使用稀疏

标注训练数据进行局域高浓度

电解质稀释剂筛选

Jonathan P. Mailoa,[1*] Xin Li,[1] Jiezhong Qiu,[1] 和 Shengyu Zhang[2*]

1) 腾讯量子实验室，腾讯，中国广东深圳 2) 腾讯量子实验室，腾讯，中国香港特别行政区

[*] 通讯作者：jpmailoa@alum.mit.edu，shengyzhang@tencent.com

使用稀疏标注训练数据进行多约束分子生成，以筛选局域高浓度电解质稀释剂

摘要

近年来，机器学习方法已被用于提出具有预期性质的分子，这对于高效探索广阔的化学空间尤为有用。然而，这些方法依赖完全标记的训练数据，因此在需要满足多项性质约束的分子时并不实用。对于所有这些性质，公开数据库通常无法提供足够的训练数据；尤其是在还需要从头算模拟或实验性质数据来训练条件分子生成模型时。在本研究中，我们展示了如何将一种仅适用于完全标记和完全未标记分子性质训练数据的半监督变分自编码器（SSVAE）模型改造为 ConGen 模型，使其也适用于标签稀疏分布的训练数据。我们使用整合多个公开分子性质数据库的数据集进行训练，评估 ConGen 在生成满足多项约束的分子方面的表现，并展示了一个应用示例：构建用于潜在锂离子电池局域高浓度电解液（LHCE）稀释剂的虚拟化学空间。

引言

在材料虚拟筛选工作中，条件分子生成能力是化学空间探索领域备受关注的课题。条件分子生成模型领域的研究要么在生成过程中不施加任何条件约束[1–5]，要么未能引入基于生成分子性质准确度的代价函数，导致模型生成分子的性质在很大范围内波动，远超出期望的属性范围。[6]这一难题源于在模型中，分子性质通常是以分子结构为输入、通过某种回归模型得到的输出。因此，难以将分子性质作为输入，用于对生成分子的化学空间施加条件约束。近期基于强化学习的研究实现了一种条件分子生成器：在假设训练期间能够持续使用分子性质评估器（化学信息学库或计算材料模拟工具）评估生成分子的前提下，经过数千次训练迭代后，该生成器便能生成优质的分子候选物。[7] 在本研究中，我们关注一项更常见于工业相关化学空间虚拟筛选的实际任务：给定来自多个来源、数量有限且往往不完整的分子性质训练标签，开发一种生成模型，生成满足多项性质约束的分子化学空间，以便将其作为高质量输入，以低成本且相对准确的方式用于虚拟筛选流程，而无需额外的模拟或实验来进一步优化生成模型。

近期的一些研究，例如 Kang 等人开发的半监督变分自编码器（SSVAE）模型[8,9]，通过采用双轨架构解决了这一问题的一部分：在该架构中，分子性质y 既是分子回归预测器子模型的输出，也是分子生成解码器子模型的输入，同时还是另一个独立的分子编码器子模型的输入。由于y 是预测器子模型的输出，即使它也被用作输入，仍可据此为整个模型构建有效的代价函数

以控制解码器生成的分子结构。由此得到的组合模型能够较好地控制生成分子的性质，因此适合高效生成满足特定条件约束的目标分子化学空间。此外，SSVAE 模型能够在训练过程中同时利用完全标记和完全未标记的分子，因此具有一定的实际应用吸引力；在许多情况下，由于缺乏模拟或实验数据，我们无法获得分子性质。然而，该模型在实际应用中仍不够实用，因为许多分子的数据只有部分标记，而 SSVAE 模型无法处理这类情况。电池电解质分子筛选就是一个实际例子：诸如分子量（分子量）和氟原子数（n_F）等“容易获取”的分子性质，可以通过化学信息学库轻松获得；而电离能（电离能）等模拟数据，以及黏度（对数黏度，即黏度的对数）等实验数据，则并不普遍可得。如果我们想生成满足多项此类约束的化学空间，那么公开数据库中的许多分子都无法用作 SSVAE 模型的完全标记训练数据。若彻底移除标签，将它们转为完全未标记的 SSVAE 训练数据，则会产生不利影响，因为这样会使我们失去训练数据集中大量宝贵的标签信息。

在本研究中，我们展示了如何让生成模型充分利用带有不完整标签的分子作为训练数据，而无需在训练期间额外请求训练数据标签。通过修改 SSVAE 模型，使其不再区分完全标记或完全未标记的分子，即可实现这一改进。现在，模型改为依靠分子性质掩码，指明给定分子中的哪些属性可用于训练，哪些不可用。我们将这一修改后的 SSVAE 方法命名为 ConGen 模型，下一节将概述实现这些实用功能所需的主要修改。当提供给模型的分子训练数据为完全标记或完全未标记时，ConGen 模型的数据工作流将与 SSVAE 模型的完全标记和完全未标记

数据工作流完全相同。然而，当提供给 ConGen 模型的训练数据是属性标签稀疏分布的分子时，其组件和代价函数会进行相应修改，使模型仅依据属性掩码使用相关属性标签。我们首先在原始 SSVAE 模型使用的训练数据集上，对该模型的用法进行基准测试；该数据集仅包含带标签和无标签分子。随后，我们展示了若干无法使用 SSVAE 模型实现的应用案例，包括为锂离子电池局部高浓度电解质稀释剂（LHCE）候选物生成用于虚拟筛选的化学空间。为此，我们合并了五个公开可用的分子属性数据库，其中包含分子量、氟原子数和氧原子数（*Mol.Wt*、n_F 和n_O）、电离能和电子亲和能（*IE* 和 *EA*），以及*Log.Vis*等不同属性。这些属性的可用程度差异很大：前三种属性完全可用（“容易”），接下来两种属性的可用率约为 3%（“中等”属性，可通过量子化学模拟获得），最后一种属性的可用率约为 0.03%（“困难”属性，可通过实验测量获得）。

基线 SSVAE 模型

我们首先介绍 S. Kang等人,[8]开发的基线 SSVAE 模型的内部工作原理，该模型构成了本研究的基础。SSVAE 模型的主要思想很简单：

1. 使用编码器子模型，将训练数据集中的输入分子结构 x 编码为潜在空间表示 $\boldsymbol{z}$。

2. 使用预测器子模型，根据训练数据集中的输入分子结构 x 预测属性标签 $\boldsymbol{y_P}$。如果训练数据库中存在实际的分子属性标签 $\boldsymbol{y_L}$，则丢弃 y_P，并使用模型内部的分子属性标签 $\boldsymbol{y = y_L}$。否则，使用 $\boldsymbol{y = y_P}$。

3. 将模型内部的分子属性标签 $\boldsymbol{y}$ 和潜在空间表示 $\boldsymbol{z}$ 作为解码器子模型的输入，以生成输出分子结构 $\boldsymbol{x_D}$。

为了处理不同类型的训练数据（有标签与无标签），SSVAE 模型会区别对待这两类数据。每个训练轮次的小批量中的训练数据集会分成两个不同的小批量（有标签与无标签）。随后，SSVAE 工作流程会运行两次，具体方式略有不同，取决于分子小批量是完全有标签还是完全无标签（**图 1**）。

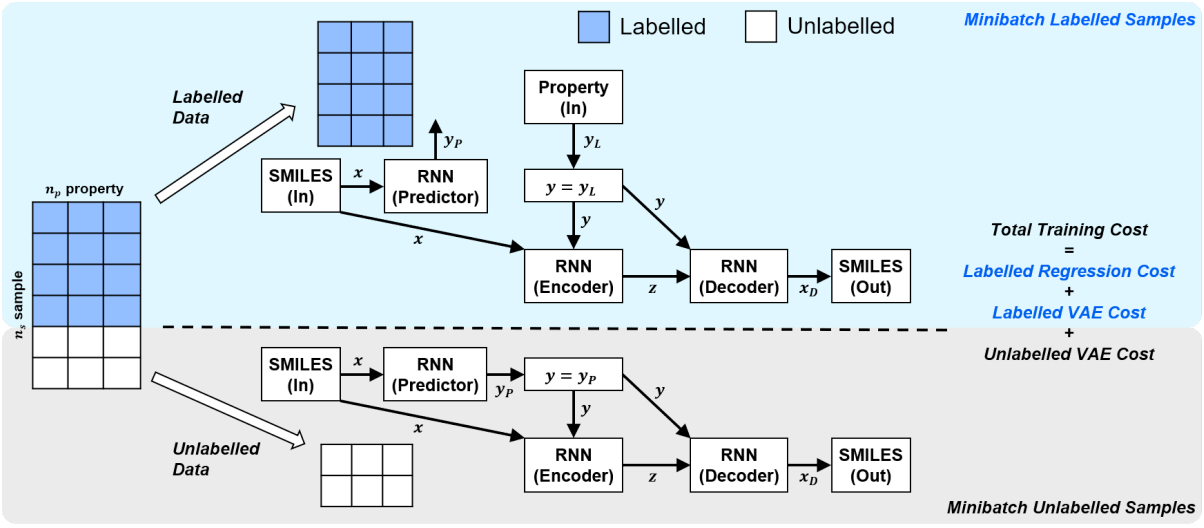

图 1 | Kang 等人原始 SSVAE 模型中有标签/无标签数据与模型的高层级区分。[8] 变分自编码器（VAE）代价分别针对无标签数据集和有标签数据集计算，而回归代价仅针对有标签数据集计算。随后将这三项代价相加，得到小批次训练代价。

在 SSVAE 方法中，一个分子条目的训练代价函数需要拆分为三部分（**公式1-3**）。为清晰起见，下面详细展开代价函数；接下来我们需要在下一节说明，如何针对 ConGen 模型中的脏数据（部分标记数据）进行修改：

a. 小批量中完全标记条目的 VAE成本函数（**公式 1**）：

$$\boldsymbol{\mathcal{L}}(\boldsymbol{x},\boldsymbol{y}) = -\sum_{i=1}^{n_L}\sum_{j=1}^{n_x}\left(x_{i,j}\ln x_{D,i,j} + \left(1-x_{i,j}\right)\ln\left(1-x_{D,i,j}\right)\right)$$

$$+\sum_{i=1}^{n_L}\frac{1}{2}\left(n_y \ln 2\pi + \ln\left(det(\boldsymbol{C})\right) + \sum_{j=1}^{n_y}\left(y_{L,i,j}-\boldsymbol{E}_j\right)\sum_{k=1}^{n_y}\left(y_{L,i,k}-\boldsymbol{E}_k\right)\boldsymbol{C}^{-1}_{k,j}\right)$$

$$-\sum_{i=1}^{n_L}\sum_{j=1}^{n_z}\frac{1}{2}\left(1+\ln\sigma\left(z_{i,j}\right)^2-\mu\left(z_{i,j}\right)^2-\sigma\left(z_{i,j}\right)^2\right)$$

b. 小批量中完全无标签条目的 VAE成本函数（**公式 2**）：

$$\boldsymbol{\mathcal{U}}(\boldsymbol{x}) = -\sum_{i=1}^{n_L}\sum_{j=1}^{n_x}\left(x_{i,j}\ln x_{D,i,j} + \left(1-x_{i,j}\right)\ln\left(1-x_{D,i,j}\right)\right)$$

$$+\sum_{i=1}^{n_L}\frac{1}{2}\left(\sum_{j=1}^{n_y}\boldsymbol{C}^{-1}_{j,j}\sigma\left(y_{P,i,j}\right)^2 + \sum_{j=1}^{n_y}\left(y_{P,i,j}-\boldsymbol{E}_j\right)\sum_{k=1}^{n_y}\left(y_{P,i,k}-\boldsymbol{E}_k\right)\boldsymbol{C}^{-1}_{k,j} - n_y + \ln\left(det(\boldsymbol{C})\right) - \sum_{j=1}^{n_y}\ln\sigma\left(y_{P,i,j}\right)^2\right)$$

$$-\sum_{i=1}^{n_L}\sum_{j=1}^{n_z}\frac{1}{2}\left(1+\ln\sigma\left(z_{i,j}\right)^2-\mu\left(z_{i,j}\right)^2-\sigma\left(z_{i,j}\right)^2\right)$$

c. 完全标记条目的回归代价函数（**公式 3**）：

$$\boldsymbol{\mathcal{R}}_{\boldsymbol{SSVAE}}(\boldsymbol{x},\boldsymbol{y}) = \beta\sum_{i=1}^{n_L}\sum_{j=1}^{n_y}\left(y_{L,i,j}-\mu\left(y_{P,i,j}\right)\right)^2$$

