	40	small	-2.06168
17	CC1CCCCO	86.07316	O=C1OCC	88.01604	CC1COC(=	102.0317			F[As-](F)(F)	2.307781	40	small	-2.24182
18	CC1CCCCO	86.07316	O=C1OCC	88.01604	CC1COC(=	102.0317			F[As-](F)(F)	3.461671	40	small	-2.77267
19	CC1CCCCO	86.07316	C1CCOC1	72.05751	O=C1OCC	88.01604	CC1COC(=	102.0317	F[As-](F)(F)	1.133226	42	small	-1.87623
20	C1CCOC1	72.05751	CC1CCCCO	86.07316	O=C1OCC	88.01604	CC1COC(=	102.0317	F[As-](F)(F)	1.131017	43	small	-1.83686
21	C1CCOC1	72.05751	CC1CCCCO	86.07316	O=C1OCC	88.01604	CC1COC(=	102.0317	F[As-](F)(F)	1.128816	43	small	-1.81294
22	CC1CCCCO	86.07316	CC1=CC=	92.0626	O=C1OCC	88.01604	CC1COC(=	102.0317	F[As-](F)(F)	1.141828	44	small	-1.83618
23	CICCI	83.95336	CC1CCCCO	86.07316					F[As-](F)(F)	0.84959	45	small	-2.48464
24	CICCI	83.95336	CC1CCCCO	86.07316					F[As-](F)(F)	0.921916	45	small	-2.27469
25	CC1CCCCO	86.07316	CICCI	83.95336					F[As-](F)(F)	1.007702	46	small	-2.31562
26	O=S1(CCC	120.0245	CC1CCCCO	86.07316					F[As-](F)(F)	0.946565	47	small	-2.2867
27	O=S1(C(C	134.0402							F[As-](F)(F)	0.793665	48	small	-2.95932
28	O=S1(C(C	134.0402	COCCOCC	134.0943					F[As-](F)(F)	0.819689	49	small	-2.56866
29	O=S1(C(C	134.0402	CC1=CC=	92.0626					F[As-](F)(F)	0.846546	50	small	-2.66461
30	O=S1(C(C	134.0402	CC1=CC=	92.0626					F[As-](F)(F)	0.907021	50	small	-2.67902

预处理环节中我们发现了这篇论文处理数据集的一些问题 (详见“发现的问题”子标题), 但为了对比模型本身的学习能力, 本次preliminary测试中依然沿用了和论文完全一致的数据集。

## Regression图



### Pearson correlation:

Unimol+XGBoost: 0.91999

MolSet: 0.905

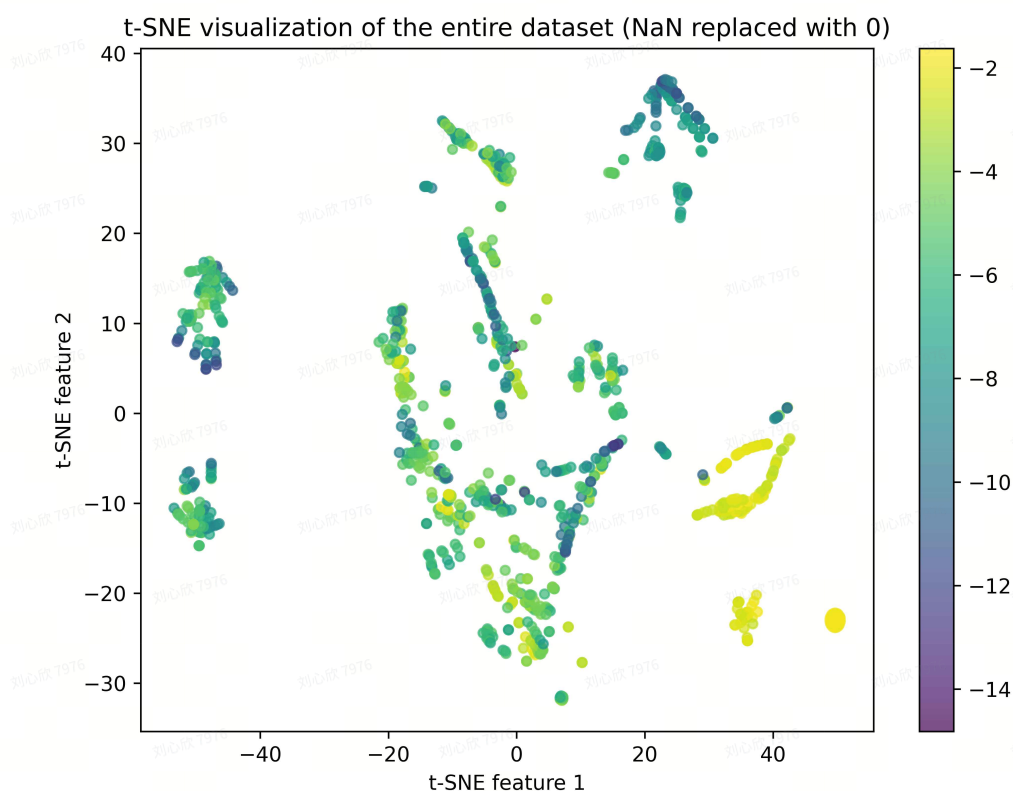
## Spearman r:

Unimol+XGBoost: 0.91638

MolSet: 0.907

用与论文完全一致的实验流程和输入，也超越了MolSet的表现。

## t-SNE图



特征太多，或许PCA/热图更直观展示模型学习到的信息。

## PCA图

### 发现的问题

MolSet作为已发表的论文有一些显而易见的问题，例如：

1. mw数据不准确。比如这里，solvent\_3的mw\_3在molset的pickle导出里是90.12

85	2414	86.07316494	CC1CCCO	86.07316	1.171522							F[As-](F)(F)
86	2422	86.07316494, 88.01604399, 90.12, 102.0316941	CC1CCCO	86.07316	O=C1OCC	1.378305	88.01604	COC	46.04186	CC1COC(=	102.0316941	F[As-](F)(F)
87	2433	86.07316494, 88.01604399, 90.12, 102.0316941	CC1CCCO	86.07316	O=C1OCC	1.30887	88.01604	COC	46.04186	CC1COC(=	102.0316941	F[As-](F)(F)
88	2444	88.01604399, 86.07316494, 102.0316941, 90.12	O=C1OCC	88.01604	CC1CCCO	1.287324	86.07316	CC1COC(=	102.0317	COC	46.04186481	F[As-](F)(F)
89	2456	86.07316494, 88.01604399, 90.12, 102.0316941	CC1CCCO	86.07316	O=C1OCC	1.296111	88.01604	COC	46.04186	CC1COC(=	102.0316941	F[As-](F)(F)
90	2464	90.12, 86.07316494, 88.01604399, 102.0316941	COC	46.04186	CC1CCCO	1.527497	86.07316	O=C1OCC	88.01604	CC1COC(=	102.0316941	F[As-](F)(F)

2. molset的pickle导出里molality是salt\_1对应的molality，把少数salt\_2无视了。

883	OCC[A	3997492		2.837684					O=S(=O)(N	2.837684		
884	OCC[A	3997492		2.270148					O=S(=O)(N	2.270148	O=S([N-]S(=	1.135074
885	OCC[A	3997492	C1COC(=C	1.135074	88.01604				O=S(=O)(N	1.135074		
886	OCC[A	3997492	CCOC(=O	1.135074	222.0892				O=S(=O)(N	1.135074		
887	OCC[A	3997492	CCOC(=O	1.135074	222.0892				O=S(=O)(N	1.135074	O=S([N-]S(=	0.567537
888	(=O)(	2258.659		0.512426					O=S(=O)(O	0.512426		
889	(=O)(	2258.659		0.960799					O=S(=O)(O	0.960799		