其中，$\boldsymbol{C} = Cov\left(\boldsymbol{y_L}\right)$ 和 $\boldsymbol{E} = E\left(\boldsymbol{y_L}\right)$ 是根据整个完全标记的训练集构建的标签协方差矩阵和均值，μ 是均值函数，σ 是标准差函数，β 是生成学习与监督学习之间的权衡超参数，而n_L、n_U、 n_x、 n_y和n_z 分别表示小批量中完全标记条目、完全未标记条目的数量，以及$\boldsymbol{x}$、 $\boldsymbol{y}$和$\boldsymbol{z}$ 的维度。最后，小批次代价函数总和就是$Cost_{SSVAE} = \boldsymbol{\mathcal{L}} + \boldsymbol{\mathcal{U}} + \boldsymbol{\mathcal{R}}_{\boldsymbol{SSVAE}}$。

最后，训练完成后，可以提取解码器子模型并独立运行，只需指定有条件属性输入y 以及随机采样的潜在空间输入$\boldsymbol{z}$ ，即可有条件地生成所需的分子输出；其中，使用束搜索算法将输出$\boldsymbol{x_D}$ 转换为分子 SMILES。这种方法的主要缺点是，训练数据集必须是完全标记或完全未标记的。SSVAE 模型按照**图 1**上文所述的方式拆分问题，是因为这样可以大幅简化模型的数据流、数学运算和行为。在实际应用中，所关注的训练数据集可能包含标签不完整的分子，以及完全标记或未标记的分子。如果训练分子数据库取自公开数据库（例如 PubChem 实验数据[10]），或由多个数据库合并而成，情况尤其如此。这两类实际中常见的"脏"数据集都无法用于训练基线 SSVAE 模型，因此严重限制了可实现的条件分子生成类型，尤其是在需要多属性条件分子生成时。电池电解液或药物分子的虚拟筛选通常就属于这种情况。

使用 ConGen 模型实现稀疏标注数据的利用

我们将 SSVAE 模型改进为 ConGen 模型。该模型专为处理“脏”训练数据而设计，因此能够利用数量大得多的训练数据来源，包括合并自不同公共和私人来源的数据。 这使我们能够执行此前无法使用 SSVAE 模型完成的有条件分子生成任务。例如，给定一个来自 ZINC 的大型带标签分子数据集[11] （包含 分子量、疏水性 辛醇-水分配系数和类药性 定量估计药物相似性）以及另一个来自 Materials Project Electrolyte Genome 的同样大型分子数据集[12] （包含 分子量、电子亲和能和电离势），我们可以训练一个条件生成模型，在分子量、辛醇 - 水分配系数和电子亲和能值上同时满足多项约束，生成具有这些属性的分子（这些属性已知有助于筛选锂电池电解质）。鉴于训练数据来源多样，原始 SSVAE 模型无法在合并后的数据库上进行训练，因为该数据库包含分子量、辛醇-水分配系数和电子亲和能标签，而训练数据标签十分稀疏。相比之下，ConGen 没有这类限制，用户可以根据需要混合来自多个来源的非理想实用数据。

ConGen模型的核心理念是采用SSVAE模型的通用高层架构，并根据需要修改其所有组件，以便能够使用含噪训练数据。我们已将整个SSVAE模型从原始的TensorFlow 1.0版本重写为PyTorch版本，以提高模型的灵活性，随后进一步实施必要的修改，使其能够使用稀疏训练数据标签。使用相同的超参数训练设置（n_{trn} = 285k 个训练分子，带标签/无标签分子按 50:50 划分，n_{val} = 15k 个验证分子，n_{tst} = 10k 个测试分子，β = 10^4，Adam 优化器学习率 $LR = 10^{-4}$），在原始 SSVAE 训练数据（仅包含完全标记和完全未标记的分子）上训练此 PyTorch 版本后，我们得到的属性预测、无条件分子生成和单属性条件分子生成任务的准确度指标（仅使用 分子量 = 250 道尔顿约束，因为原始 SSVAE 代码仅支持单属性约束）与 TensorFlow 版本相当（**表 1**）。在无条件和条件生成任务中，均生成 100 个分子。

Task	属性	SSVAE	ConGen
预测器回归 *MAE*	分子量（道尔顿）	0.95	1.22
	LogP	0.06	0.08
	QED	0.013	0.014
解码器无条件生成	分子量（道尔顿）	360 ± 65	363 ± 64
	LogP	2.95 ± 1.06	3.01 ± 1.07
	QED	0.723 ± 0.142	0.713 ± 0.154
解码器有条件生成	分子量（道尔顿）	249 ± 6	251 ± 5
	LogP	2.38 ± 0.89	2.13 ± 0.91
	QED	0.810 ± 0.072	0.816 ± 0.095

表 1 | SSVAE（TensorFlow 1.0）与基线 ConGen（PyTorch）模型在原始 SSVAE 模型任务上的比较。基线 ConGen 与 SSVAE 等效，但其使用 PyTorch 实现。此比较基于 SSVAE 的"清理后"原始训练数据集进行，该数据集仅包含完全标记和完全未标记的分子。采用相同的训练超参数设置，并获得了相对相当的性能指标。细微差异可归因于原始模型训练超参数设置较为激进。对于性质预测任务，使用预测器子模型计算相对于训练标签的平均绝对误差（MAE）。对于无条件／有条件生成任务，使用解码器子模型生成分子，并使用 RDKit 化学信息学库计算分子的属性标签。

确认两个模型等效后，我们对 ConGen 子模型中的输入数据预处理和分子数据工作流进行了修改（**图 2**）。首先，我们使模型能够将带有不同类型属性标签的分子训练数据标签合并到一个新的性质标签矩阵 $\boldsymbol{y_L}$ 中。这会导致合并后的数据库中有相当一部分缺少[分子、属性]条目标签。对于所有数据库中均没有可用标签的条目，我们将其属性标签标记为无效。为此，可以生成一个掩码矩阵 $\boldsymbol{M}$，其中无效条目为"0"，具有可用性质值的条目为"1"。对于不同数据库中均有多个可用属性标签的条目，我们选择合并过程中最后一个数据库中的可用标签。现在，$\boldsymbol{y_L}$ 和 $\boldsymbol{M}$ 矩阵都必须作为输入提供给 ConGen 模型。ConGen 不再根据分子是完全标记还是完全未标记来区分数据工作流。相反，ConGen 会针对中间标签 $\boldsymbol{y}$ 实施一个选择器，根据掩码 $\boldsymbol{M}$ 的值，选择使用现有标签 $\boldsymbol{y_L}$，或使用预测器子模型生成的预测属性标签 $\boldsymbol{y_P}$（**公式 4**）：

$$\boldsymbol{y}(i,j) = \begin{cases} \boldsymbol{y_L}(i,j) & if \;\; \boldsymbol{M}(i,j) == 1 \\ \boldsymbol{y_P}(i,j) & if \;\; \boldsymbol{M}(i,j) == 0 \end{cases}$$

其中，i 和j 分别表示分子和属性类型索引。通过这一修改，统一数据工作流可用于完全标记、完全未标记和部分未标记的分子。此外，当小批量中的分子为完全标记或完全未标记时，ConGen模型对其执行的数学运算将与SSVAE 模型中的运算完全相同。

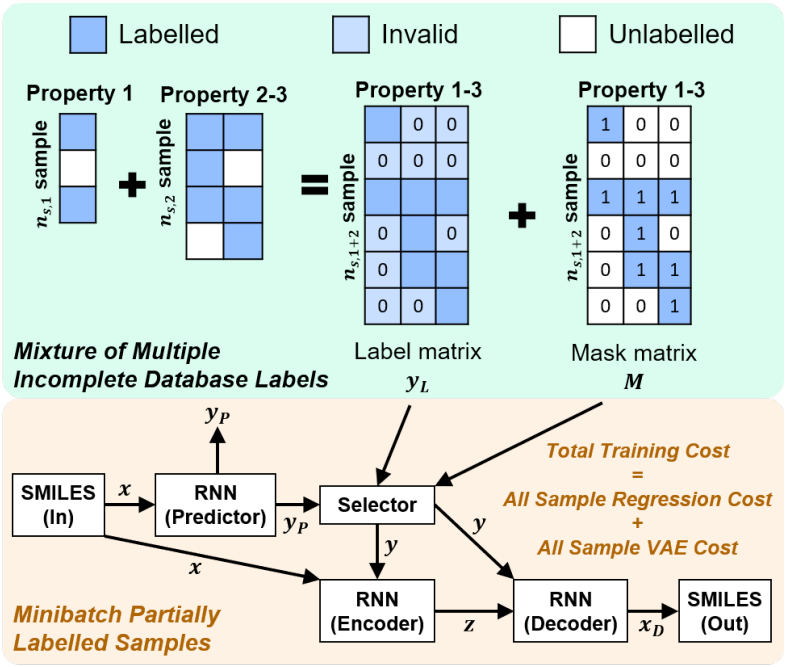

图 2 | ConGen模型中的脏训练标签数据合并与高层脏数据工作流。 ConGen模型不再区分完全标记、完全未标记和部分标记的分子输入。统一数据工作流由掩码矩阵$\boldsymbol{M}$控制。$n_{s,\ 1}$、$n_{s,\ 2}$和$n_{s,\ 1+2}$ 分别表示第一个、第二个以及合并后的属性数据库中的样本数量。

然而，对于训练代价函数及后续的分子生成，情况并没有那么简单。需要注意的是，SSVAE 模型中训练代价函数的实现很大程度上取决于分子是完全标记还是完全未标记。SSVAE 代价函数由三个主要部分组成，旨在确保预测器、编码器和解码器都准确无误（**公式 1-3**）；我们需要设计脏数据 VAE 代价函数，以替代 $\mathcal{L}$ 和 $\mathcal{U}$，因为我们不再拥有完全标记和完全未标记的分子。值得注意的是，在原始 SSVAE 模型运行期间，小批量内的分子输入之间不会相互影响（例如，如果分子 A 和 B 同时处理，模型对两者的输出

不会因另一分子也在同时处理而受到影响。这确保了某个分子的任何中间值（$\boldsymbol{y_L}$、 $\boldsymbol{y_P}$、 $\boldsymbol{z}$、 $\boldsymbol{x_D}$ 等）都仅由该分子输入 $\boldsymbol{x}$ 决定。因此，为 ConGen 模型实现新的代价函数就不那么复杂了。$\boldsymbol{\mathcal{L}}$ 和 $\boldsymbol{\mathcal{U}}$之间有大量重叠项，因此我们可以为 ConGen 模型设计新的 VAE 代价函数 $\boldsymbol{\mathcal{G}}$，利用掩码矩阵 $\boldsymbol{M}$ 处理部分有标签条目（**公式 5**）。当所有条目都完全标记时，$\boldsymbol{M}$ 的值都将为 1，而 $\boldsymbol{\mathcal{G}}$ 应转换为 $\boldsymbol{\mathcal{L}}$，但不包括一些不会影响训练的常数项。当所有条目都完全未标记时，$\boldsymbol{M}$ 的值都将为 0，而 $\boldsymbol{\mathcal{G}}$ 应转换为 $\boldsymbol{\mathcal{U}}$，同样不包括一些常数项。类似地，我们的新回归代价函数 $\boldsymbol{\mathcal{R}_{ConGen}}$ 应仅对小批量中的已标记条目求和。通过确保这种行为，在提供完全标记/未标记数据时，后续的 ConGen 成本函数微分和模型参数优化将与 SSVAE 版本完全相同。不过，它现在也适用于含噪稀疏标注训练数据。下文将定义适用于 ConGen 小批量的新代价函数，尤其是针对脏数据的代价函数：

a. 小批量中含噪已标记条目的 VAE成本函数（**公式 5**）：

$$\boldsymbol{\mathcal{G}}(\boldsymbol{x},\boldsymbol{y}) = -\sum_{i=1}^{n_S}\sum_{j=1}^{n_x}\left(x_{i,j}\ln x_{D,i,j} + \left(1-x_{i,j}\right)\ln\left(1-x_{D,i,j}\right)\right)$$

$$+\sum_{i=1}^{n_S}\frac{1}{2}\left(n_y\ln 2\pi + \sum_{j=1}^{n_y}\boldsymbol{M}_{i,j}\left(y_{i,j}-\boldsymbol{E}_j\right)\sum_{k=1}^{n_y}\boldsymbol{M}_{i,k}\left(y_{i,k}-\boldsymbol{E}_k\right)\boldsymbol{C}_{k,j}^{-1}\right)$$

$$+\sum_{i=1}^{n_S}\frac{1}{2}\left(\sum_{j=1}^{n_y}\boldsymbol{C}_{j,j}^{-1}\left(1-\boldsymbol{M}_{i,j}\right)\sigma\left(y_{i,j}\right)^2 + \sum_{j=1}^{n_y}\left(1-\boldsymbol{M}_{i,j}\right)\left(y_{i,j}-\boldsymbol{E}_j\right)\sum_{k=1}^{n_y}\left(1-\boldsymbol{M}_{i,k}\right)\left(y_{i,k}-\boldsymbol{E}_k\right)\boldsymbol{C}_{k,j}^{-1} - n_y - \sum_{j=1}^{n_y}\left(1-\boldsymbol{M}_{i,j}\right)\ln\sigma\left(y_{i,j}\right)^2\right)$$

$$-\sum_{i=1}^{n_S}\sum_{j=1}^{n_z}\frac{1}{2}\left(1+\ln\sigma\left(z_{i,j}\right)^2 - \mu\left(z_{i,j}\right)^2 - \sigma\left(z_{i,j}\right)^2\right)$$

b. 小批量中脏数据已标记条目的回归代价函数（**公式6**）：

$$\boldsymbol{\mathcal{R}_{ConGen}}(\boldsymbol{x},\boldsymbol{y}) = \beta\sum_{i=1}^{n_S}\sum_{j=1}^{n_y}\boldsymbol{M}_{i,j}\left(y_{i,j}-\mu\left(y_{P,i,j}\right)\right)^2$$

其中，n_S 表示脏数据小批次中的所有样本数。很容易证明，在此方案下，$\boldsymbol{\mathcal{G}}$ 会根据$\boldsymbol{M}$的取值转换为$\boldsymbol{\mathcal{L}}$ 或$\boldsymbol{\mathcal{U}}$，但常数项除外，因为它们不会对模型参数优化过程产生任何影响。请注意，与 SSVAE 代价函数相比，出于数值方面的考虑，我们有意从上述代价函数中移除了常数项$\ln\big(det(\boldsymbol{C})\big)$，相关原因将在下一段关于脏数据协方差矩阵 $\boldsymbol{C}$的内容中说明。至关重要的是，在这个新的代价函数中，只有$\boldsymbol{\mathcal{L}}$ 和$\boldsymbol{\mathcal{U}}$ 中对应的已标记/未标记矩阵元素会参与对n_s 和 n_y 的求和，从而形成$\boldsymbol{\mathcal{G}}$。现在，小批次代价函数总值仅为 $Cost_{ConGen} = \boldsymbol{\mathcal{G}} +$

$\boldsymbol{\mathcal{R}_{ConGen}}$。

需要注意的是，由于我们只有部分有标签条目，因此没有完整的 $\boldsymbol{y_L}$，相应地，$\boldsymbol{C} = Cov$（$\boldsymbol{y_L}$） 和 $\boldsymbol{E} = E$（$\boldsymbol{y_L}$） 只能利用不完整条目进行计算，这使得这些矩阵的定义不明确，尤其是 Cov（$\boldsymbol{y_L}$）。对于 SSVAE 模型，$\boldsymbol{C}$ 是定义明确的，因为我们可以直接将无标签分子条目从训练集中完全剔除，并根据带有完整标签的分子条目直接计算 $\boldsymbol{C}$ 和 $\boldsymbol{E}$（只要存在大量能够良好代表完整训练数据集的完全标记分子，这就是一种很好的近似）。这一步可以在模型构建期间执行一次，并在整个模型训练过程中始终将其设为这些值。然而，对于 ConGen 而言，这种策略并不适用，因为训练数据是不干净的。在这种情况下，更合理的做法是根据有效条目计算标签均值 $\boldsymbol{E}$，并忽略 y_L 矩阵中的无效值。同样，更合理的做法是根据可用的 y_L 矩阵条目计算协方差矩阵 $\boldsymbol{C}$ 条目，同时忽略无效条目。换言之，对于 $\boldsymbol{E}$ 和 $\boldsymbol{C}$ 的计算，我们有如下情况（**公式 7-8**）：

$$\boldsymbol{E}_j = E(\boldsymbol{y_L})_j = \frac{\sum_{i=1}^{n_s} \boldsymbol{y}_{\boldsymbol{L},i,j} \boldsymbol{M}_{i,j}}{\sum_{i=1}^{n_s} \boldsymbol{M}_{i,j}}$$

$$\boldsymbol{C}_{j,k} = Cov(\boldsymbol{y_L})_{j,k} = \frac{\sum_{i=1}^{n_s} \big(\boldsymbol{y}_{\boldsymbol{L},i,j} - \boldsymbol{E}_j\big)\big(\boldsymbol{y}_{\boldsymbol{L},i,k} - \boldsymbol{E}_k\big)\boldsymbol{M}_{i,j}\boldsymbol{M}_{i,k}}{\big(\sum_{i=1}^{n_s} \boldsymbol{M}_{i,j}\boldsymbol{M}_{i,k}\big) - 1}$$

在 SSVAE 模型所使用的这类干净训练数据中，掩码矩阵 $\boldsymbol{M}$的所有条目均为 1，因此可以从数学上证明，协方差矩阵 $\boldsymbol{C}$ 始终是半正定（PSD）矩阵。相应地，在 SSVAE 中，代价函数中上述的对数行列式项 $\ln(det(\boldsymbol{C}))$始终有明确定义。然而，当掩码矩阵 $\boldsymbol{M}$ 的条目不再全部为 1 时，这一数学保证便不再成立。[13] 因此，尝试对负数进行对数运算可能会导致训练错误。不过，由于项$\ln(det(\boldsymbol{C}))$只是一个常数，我们可以将其从 ConGen 代价函数中移除，而不会对数学上的训练产生任何影响，正如我们在**公式 5**中所做的那样。

真正的物理问题源于$\boldsymbol{E}$ 和$\boldsymbol{C}$ 本身的质量。当训练数据标签的可用性较低（掩码矩阵 $\boldsymbol{M}$ 中有大量 0 项）时，就会出现严重问题，因为矩阵 $\boldsymbol{E}$ 和 $\boldsymbol{C}$ 无法准确表示真实的分子属性标签，尤其是在训练数据集中存在许多无效标签的情况下。在训练迭代过程中始终保持 $\boldsymbol{E}$ 和 $\boldsymbol{C}$ 的值不变，意味着后续模型训练和条件生成过程完成后，我们将无法很好地控制条件生成分子的属性。我们可以使用插补技术[13] 来缓解这一问题：当$\boldsymbol{y_L}$中没有有效标签时，使用预测器子模型预测的分子属性标签重新计算$\boldsymbol{E}$ 和$\boldsymbol{C}$。换句话说，我们会在一个训练轮次内跟踪来自选择器的小批量$\boldsymbol{y}$ （**公式 4**、**图 2**），并在该轮次中的每个训练和验证周期完成后，使用$\boldsymbol{y}$ 而非$\boldsymbol{y_L}$ 重新计算$\boldsymbol{E}$ 和$\boldsymbol{C}$ 。此更新会在整个训练过程中迭代执行。将最终的$\boldsymbol{E}$ 和$\boldsymbol{C}$ 作为 ConGen 模型参数的一部分加以保存十分重要，因为后续的分子生成任务需要使用质量更高的$\boldsymbol{E}$ 和$\boldsymbol{C}$ 参数来执行（**公式 8–9**）：

$$\boldsymbol{E}_j = E(\boldsymbol{y})_j = \frac{\sum_{i=1}^{n_a} \boldsymbol{y}_{i,j}}{n_a - 1}$$

$$\boldsymbol{C}_{j,k} = Cov(\boldsymbol{y})_{j,k} = \frac{\sum_{i=1}^{n_a} (\boldsymbol{y}_{i,j} - \boldsymbol{E}_j)(\boldsymbol{y}_{i,k} - \boldsymbol{E}_k)}{n_a - 1}$$

其中，n_a 是训练数据集中所有分子的数量。在训练初期，$\boldsymbol{E}$ 和$\boldsymbol{C}$ 的质量并不理想。然而，随着预测器子模型在后续训练迭代中变得更加准确，$\boldsymbol{E}$ 和$\boldsymbol{C}$ 将能更好地代表真实的样本总体，因此最终可以获得更准确的分子属性预测和条件生成结果。

我们还利用 ConGen 模型的模块化特性（继承自 SSVAE 的模块化设计），进一步提升模型在训练属性标签稀缺的数据集（例如从头算模拟或实验属性数据集）上的性能。在 ConGen 中实现迁移学习十分简单：只需将预测器和编码器子模型中的循环神经网络（RNN）替换为在规模大得多（但成本较低）的分子属性数据集上预训练的双向变换器编码器表示（BERT）模型即可。这里我们使用 ChemBERTa 模型，这是一种大规模的自监督的基于变换器的预训练模型，仅需输入分子 SMILES，并且已得到全面评估。[14] 在构建子模型时，我们在迁移后的 ChemBERTa 模型之上添加一个全连接网络线性层。此后，我们将这种迁移模型简称为“BERT”。使用 BERT 替换 RNN 编码器时，ChemBERTa 的所有层参数都会被冻结。然而，使用 BERT 替换 RNN 预测器时，最后一个 ChemBERTa 层的参数可以通过 PyTorch 优化器进行微调。虽然我们没有用其他类型的解码器子模型替换 RNN 解码器，但原则上如有需要，实现这一替换也很简单。对于仅使用 RNN 子模型的标准 ConGen 模型训练，我们设置 Adam 优化器 $LR = 10^{-4}$ ，并将梯度的绝对值裁剪至最大 10^2。对于使用 BERT 预测器替换子模型和解码器替换子模型进行的 ConGen 模型训练，我们为基于 BERT 的子模型使用显著较低的 Adam 优化器 $LR = 3 \times 10^{-5}$ ，并使用 $LR = 10^{-3}$ 来优化解码器子模型参数。