我认为只有1077条的数据集，这些问题不能忽视。

## 对比ACS central 2023

### 数据集复现

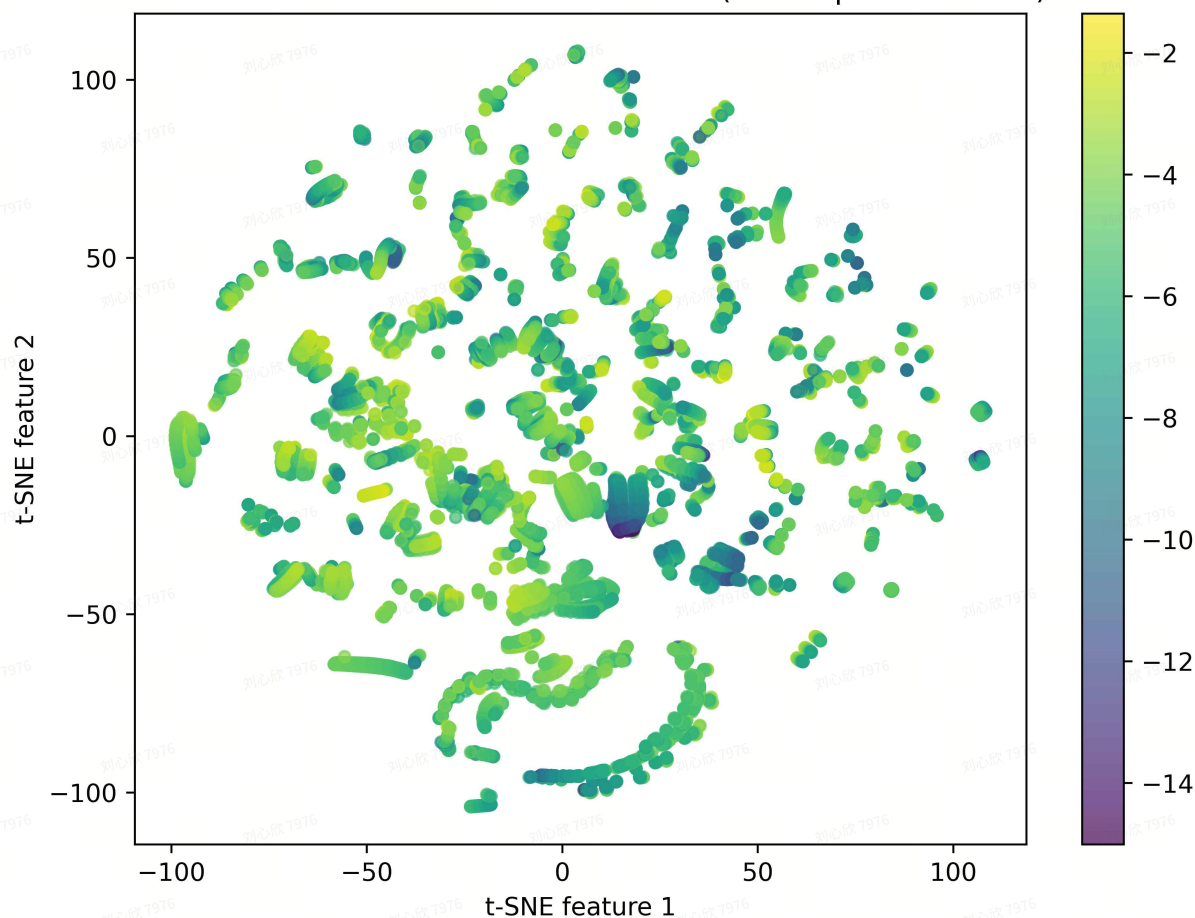
基本不需要特殊处理；该数据集共10000条；暂未发现明显缺陷。

### Regression图

### t-SNE



t-SNE visualization of the entire dataset (NaN replaced with 0)



## 发现的问题

原文的测试流程要求训练集与两个非训练集中不能有相同的polymer分子，在此基础上10-fold crossvalidation。虽然更好地表现了模型的泛化性能，但我认为这样选出的模型参数会有很大欠拟合风险。另外，一万条的数据集不算小数据集，不是很明白为什么要10-fold这么多。我认为多次随机单折、分层抽样更适合【specialist】模型。

## 总结

Unimol+XGBoost虽简单但有效。