最后，我们在**表2**中展示了 ConGen 模型在脏数据集上的表现。训练数据标签混合自两个不同的数据库：1）ZINC数据库，包含分子量、辛醇-水分配系数和定量估计药物相似性[11]等属性，这些属性曾用于 SSVAE 论文中；[8] 2）Materials Project

电解质基因组数据库，包含电离能和电子亲和能等属性。[12] ConGen模型在这5种属性上进行训练，而 SSVAE 模型无法做到这一点。作为多属性条件生成的示例，我们向模型提出查询，要求生成同时具有3种属性的分子：分子量 = 250 道尔顿、辛醇-水分配系数 = 2.5 和电离能 = 5 电子伏特。相应的回归和条件生成结果见下文**表2**。我们使用 RDKit[15] （用于验证分子量和辛醇-水分配系数）以及量子化学方法（用于验证电离能，见**方法**）验证生成分子的属性。总体来看，基于 BERT 的 ConGen 在属性预测任务上的表现不如基于 RNN 的 ConGen 模型，但在条件生成任务上的表现与基于 RNN 的 ConGen 相当（在分子量和辛醇-水分配系数方面表现良好，但对电离能的预测准确度较低）。我们原本预期，迁移得到的基于 BERT 的 ConGen 在分子量和辛醇-水分配系数等常见属性标签上的表现不如基于 RNN 的 ConGen，而在电离能等稀有属性标签上的表现更好。基于 RNN 和 BERT 的 ConGen 在分子条件生成任务中表现相当，这一点值得今后进一步研究。我们推测，仅从 Materials Project Electrolyte Genome 数据库获取的量子化学属性训练标签数量仍然不足[12] ，而且仍需要更准确、数据效率更高的预测器子模型。目前，基于 BERT 的 ConGen 计算成本更高，却未能显著优于基于 RNN 的 ConGen，因此在接下来的大规模电解质稀释剂筛选部分，我们将仅使用基于 RNN 的 ConGen。

Task	模型	分子量（道尔顿）	*LogP*	*QED*	电子亲和能（电子伏特）	电离能（电子伏特）
预测器回归 测试集 平均绝对误差	**RNN**	2.70	0.05	0.009	0.20	0.16
	BERT	6.07	0.15	0.017	0.22	0.19
解码器无条件生成	**RNN**	312 ± 95	2.07 ± 1.28	0.677 ± 0.124	1.79 ± 0.84	5.99 ± 0.44
生成	**BERT**	271 ± 145	2.15 ± 1.11	0.583 ± 0.138	1.72 ± 0.82	6.40 ± 0.34
解码器有条件	**RNN**	248 ± 4	2.55 ± 0.23	0.672 ± 0.082	2.06 ± 0.55	6.53 ± 0.62
生成	**BERT**	252 ± 3	2.45 ± 0.36	0.756 ± 0.127	1.80 ± 0.64	6.36 ± 0.41

表 2 | ConGen模型在"脏数据"任务上的性能比较，包括基于 RNN 的 ConGen 和基于 BERT 的 ConGen。 原始 SSVAE 模型无法处理这种"脏数据"任务，但在实践中，这类任务对条件生成模型的训练很有用，因为分子属性标签往往缺失或不完整。引入预训练 BERT 可以提升预测器子模型对 EA 和 IE 等"罕见"属性的预测能力，尽管在某些情况下，这可能会降低预测器子模型对"常见"属性（本例中的分子量、辛醇-水分配系数和定量估计药物相似性）的预测性能。条件生成协同约束为分子量 = 250 道尔顿、辛醇-水分配系数 = 2.5 和 IE = 5 电子伏特。回归平均绝对误差根据数据库中的属性标签计算，而生成分子的属性则通过 RDKit 库或从头算模拟进行验证。

应用案例：锂离子电池局部高浓度电解质稀释剂筛选

最后，我们通过一个实际案例展示 ConGen模型的用法：生成化学空间，以便进一步虚拟筛选锂离子电池局部高浓度电解质（LHCE）中的稀释剂分子。近期锂离子电池电解液开发方面的进展促成了 LHCE 型电解质的发现。这类电解质在微观上类似于高盐浓度电解质（HCE），但在宏观上则更接近传统电解质。[16] LHCE 的优势在于，它在较宽的电化学窗口内保持稳定，此外还能形成稳定的固体电解质界面（SEI）层，这对电池的长期稳定性至关重要。[17,18]从成本角度看，LHCE 也很重要，因为与需要大量昂贵锂盐的 HCE 相比，它能减少所需锂盐的用量。[19] 最后，LHCE 的溶液黏度可以显著低于 HCE，这不仅有助于改善电解液的锂离子传输性质，还能促进电极润湿，从而更好地优化锂离子电池电芯的能量容量。[20] 从化学组成上看，LHCE 与 HCE 及传统电解质的区别在于，前者加入了充当电解液稀释剂的小分子。[16] 这些稀释剂分子通常是氢氟醚（HFE），例如双(2,2,2-三氟乙基)醚（BTFE）和1,1,2,2-四氟乙基-2,2,3,3-四氟丙基醚（TTE）；也可以是含氟原甲酸酯，例如三(2,2,2-三氟乙基)原甲酸酯（TFEO）。[16–18] 这类化合物的独特之处在于，它们具有足够的极性，但极性低于 LHCE 中使用的锂盐阴离子。因此，在适当的浓度范围内，$锂^+$ 阳离子主要会与极性盐阴离子在第一$锂^+$ 溶剂化壳层中配位。随后，稀释剂分子会从第二壳层开始，主要通过其极性氧原子与这些盐簇配位。此外，稀释剂分子中的含氟组分倾向于在 LHCE 中形成自身的非极性网络。因此，在 LHCE 中加入稀释剂，可确保盐簇在局部呈现类似 HCE 的结构（更稳定），同时在宏观上由稀释剂将这些

盐簇彼此分隔，并确保溶液黏度更低、具有离子导电性，且理想情况下不易燃（这是由于 LHCE 中易燃溶剂分子的比例降低）。

这些 LHCE稀释剂分子需要满足多项标准，例如电化学稳定性、不易燃性和低黏度。虽然目前已有几种已知有效的局部高浓度电解液稀释剂，但要丰富适用于优化特定类型锂离子电池的稀释剂化学空间，在这一领域寻找更多相关化合物至关重要。我们利用 ConGen模型，根据分子量、氟原子数、氧原子数、电离能、电子亲和能和对数黏度等结构与化学属性，生成局部高浓度电解液稀释剂的候选分子。为此，我们使用以下 5 个公开数据集的混合数据训练 ConGen模型：

- 分子量 来自 ZINC 的数据库[8,11]（310,000 个独特条目）
- 分子量、模拟的电离能、电子亲和能数据库，来源于Materials Project Electrolyte Genome[12]（62,274 个独特条目）
- 分子量、模拟的电离能、电子亲和能数据库，来源于 Austin Apple Github[21]（26,394 个独特条目）
- 氧氟碳化合物氟原子数、氧原子数数据库，来源于 PubChem[10]（200,000 个独特条目）
- 文献中的实验对数黏度数据库[22]（322 条唯一记录）

在适用的情况下，我们会使用 RDKit 补充各数据库中相应分子的分子量、氧原子数$_F$和氧原子数$_O$ 缺失属性标签，因为使用 RDKit 进行这项操作在计算上高效且成本低廉。[15] 合并后的数据库包含 571,023 个唯一分子。最后，我们评估模型的性能。根据已知的现有 LHCE 稀释剂，我们假设 LHCE 稀释剂分子需要具备以下属性：

- 电化学性质：电子亲和能 <= 0.5 电子伏特，电离能 >= 7.0 电子伏特
- 黏度性质：对数黏度 <= 0.0
- 结构性质：分子量 <= 300，n_F >= 4，氧原子数$_O$ = 1-2

在 ConGen 框架下，我们只需在

生成周期中将多个属性标签“锚点”同时用作解码器输入，即可实现这一多条件分子结构生成任务。例如，我们可以选择以下标签锚点来满足上述条件：

1. 电子亲和能 = 0 **或** 0.5 电子伏特 2. 电离能 = 7.0 **或** 7.5 电子伏特 3. 对数黏度 = -0.1 **或** 0.0 4. 分子量 = 250 **或** 300 道尔顿。5. n_F = 4 **或** 6 6. n_O = 1 **或** 2

因此，在上述示例中，我们有 $2^6 = 64$ 种可用于条件生成的多约束属性锚点组合。对于每组锚点，我们生成 5 个分子样本，使用基于RNN的ConGen模型共生成了 320 个条件采样分子（**查询 1**）。下方展示了仅基于可用属性标签的训练数据标签分布，见

图3.

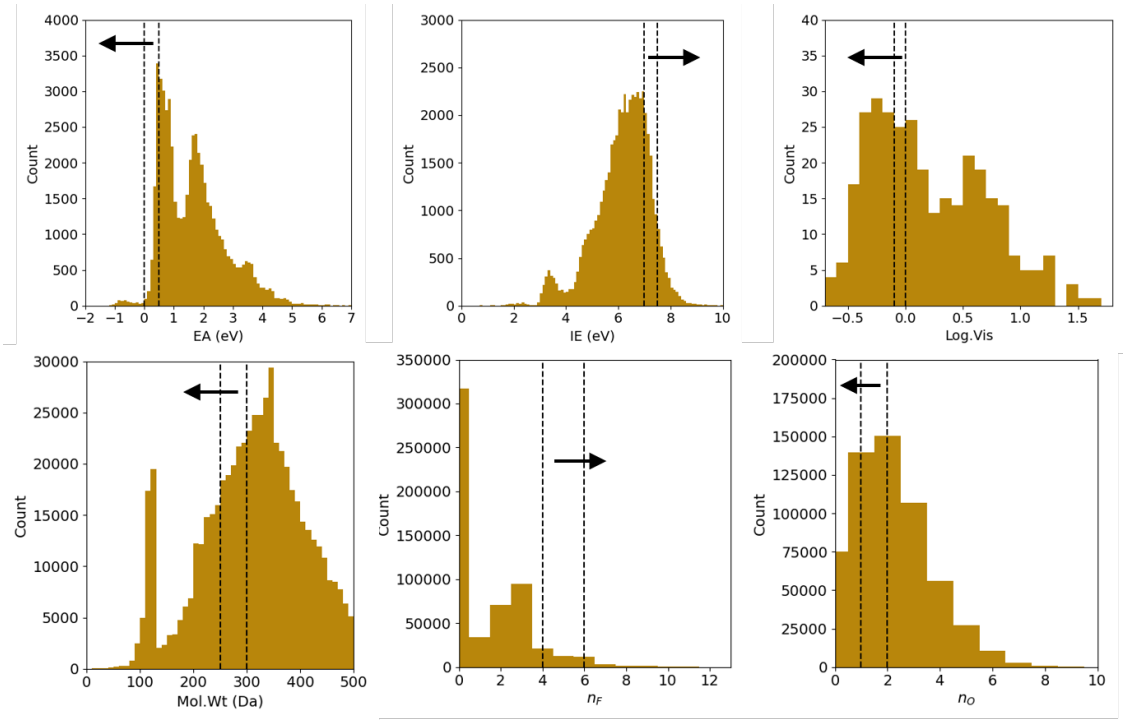

图 3 | 训练数据的分子属性标签分布。虚线表示我们将在后续条件分子生成中使用的属性标签“锚点”。箭头表示偏好的生成分子的属性范围。锚点分别为：电子亲和能 = [0,0.5], 电离能 = [7.0,7.5], 对数黏度 = [-0.1,0.0], 分子量 = [250,300], n_F = [4,6], n_O = [1,2]。

测试集上的回归、无条件分子生成以及条件分子生成结果如下所示，见 **图4** 和 **表3**。为了计算生成分子的真实属性标签值，我们采用了多种方法。对于 分子量、n_F 和 n_O，可以使用 RDKit 等简单的化学信息学工具快速计算其真实值。对于 电子亲和能 和 电离能，我们采用与先前工作[12] 相同的计算设置进行量子化学计算，以计算真实值。我们发现，我们能够很好地控制生成分子的结构性质（分子量、n_F 和 n_O）以及 电离能；不过，与锚点的 电子亲和能 均值（0.25 电子伏特）相比，生成分子的 电子亲和能 整体偏高约 2.0 电子伏特。我们推测，这种系统性偏移可能是由于我们的绝热电子亲和能计算流程与 Materials Project Electrolyte Genome 团队采用的流程略有不同所致；此外，我们向 ConGen 模型查询生成的分子，其 电子亲和能 标签锚点位于训练数据集 电子亲和能 标签分布的最左端（因此，这是我们采用的 6 个联合约束中最难满足的约束）。

目前我们尚不具备实验验证能力，无法测量生成分子的对数黏度，因此遗憾的是，无法展示这些分子的对数黏度属性的准确度指标。尽管如此，我们已在**表4**中列出了 ConGen 模型根据属性标签输入锚点生成的所有分子，以供具备实验能力的其他研究团队日后验证。要进一步提升生成的局部高浓度电解液稀释剂候选分子的质量，可能还需要加入其他分子性质标准。若要进一步细化生成的局部高浓度电解液稀释剂化学空间，只需添加一个包含所需分子性质的新逗号分隔值（CSV）文件用于训练，因此纳入更多分子性质约束应该很容易。生成的 320 个分子 SMILES 中，有 6 个是无效分子，3 个是重复项，5 个属于训练集。因此，我们根据此查询生成了 306 个新的唯一候选分子，用于计算验证。我们还使用基于 RNN 的 ConGen 模型生成了 64,000 个候选分子（每种锚点组合查询 1,000 次，见**图4**），尽管既未对电子亲和能也未对电离能进行从头算

由于计算成本高昂，并未对这些额外分子进行计算验证（**查询 2**）。在这次针对 64,000 个分子的新查询中，有 1,486 个无效分子，41,117 个重复项，356 个属于训练集。相应地，**查询 2**生成了 21,041 个新的唯一候选局部高浓度电解液稀释剂分子。未来还需要开展工作，以减少大规模查询重复项的数量。

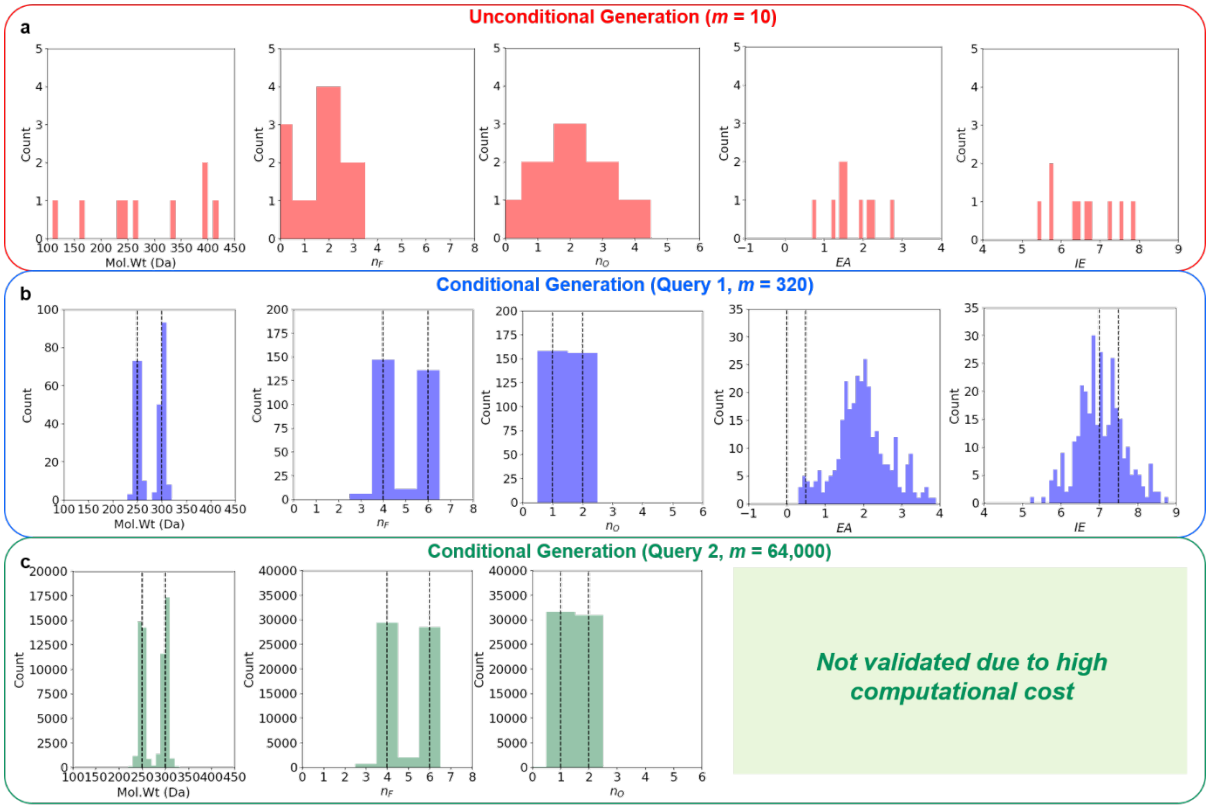

图 4 | ConGen 无条件和多约束条件分子生成的性质分布。（a）无条件分子生成，展示无约束条件下的性质分布。当利用多个联合约束进行条件生成时，我们可以实现高度针对性的分子生成。**查询 1**的结构和电化学稳定性性质验证结果（包含 320 个分子）如**（b）**所示，而**查询 2**的结构性质验证结果（包含 64k 个分子）如**（c）**所示。可以看到，在同时施加多种性质约束时生成的分子仍然很好地遵循其条件性质锚点（尽管最难满足的性质电子亲和能分布略有偏移），并且生成分子的性质分布与无性质约束条件下生成的分子显著不同。条件生成的性质锚点输入以虚线表示。

Task	分子量（道尔顿）	n_F	n_O	电子亲和能（电子伏特）	电离能（电子伏特）	对数黏度
预测器回归 测试集 平均绝对误差	1.60	0.01	0.02	0.20	0.21	0.14
解码器无条件生成	302 ± 110	1.50 ± 1.12	2.30 ± 1.42	1.71 ± 0.54	6.58 ± 0.75	N/A
解码器有条件生成（查询 1）	275 ± 26	5.02 ± 1.08	1.50 ± 0.50	1.99 ± 0.73	7.04 ± 0.61	N/A
解码器有条件生成（查询 2）	274 ± 26	5.02 ± 1.05	1.49 ± 0.50	N/A	N/A	N/A

表 3 | LHCE稀释剂分子的分子性质预测准确度及生成分子的性质分布统计。从回归测试结果可以看出，预测器子模型对分子性质的预测准确度较高。此外，无条件生成与有条件生成所生成分子的性质分布存在差异

生成案例表明，条件生成器能够根据我们选择的属性标签输入锚点生成正确的分子。 回归 平均绝对误差根据数据库中的属性标签计算，而生成分子的属性则通过 RDKit 库或 从头算模拟进行验证。

表 4 | 使用多约束 ConGen 模型（查询 1）生成的锂离子电池 LHCE 稀释剂分子候选项。

['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.5, 7.0, 0.0, 250, 4.0, 1.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.0, 7.0, 0.0, 250, 4.0, 1.0]
Cc1ccc(OCC(F)(F)C(F)(F)Cl)cc1 Nc1c(F)cc(OC(F)(F)F)cc1CCl CC(C)C(=O)Nc1ccc(F)c(C(F)(F)F)c1 CN(C)C(=O)Nc1c(F)c(F)cc(F)c1CF OCc1nc(C(F)(F)F)nc2c(F)cccc12	CN(C)C(=O)Nc1cc(F)cc(C(F)(F)F)c1 CC(C)(C)OC(c1cc(F)cc(F)c1)C(F)F CCOC(=N)c1c(F)cccc1CC(F)(F)F Nc1cc(C(F)(F)F)ccc1OCCC(F)C Fc1cc(OC(F)(F)F)cc(C2CCNC2)c1
['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.5, 7.0, 0.0, 250, 4.0, 2.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.0，7.0，0.0，250，4.0，2.0]
CN(C)C(=O)OC1(C(F)(F)F)CCC(F)C1 CCOC(=O)c1cc(C(F)(F)F)nc(F)c1C CC(=O)NCC(O)c1c(F)c(F)cc(F)c1F OCc1c(OC(F)(F)F)ccc(F)c1C1CC1 COc1ccc(OCC(F)(F)C(F)F)c(C)c1	Oc1cc(OC(F)(F)F)cc(F)c1CCl OC(OCC(F)(F)C(F)F)c1ccsc1 COC(=O)CC(CC(F)(F)F)c1ccccc1 Cc1cc(OCC(F)(F)C(F)(F)CO)ccn1 COC(=O)Cc1c(F)cnc(C(F)F)c1CF
['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.5, 7.0, 0.0, 250, 6.0, 1.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.0, 7.0, 0.0, 250, 6.0, 1.0]
Oc1nc(F)cc(CC(F)(F)C(F)(F)F)n1 FC(C(F)(F)F)C(F)(F)COC1CCCC1 FC(F)(F)C(F)(F)COc1ccccc1F OC(CCC(F)(C(F)(F)F)C(F)F)C1CC1 Cn1nc(C(F)(F)F)c(C(F)(F)F)c1CO	CC(OC(F)(F)F)c1ccccc1C(F)(F)F Cc1ncc(OC(F)(F)F)nc1C(F)(F)F OC(CCC(F)(F)C(F)(F)C(F)F)C1CC1 Fc1ccc(OCC(F)(F)F)cc1C(F)F NC(c1ccoc1)C(C(F)(F)F)C(F)(F)F
['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.5、7.0、0.0、250、6.0、2.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.0, 7.0, 0.0, 250, 6.0, 2.0]
COC(=O)CCCC(F)(F)C(F)(F)C(F)F OC(O)c1cc(C(F)(F)F)c(F)c(F)c1F Oc1cc(C(F)(F)F)cc(C(F)(F)F)c1O COC(=O)CCC(F)(C(F)(F)F)C(F)(F)F -- 无效--	OC(O)(CCCC(F)(F)F)CC(F)(F)F CC(=O)NCC(O)(C(F)(F)F)C(F)(F)F CCOc1ccc(C(F)(F)F)c(C(F)(F)F)c1 OC(OCC(F)(F)F)c1cccc(F)c1F Oc1c(OC(F)(F)F)cccc1C(F)(F)F
['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.5, 7.0, 0.0, 300, 4.0, 1.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.0, 7.0, 0.0, 300, 4.0, 1.0]
Cc1noc(-c2ccc(C(F)(F)F)cc2Cl)c1C(F)F Cc1cc(OC(F)(F)F)cc(F)c1I COc1cnc(C(F)(F)F)c(F)c1CBr Nc1c(F)cc(Oc2ccc(C(F)(F)F)cc2)c(Cl)c1 COc1cnc(-c2ccc(C(F)(F)F)c(F)c2)c(Cl)c1	Fc1cnc(OC(F)(F)F)c(I)c1 O=C(Nc1ncc(C(F)(F)F)cc1Cl)c1ccccc1 Cn1cc(-c2noc(-c3cc(F)c(F)c(F)c3F)n2)s1 NC(=O)c1c(F)cccc1Nc1cc(C(F)(F)F)ccn1 OC(Cc1ccc(C(F)(F)F)c(F)c1)c1cccs1
['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.5, 7.0, 0.0, 300, 4.0, 2.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.0, 7.0, 0.0, 300, 4.0, 2.0]
OC(Cc1cc(F)c(Br)cc1F)C(F)(F)O COC(=O)CCc1ccc(Cl)cc1C(F)(F)C(F)F CCOC(=O)Nc1c(C(F)(F)F)cc(F)nc1CCl N[C@@H](Cc1cc(F)c(F)c(F)c1F)c1ccc(O)cc1O CCC(=O)NCC(=O)Nc1cc(C(F)(F)F)cc(F)c1C	NC(=O)COc1ccccc1-c1c(F)c(F)cc(F)c1F OC(O)(c1cc(F)c(F)c(F)c1)c1ccc(F)cc1Cl NCc1cnc(OC(F)(F)F)nc1Oc1c(F)cccc1 COc1ccc(CNc2cc(F)c(F)c(F)c2F)cc1O FC(F)(F)Oc1cccc(Oc2cc(F)cc(Cl)c2)c1

['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.5, 7.0, 0.0, 300, 6.0, 1.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.0、7.0、0.0、300、6.0、1.0]
COC(c1cc(C(F)(F)F)nc(C(F)(F)F)c1)C1CC1 Nc1ccc(C(=O)NCC(F)(F)C(F)(F)C(F)F)cc1 NCc1cc(OC(F)(F)F)c(Cl)cc1C(F)(F)F NC(=O)c1cc(C(F)(F)F)cc(C(F)(F)F)c1CCl CC(C)C(=O)Nc1cc(C(F)(F)F)cc(C(F)(F)F)c1	COc1c(C(F)(F)F)ncc(C(F)(F)F)c1CCl OC(c1ccc(F)cc1)c1c(F)c(F)c(F)c(F)c1F Nc1cc(C(F)(F)F)cc(OC(F)(F)F)c1CCl FC(F)(F)c1cccc(-c2ccc(OC(F)(F)F)cc2)c1 Nc1c(F)cccc1Oc1cc(F)c(F)c(C(F)(F)F)c1
['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.5, 7.0, 0.0, 300, 6.0, 2.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.0, 7.0, 0.0, 300, 6.0, 2.0]
OCc1ccc(C(F)(F)F)cc1OCCC(F)(F)CF OB(O)c1c(C(F)(F)F)ccc(Cl)c1C(F)(F)F COc1ccc(OC(C(F)(F)F)C(F)(F)C(F)F)nc1 OCc1c(OC(F)(F)F)ncc(C(F)(F)F)c1C1CC1 CS(=O)(=O)Nc1cc(C(F)(F)F)cc(C(F)(F)F)c1	COc1cc(C(F)(F)F)c(C(F)(F)F)c(CC(N)=O)c1 Cc1cc(OC(F)(F)F)nc(OC(F)(F)F)c1CC#N Cc1ccc(COCC(F)(F)C(F)(F)C(F)F)cc1O O=C(O)Cc1cc(C(F)(F)F)nc(C(F)(F)F)c1CN OC(c1cccc(OC(F)(F)F)c1)c1ccc(F)cc1F
['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.5, 7.0, -0.1, 250, 4.0, 1.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.0, 7.0, -0.1, 250, 4.0, 1.0]
Oc1ccc(-c2ccc(F)c(F)c2)c(F)c1F COC(c1c(F)c(F)nc(F)c1F)C1CCC1 NCc1cn(CC(F)(F)F)nc1CC(=O)F NCc1cc(OC(F)(F)F)cc(Cl)c1F Cn1cnc(OC(F)(F)F)c1-c1ccc(F)cc1	Cc1nc(-c2cccc(C(F)(F)F)c2F)c(C)o1 CC(NC(=O)C(F)(F)C(F)F)c1ccccc1 CC(CO)Nc1ccccc1C(F)(F)C(F)F CC(C)Oc1nc(C(F)(F)F)c(F)cc1CN OCC1Cc2cc(C(F)(F)F)cc(F)c2S1
['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.5, 7.0, -0.1, 250, 4.0, 2.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.0, 7.0, -0.1, 250, 4.0, 2.0]
COc1cc(C(F)(F)F)nc(OC)c1CCF CC(O)c1c(OC(F)(F)F)cc(F)cc1CN OCCC(=O)Nc1ccc(F)c(C(F)(F)F)c1 CC(N)(C(=O)O)c1nc(C(F)(F)F)ccc1F O=C(O)c1ccn(CCC(F)(F)C(F)F)c1F	OCC(O)Cc1ncc(C(F)(F)F)cc1CF O=C(O)CC(CC(F)(F)C(F)F)c1ccc[nH]1 CCC(NCC(F)(F)C(F)F)C(=O)OCC COc1ccc2c(F)c(F)c(F)c(F)c2c1O NCc1cc(OC(F)(F)F)c(O)c(CF)c1C
['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.5, 7.0, -0.1, 250, 6.0, 1.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.0，7.0，-0.1，250，6.0，1.0]
OCc1nc(C(F)(F)F)c(C(F)(F)F)s1 OCCc1c(F)c(F)c(C(F)(F)F)c(F)c1 C[C@@H](O)c1c(F)c(F)c(C(F)(F)F)c(F)c1 OC(c1cc(F)cc(F)c1)C(F)(F)C(F)F CCc1c(OC(F)(F)F)n[nH]c1C(F)(F)F	N[C@@H](CO)c1c(F)c(F)c(F)c(F)c1CF CCc1ccc(OC(F)(F)F)c(C(F)(F)F)c1 Nc1cnc(OC(F)(F)F)c(C(F)(F)F)c1 OC[C@@H](c1cc(F)c(F)c(F)c1)C(F)(F)F COc1ncc(C(F)(F)F)c(C(F)(F)F)n1
['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.5、7.0、-0.1、250、6.0、2.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.0, 7.0, -0.1, 250, 6.0, 2.0]
C=C(O)CC(=O)C(C(F)(F)F)C(F)(F)CF OC(F)(F)C(F)(F)Oc1ccc(F)c(F)c1 OC[C@@H](O)CCC(F)(F)C(F)(F)C(F)(F)F OCCOCCC(F)(F)C(F)(F)C(F)(F)F OCc1c(O)c(F)c(F)c(F)c1C(F)(F)F	C[Si](C)(O)OC(F)(F)C(F)(F)C(F)(F)F COCC(O)CN(C(F)(F)F)C(F)(F)F CCOC(C)C(O)(C(F)(F)F)C(F)(F)CF O=C(O)CCCC(F)(F)C(F)CC(F)(F)F *--invalid--*

['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O']： [0.5, 7.0, -0.1, 300, 4.0, 1.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O']： [0.0、7.0、-0.1、300、4.0、1.0]
O=C(Cn1nc(C(F)(F)F)cc1Cl)c1ccccc1F CC1CCN(C(=O)Nc2cc(F)cc(C(F)(F)F)c2)CC1 OC(c1cc(F)cc(F)c1)c1ccc(C(F)(F)Cl)cc1 OC(c1cc(F)cc(F)c1)c1cnc(C(F)(F)Cl)cc1 CCc1nc(-c2ccc(OC(F)(F)F)cc2)nc(C)c1F	Cc1ccc(CC(=O)Nc2cc(F)cc(F)c2)c(F)c1F Cc1cc(C(F)(F)F)nc(Oc2cc(F)cc(Cl)c2)n1 CCCc1ncc(C(F)(F)F)c(Oc2ccc(F)cc2)n1 CC(OCC(F)(F)C(F)F)c1ccc(Cl)cc1Cl N#Cc1ccc(OCC(F)(F)C(F)F)cc1Br
['电子亲和能'、'电离能'、'黏度对数'、'分子量'、'n_F'、'n_O']： [0.5, 7.0, -0.1, 300, 4.0, 2.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O']： [0.0, 7.0, -0.1, 300, 4.0, 2.0]
Cc1cc(OC(F)(F)C(F)(F)C(=O)NC2CC2)cs1 COC(=O)c1ncc(C(F)(F)F)c(F)c1Br COC(=O)Cc1nc(C(F)(F)F)c(F)cc1CCl CCOC(=O)Cc1cc(C(F)(F)F)cc(F)c1CCl FC(F)(F)Oc1cc(OC(F)F)cc(Br)n1	NC(=O)COc1ccc(C(F)(F)F)c(F)c1Br COC(=O)c1ccc(C(F)(F)F)c(-c2ccc(F)cc2)c1 FC(F)(F)Oc1ccc(OCc2ccncc2)c(F)c1C CCOc1cc(OC(F)(F)F)c(F)cc1Br N[C@@H](CC(=O)O)c1c(C(F)(F)F)cc(F)cc1CCl
['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O']： [0.5, 7.0, -0.1, 300, 6.0, 1.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O']： [0.0, 7.0, -0.1, 300, 6.0, 1.0]
FC(F)C(F)(F)Oc1cc(C(F)(F)F)cnc1CCl OCc1c(C(F)(F)F)ccc(C(F)(F)F)c1CCl CCc1nc(OC(F)(F)F)c(C(F)(F)F)cc1CC#N OC(c1nc2ccccc2s1)C(F)(F)C(F)(F)CF Fc1ccc(-c2ccc(OC(F)(F)F)cc2)c(F)c1F	Nc1ccc(OCC(F)(F)C(F)(F)C(F)F)cc1C#N Oc1cc(F)c(-c2ccc(C(F)(F)F)cc2)c(F)c1F CCN(CC(F)(F)C(F)(F)C(F)(F)F)C(=O)NC1CC1 Fc1ccc(C(F)(F)F)c(Oc2cccc(F)c2)c1F Fc1cc(OC(F)(F)F)ccc1-c1ccc(F)c(F)c1
['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O']： [0.5, 7.0, -0.1, 300, 6.0, 2.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O']： [0.0, 7.0, -0.1, 300, 6.0, 2.0]
Cc1c(OC(F)(F)F)cnc(C(F)(F)F)c1CC(N)=O O=C(O)C(CC(F)(F)C(F)(F)C(F)F)c1ccccc1 OCc1cc(OC(F)(F)F)c(Cl)cc1C(F)(F)F C[C@@](N)(C(=O)O)c1nc(C(F)(F)F)c(C(F)(F)F)n1C CCOC(=O)c1cc(C(F)(F)F)ccc1C(F)(F)CF	Oc1ccc(COCC(F)(F)C(F)(F)C(F)F)cc1F OC(O)(Cc1ccc(C(F)(F)F)cc1)CC(F)(F)CF OC(O)(c1cc(C(F)(F)F)cc(C(F)(F)F)c1)C1CC1 O=C(NCC(F)(F)C(F)(F)C(F)F)c1ccc(O)cc1 FC(F)(F)COCCOc1c(F)cc(C(F)F)cc1N
['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O']： [0.5, 7.5, 0.0, 250, 4.0, 1.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O']： [0.0，7.5，0.0，250，4.0，1.0]
Cc1ccc(CNC(=O)C(F)(F)C(F)F)cc1 Cc1cc(CC(=O)NCC(F)(F)F)ccc1F O=C(c1cccnc1)c1cc(F)c(F)c(F)c1F FC(F)(F)c1cccc(Oc2ccccc2)c1F CCNC(=O)Nc1ccc(F)c(C(F)(F)F)c1	Cc1c(OC(F)(F)F)cc(F)cc1CCl OC(CC(F)(F)F)c1ccc(F)c(Cl)c1 COc1cc(C(F)(F)F)c(F)cc1CCl OCc1cnc(C(F)F)c(Cl)c1C(F)(F)F Nc1cc(OCC(F)(F)C(F)F)ccc1C#N
['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O']： [0.5, 7.5, 0.0, 250, 4.0, 2.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'，'n_O']： [0.0，7.5，0.0，250，4.0，2.0]
OCCC(=O)Nc1c(F)cccc1C(F)(F)F O=C(Cc1ccc(OC(F)(F)F)cc1)C1CC1 CCOc1c(OC(F)(F)F)cc(F)cc1C#N CCOC1(C(F)(F)F)Oc2ccc(F)cc2C1 CC(C)CC(=O)NCC(O)CC(F)(F)C(F)F	OCc1cc(OCCC(F)(F)F)cc(F)c1C OCc1c(C(F)F)ncc(OC(F)F)c1CC COc1nc(OC(F)(F)F)c(F)cc1CC#N Cc1ccc(C(F)(F)C(F)(F)C(=O)O)cc1C Cc1nc(C(F)(F)F)c(CC(=O)O)cc1CF

['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.5, 7.5, 0.0, 250, 6.0, 1.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.0，7.5，0.0，250，6.0，1.0]
O=C(NCC(F)(F)C(F)CF)CC(F)(F)F Cc1cnc(OC(F)(F)F)c(C(F)(F)F)c1 O=C(Nc1cc(F)cc(F)c1F)C(F)(F)F NCC(O)CCC(C(F)(F)F)C(F)(F)F Fc1cc(OCC(F)(F)F)ccc1C(F)F	OC(c1cc(F)c(F)c(F)c1)CC(F)(F)F FCOc1cc(C(F)(F)F)ccc1C(F)(F)F Cc1ncc(C(F)(F)F)c(OC(F)(F)F)n1 OC(c1c(F)c(F)c(F)c(F)c1F)C(F)F OCCc1c(F)c(F)c(C(F)(F)F)c(F)c1
['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.5，7.5，0.0，250，6.0，2.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.0, 7.5, 0.0, 250, 6.0, 2.0]
O=C(O)CCCCC(F)(F)C(F)(F)C(F)F CCOC(=O)CC(C(F)(F)F)C(F)(F)CF Fc1cccc(OC(F)(F)OC(F)(F)F)c1 Oc1cc(OC(F)(F)F)cc(C(F)(F)F)c1 CCC(=O)OCC(C(F)(F)F)C(F)(F)CF	FC(F)(F)C1(C(F)(F)F)OCCCCOC1 COC(=O)CCC(C(F)(F)F)C(F)(F)CF Cc1c(O)cc(C(F)(F)F)cc1OC(F)(F)F O=C(OCCCCC(F)(F)F)CC(F)(F)F CC(O)CCC(O)(C(F)(F)F)C(F)(F)F
['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.5、7.5、0.0、300、4.0、1.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.0, 7.5, 0.0, 300, 4.0, 1.0]
O=C(NCC(F)(F)C(F)F)c1ccc(Cl)cc1Cl CCc1ncc(OC(F)(F)F)c(F)c1CBr FC(F)(F)c1ccc(Oc2ncccc2Cl)c(F)c1 Nc1ncc(F)cc1C(=O)Nc1ccc(C(F)(F)F)cc1 CCCc1nc(OC(F)(F)F)nc(F)c1Br	CN(C)c1cc(C(F)(F)F)nc(Oc2ccc(F)cc2)n1 OC(c1cccc(C(F)(F)F)c1)c1c(F)cccc1Cl Cc1cnc(C(F)(F)F)c(Oc2ccc(F)c(Cl)c2)n1 N#Cc1ccc(COc2ccccc2C(F)(F)F)c(F)c1 CNc1ccc(Oc2ccc(C(F)(F)F)nc2)c(F)c1C
['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.5, 7.5, 0.0, 300, 4.0, 2.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.0, 7.5, 0.0, 300, 4.0, 2.0]
CC(Oc1cccc(C(F)(F)F)c1)C(=O)c1ccccc1 COCc1nc(OC(F)(F)F)c(F)cc1Br O=C(NCC(F)(F)CO)c1c(F)cc(F)cc1CCl FC(F)(F)Oc1cc(F)c(OCc2cccnc2)c(C)c1 CS(=O)(=O)Nc1ccc(C(F)(F)C(F)F)c(Cl)c1	CS(=O)(=O)N1CCN(c2c(F)c(F)cc(F)c2F)C1 Oc1cc(OC(F)(F)F)ccc1CBr CS(=O)(=O)Nc1ccc(SC(F)(F)C(F)F)cc1 O=S(=O)(c1cccc(C(F)(F)F)c1)c1ccc(F)cc1 CC(Nc1cc(C(F)(F)F)cc(F)c1)C(=O)OC(C)(C)C
['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.5、7.5、0.0、300、6.0、1.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.0, 7.5, 0.0, 300, 6.0, 1.0]
Cc1c(C(F)(F)F)cc(OC(F)(F)F)nc1CCl Nc1cc(OC(F)(F)F)cc(C(F)(F)F)c1CCl FC(F)(F)Oc1ccc(-c2ccc(C(F)(F)F)cc2)cc1 O=Cc1ccc(C(F)(F)F)c(C(F)(F)F)c1CCl *--invalid--*	Oc1c(F)cccc1-c1cc(C(F)(F)F)cc(F)c1F OC(c1cccc(C(F)(F)F)c1)c1cccc(F)c1F CC(=O)Nc1c(C(F)(F)F)cnc(C(F)(F)F)c1CN OCc1cc(C(F)(F)F)cc(C(F)(F)F)c1CCl OC(c1ccc(C(F)(F)F)nc1)c1c(F)cc(F)cc1F
['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.5, 7.5, 0.0, 300, 6.0, 2.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.0, 7.5, 0.0, 300, 6.0, 2.0]
OCC(Oc1cccc(C(F)(F)F)c1)CC(F)(F)CF O=Cc1cc(OCC(F)(F)C(F)(F)C(F)F)ccc1N OCc1c(OC(F)(F)F)ncc(C(F)(F)F)c1Cl NC(COCC(F)(F)F)c1cccc(OC(F)(F)F)c1 OCc1ccccc1OCCC(F)(F)C(F)(F)C(F)F	C[C@@H](NC(=O)O)c1cc(C(F)(F)F)cc(C(F)(F)F)c1 CCOC(=O)Cc1c(F)cc(C(F)(F)F)cc1C(F)F NC(=O)c1c(OC(F)(F)F)cnc(C(F)(F)F)c1CN O=C(NCC(F)(F)F)N1CCC(O)(C(F)(F)F)CC1 FC(F)(F)Oc1ccc(OC(F)(F)C(F)Cl)cc1

['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.5, 7.5, -0.1, 250, 4.0, 1.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.0，7.5，-0.1，250，4.0，1.0]
CNCC(=O)Nc1cc(F)c(C(F)(F)F)cc1 OC(CC(F)(F)F)c1ccc(F)c(Cl)c1 Fc1ccc(COc2ccccc2F)c(F)c1F OCc1cc(C(F)(F)F)c(F)cc1CCl COc1c(C(F)(F)F)ccc(F)c1CCl	OC(c1cc(F)cc(C(F)(F)F)c1)C1CCC1 OCCc1cc(C(F)(F)F)c(Cl)cc1F OCCc1nc(C(F)(F)F)c(F)cc1Cl Nc1cnc(OCCCC(F)(F)F)c(F)c1F Nc1ccc(OCCC(F)(F)C(F)F)c(C)c1
['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.5，7.5，-0.1，250，4.0，2.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.0, 7.5, -0.1, 250, 4.0, 2.0]
CN(C(=O)O)c1cc(C(F)(F)F)cc(F)c1C CS(=O)(=O)CCSCC(F)(F)C(F)F CCC(=O)NCC(=O)NCC(F)(F)C(F)F CNC1CC(C(F)(F)C(F)(F)C(=O)O)CC1 COC(=O)Nc1cccc(C(F)(F)C(F)F)c1	CCCC(=O)OCCCCC(F)(F)C(F)(F)F O=Cc1cc(OCC(F)(F)C(F)F)cs1 OC(COCC(F)(F)F)c1ccc(F)c(C)c1 CCCOC(=O)Nc1c(F)c(F)cc(F)c1F CCOC(=O)Nc1nc(C(F)(F)F)ccc1F
['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.5, 7.5, -0.1, 250, 6.0, 1.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.0，7.5，-0.1，250，6.0，1.0]
C[C@H](O)c1cc(C(F)(F)F)cc(C(F)(F)F)c1 OCC(F)(F)C(F)(F)c1cc(F)cc(F)c1 C[C@H](N)CC(=O)NC(C(F)(F)F)C(F)(F)F Cc1cc(OC(F)(F)F)nc(C(F)(F)F)c1 *--invalid--*	Cc1c(OC(F)(F)F)cccc1C(F)(F)F Nc1ncc(F)c(OC(F)(F)F)c1C(F)F CNCC(=O)NCC(F)(F)C(F)(F)C(F)F Fc1cc(F)c(OCCC(F)(F)F)c(F)c1 FC(F)(F)CCOc1cc(F)c(F)c(F)c1
['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.5，7.5，-0.1，250，6.0，2.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.0, 7.5, -0.1, 250, 6.0, 2.0]
OCc1c(F)c(F)c(OC(F)(F)F)c(F)c1 C=CCOC(=O)CC(F)(F)C(F)(F)C(F)F OCc1cc(C(F)(F)F)cc(C(F)(F)F)c1O Cc1c(OC(F)(F)F)[nH]c(C(F)(F)F)c1O C=CCOC(=O)C(C(F)(F)F)C(F)(F)CF	COC(=O)CCC(C(F)(F)F)C(F)(F)CF CC(CC(F)(F)C(F)(F)C(F)F)CC(=O)O CC(C)(CC(F)(F)C(F)(F)C(F)F)C(=O)O Cc1ccc(OC(F)(F)F)c(OC(F)(F)F)c1 FCOc1c(C(F)(F)F)[nH]c(C(F)F)c1O
['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.5，7.5，-0.1，300，4.0，1.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.0, 7.5, -0.1, 300, 4.0, 1.0]
OC(c1c(F)c(F)c(F)c(F)c1Br)C1CC1 Oc1nc(F)c(C(F)(F)F)cc1I OC(Cc1ccc(F)c(Br)c1)CC(F)(F)F COc1nc(C(F)(F)F)ccc1CBr O=C(Cc1nc(C(F)(F)F)ns1)c1ccc(F)cc1C	OC(c1cccc(C(F)(F)F)c1)c1ccc(F)cc1Cl CC1CC(c2ccc(C(F)(F)F)cc2)NC(=O)NC1F CC(NC(=O)CC(F)(F)C(F)F)c1ccc(Cl)cc1 COc1cc(Br)cc(CC(F)(F)C(F)F)c1 Cn1nc(C(F)(F)F)cc1C(=O)Nc1ccc(F)cc1C
['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.5, 7.5, -0.1, 300, 4.0, 2.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.0, 7.5, -0.1, 300, 4.0, 2.0]
CC(=O)Nc1nc(-c2ccc(C(F)(F)F)c(F)c2)c(C)o1 Oc1cccc(Oc2c(F)c(F)c(F)c(F)c2Cl)c1C O=S(=O)(Cc1ccc(F)cc1)c1c(F)cc(F)cc1F OC(O)(c1c(F)cccc1F)c1cc(F)c(Cl)cc1F *--invalid--*	CC(Oc1cccc(C(F)(F)F)c1)c1cccc(F)c1O COc1nc(C(F)(F)F)ccc1OCc1ccc(F)cc1 OC(COCC(F)(F)C(F)F)Cc1ncccc1Cl FC(F)C(F)(F)Oc1ccccc1COc1ccccc1 COC(=O)c1ncc(C(F)(F)F)cc1-c1ccc(F)cc1

['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.5, 7.5, -0.1, 300, 6.0, 1.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.0, 7.5, -0.1, 300, 6.0, 1.0]
OC(C(F)(F)C(F)(F)C(F)(F)F)C(Cl)(Cl)Cl	OCC(c1ccc(C(F)(F)F)c(C(F)(F)F)c1)C1CC1
OCC(c1c(F)cc(F)cc1Cl)C(F)(F)C(F)(F)F	FC(F)(F)Oc1cc(C(F)(F)F)c(Cl)cc1Cl
FC(F)(F)Oc1ccnc(-c2ccc(C(F)(F)F)cc2)c1	Cc1ccc(OCC(F)(F)C(F)(F)C(F)(F)Cl)cc1
CS(=O)c1cc(C(F)(F)F)cc(C(F)(F)F)c1C#N	Fc1cc(OC(F)(F)F)ccc1-c1cc(F)cc(F)c1
OC(F)(C(F)(F)F)C(F)(F)I	*--invalid--*
['电子亲和能'、'电离能'、'黏度对数'、'分子量'、'n_F'、'n_O']： [0.5、7.5、-0.1、300、6.0、2.0]	['电子亲和能', '电离能', '黏度对数', '分子量', 'n_F'、'n_O'] : [0.0, 7.5, -0.1, 300, 6.0, 2.0]
FC(F)(F)c1cc2c(cc1C(F)(F)F)OCCCCO2	OC(Cc1ccc(OC(F)(F)F)cc1)CC(F)(F)CF
CCCC(=O)OCCCCC(F)(F)C(F)(F)F	OCc1ccc(COCC(F)(F)C(F)(F)C(F)F)cc1
COC(=O)Cc1cc(C(F)(F)F)cc(C(F)(F)F)c1C	Oc1cc(OC(F)(F)F)ccc1SCC(F)(F)F
O=C(Cc1cc(C(F)(F)F)cc(C(F)(F)F)c1)C(N)=O	NC(=O)Cc1cc(C(F)(F)F)nc(OC(F)(F)F)c1C
CCc1cc(C(F)(F)F)cc(C(F)(F)F)c1CC(=O)O	OC(c1c(F)c(F)c(F)c(F)c1F)c1ccc(F)cc1O

结论

总之，我们展示了一种新颖的条件分子生成算法 ConGen，该算法基于半监督变分自编码器（SSVAE）技术。然而，与 SSVAE 模型不同，ConGen模型经过专门设计，能够处理标签不完整的含噪训练数据。这一点十分重要，因为实际上，我们从公开数据库中找到的分子，或通过内部模拟和实验表征的分子，其可用属性往往不完整，原因包括知识产权、商业秘密等因素，以及实验或计算成本方面的考量。基线 SSVAE 模型的用户必须删除标签不完整的分子，或为这些分子赋予虚拟标签，但这会降低模型准确率。另一方面，ConGen模型可以轻松混合来自多个来源的含噪训练数据集。ConGen还经过灵活设计，能够用其他类型的模型替换其子模型，使用户可以纳入预训练模型；这尤其有助于训练数据有限的情况。最后，我们展示了该模型的实际用途：针对锂离子电池局部高浓度电解液（LHCE）稀释剂候选物，在满足多项共同约束要求的前提下，生成用于虚拟筛选的化学空间。

实验方法

从头算电子亲和能与电离亲和能验证

这些分子的 EA 和 IE 计算采用 PySCF 实现的[23,24] DFT Kohn-Sham 方法，在 PBE6-31+G* 理论水平[25] 下进行，并在气相几何优化中加入 Grimme 色散校正[26] ；单点能计算则采用 B3LYP/6-31+G* 理论水平，并使用积分方程形式的极化连续介质模型（IEF-PCM）[27] 隐式溶剂化模型，对四氢呋喃（THF）的溶剂化能进行校正。振动频率在与几何优化相同的理论水平下，于 298.15K 计算，以确认每个优化后的驻点是否为能量极小值。在此，我们对不同电荷状态（阳离子、阴离子、中性）的分子分别进行几何优化，以计算绝热 IE/EA。

数据可用性

所有训练数据来源，以及对无条件和有条件生成分子进行的所有结构与计算验证，均可在 ESI 中获取。

作者贡献

J.P.M. 构思了本项目，开发了 ConGen模型，并开展和分析了模型训练及分子生成实验。S.Z. 负责监督本项目。X.L. 对生成分子进行了从头算计算验证。J.Q. 为模型训练优化工作提供了支持。J.P.M. 撰写了论文初稿，所有作者均审阅了论文。

利益冲突

没有需要声明的利益冲突。

致谢

本研究中的计算工作均在腾讯云平台上完成。

参考文献

1. Skinnider, M. A. 等 深度生成模型实现了对新型精神活性物质的自动结构解析。《自然·机器智能》 *Nat. Mach. Intell.* **3**, 973–984 (2021). 2. Li, Y., Pei, J. & Lai, L. 利用三维深度生成模型进行基于结构的从头药物设计。《化学科学》 *Chem. Sci.* **12**, 13664–13675 (2021). 3. Peng, H., Liu, Z., Yan, X., Ren, J. & Xu, J. 一种从头子结构生成算法，用于识别肝 X 受体激动剂的特权化学片段 β 。《科学报告》 *Sci. Rep.* **7**, 11121 (2017). 4. Prykhodko, O. 等 一种基于潜在向量生成对抗网络的从头分子生成方法。《化学信息学杂志》 *J. Cheminform.* **11**, 74 (2019). 5. Walters, W. P. & Barzilay, R. 深度学习在分子生成与分子性质预测中的应用。《化学研究述评》 *Acc. Chem. Res.* **54**, 263–270 (2021). 6. Gebauer, N. W. A., Gastegger, M., Hessmann, S. S. P., Müller, K. R. & Schütt, K. T. 利用条件生成神经网络逆向设计三维分子结构。《自然·通讯》 *Nat.Commun.* **13**, 973 (2022). 7. Tan, Y., Dai, L., Zheng, S., Chen, H. & Yang, Y. DRlinker：用于片段连接设计优化的深度强化学习。《化学信息与建模杂志》 *J. Chem. Inf. Model.* (2022) doi:10.1021/acs.jcim.2c00982. 8. Kang, S. & Cho, K. 利用深度生成模型进行有条件分子设计。《化学信息与建模杂志》 *J. Chem. Inf. Model.* **59**, 43–52 (2018). 9. Elton, D. C., Boukouvalas, Z., Fuge, M. D. & Chung, P. W. 用于分子设计的深度学习——最新进展综述。《分子系统设计与工程》 *Mol. Syst. Des. Eng.* **4**, 828–849 (2019). 10. Kim, S. 等 2021 年的 PubChem：新增数据内容并改进网络界面。《核酸研究》 *Nucleic AcidsRes.* **49**, D1388–D1395 (2021). 11. Irwin, J. J. & Shoichet, B. K. ZINC——用于虚拟筛选的市售化合物免费数据库。《化学信息与建模杂志》 *J. Chem. Inf. Model.* **45**, 177–182 (2005). 12. Qu, X. 等 电解质基因组计划：运用大数据推进电池材料发现。《计算材料科学》 *Comput. Mater. Sci.* **103**, 56–67 (2015). 13. Lorenzo-Seva, U. & Ferrando, P. J. 探索性项目因子分析中的非正定相关矩阵：成因、后果及解决方案。《结构方程模型》 *Struct. Equ. Model.* **28**, 138–147 (2021). 14. Chithrananda, S., Grand, G. & Ramsundar, B. ChemBERTa：面向分子性质预测的大规模自监督预训练。《第 34 届神经信息处理系统大会论文集》 *Proc. 34th Neural Inf. Process. Syst.* (2020). 15. RDKit：开源化学信息学工具。http://www.rdkit.org/ (2020). 16. Cao, X., Jia, H., Xu, W. & Zhang, J.-G. 综述——用于锂电池的局域高浓度电解质。《电化学学会志》 *J. Electrochem. Soc.* **168**, 010522 (2021). 17. Ren, X. 等 在实际条件下实现高压锂金属电池。*Joule***3**, 1662–1676 (2019).

18. Chen, S. 等 采用阻燃电解质的高效锂金属电池。*Joule* **2**, 1548 1558 2018– ( )。19. Yamada, Y. & Yamada, A. 综述——锂电池用超浓电解质。《电化学学会志》 **162**, A2406–A2423 (2015)。20. Giffin, G. A. 浓度在非水系锂电池电解质溶液中的作用。《自然·通讯》 **13**, 5250 (2022)。21. Austin Apple Github. https://github.com/AustinApple/SSVAE-for-electrolyte-molecule-design (2020)。22. Goussard, V. 等 一种用于快速估算液体黏度的新型机器学习工具：在化妆品油中的应用。《化学信息与建模杂志》 **60**, 2012–2023 (2020)。23. Sun, Q. 等 PySCF 程序包的最新进展。《化学物理学杂志》 **153**, 024109 (2020)。24. Sun, Q. Libcint：用于高斯基函数的高效通用积分库。《计算化学杂志》 **36**, 1664–1671 (2015)。25. Perdew, J. P., Burke, K. & Ernzerhof, M. 广义梯度近似简明实现。《物理评论快报》 **77**, 3865–3868 (1996)。26. Grimme, S., Antony, J., Ehrlich, S. & Krieg, H. 针对 H–Pu 共 94 种元素的密度泛函色散校正（DFT-D）的一致且准确的从头算参数化。《化学物理学杂志》 **132**, 154104 (2010)。27. Tomasi, J., Mennucci, B. & Cancès, E. PCM 溶剂化方法的 IEF 版本：一种用于在量子力学从头算水平研究分子溶质的新方法概述。《分子结构杂志：*THEOCHEM*》 **464**, 211–226 (1999)。