Molecular Structure-Property Co-Trained Foundation Model for In Silico Chemistry

Jinho Chang\textsuperscript{1} and Jong Chul Ye\textsuperscript{1,†}

\textsuperscript{1}Graduate School of AI, KAIST, Daejeon, South Korea

\textsuperscript{†}Correspondence should be addressed to J.C.Y. (jong.ye@kaist.ac.kr)

Abstract

Recently, deep learning approaches have been extensively studied for various problems in chemistry, such as virtual screening, \textit{de novo} molecule design, etc. Despite the impressive successes, end-to-end training for specific tasks usually requires separately designed networks, so it’s often difficult to acquire a unified principle to synergistically combine existing architectures and training datasets for novel tasks. To address this, inspired by recent advances of pre-trained multi-modal foundation models such as Vision-Language Pretrained models (VLP), here we present a novel multimodal foundation model that can be used \textit{in silico} for various downstream tasks in chemistry. Specifically, our framework, dubbed as the structure-property multi-modal (SPMM) foundation model, is based on the dual-stream transformer with X-shape attention, so that it can align the molecule structure and the chemical properties in a common embedding space. Accordingly, SPMM can simultaneously perform chemical property prediction from given structure-describing strings and allows the generation of molecular structures for given chemical properties, which was previously not possible with a single architecture. Furthermore, we show that the outstanding unimodal representation of a molecule emerges from multimodal learning, which has the potential to be fine-tuned for many other downstream tasks.
Introduction

For the last decade, deep learning has emerged as a promising tool in chemistry research for estimating many biochemical properties and interactions between molecules, polymers, and proteins, which are difficult to obtain experimentally\textsuperscript{1-3}. Various artificial intelligence (AI)-based approaches in the chemical domain employed deep neural networks to extract desired characteristics like intrinsic properties, biochemical activities, and chemical reactions from raw molecule data\textsuperscript{4-6}. Moreover, \textit{de novo} molecule design has been extensively studied using generative models\textsuperscript{7}, graph networks\textsuperscript{8}, etc.\textsuperscript{9,10} More recently, unsupervised learning approaches of learning better representations of the chemical inputs have been suggested \textsuperscript{11-13} to overcome the limitation of learning separate features for each task in a supervised manner.

These recent approaches are on the same track as the concept of the foundation model, often considered as a new paradigm of deep learning\textsuperscript{14}. Specifically, foundation models refer to the neural network models that are pre-trained in a self-supervised manner with a wide range of data, providing features that can be readily fine-tuned for various downstream tasks. For example, large pre-trained language models such as BERT\textsuperscript{15} and GPT-3\textsuperscript{16} learn the context between word tokens using self-supervised objectives like masked language model (MLM) and next world prediction (NWP), demonstrating powerful performance in many different NLP tasks. The success in NLP has also inspired the chemical foundation models\textsuperscript{17,18}, where molecule features that describe how atoms and functional groups are positioned are learned in a self-supervised manner.

Meanwhile, in the computer vision field, multimodal learning methods like Vision-Language Pre-training (VLP) have achieved outstanding performance in downstream tasks that require an understanding of both image and text\textsuperscript{19,20}. Specifically, VLP models learn the cross attention between different modality data by handling them collectively through Transformer architecture\textsuperscript{21-24}. By
embedding images and languages in the common feature space, VLP enables various tasks such as visual question answering (VQA), image-text retrieval, text-to-image attention, text-driven image generation, image-driven text generation, etc., which are not possible using single modality foundation models.

Inspired by the success of VLP, we are interested in the cross-modal comprehension between molecule structure and the property that enables meaningful tasks in many applications like property predictions, conditional molecule design, etc. Specifically, we propose a novel molecule Structure-Property Multi-Modal foundation model (SPMM) which allows various chemistry experiments \textit{in silico}. More specifically, SPMM is pre-trained with a wide range of molecules' structure and a vector of its properties by employing an X-attention Transformer architecture and training objectives through cross-attention between structure-describing strings and property descriptor vectors. We expect that learning structural features with information from the associate properties would give us a better representation of the input for various downstream tasks.

Figure 1 illustrates the overall model architecture and training objectives for SPMM, which extends the structure of the dual-stream VLP models. Specifically, a dual-stream model encodes the input for each modality with a unimodal encoder, then performs cross-attention by using one modality feature as a query and the other modality feature as a key/value. Since a molecule can also be represented with various forms of data such as a fingerprint, strings like SMILES (Simplified molecular-input line-entry system), InChI (International Chemical Identifier), or a molecular graph that contain almost the same information about complete molecular structure, we employ SMILES as a molecule structure describing language model.

In particular, SPMM takes a given molecule’s SMILES string and PV (property vector) as multi-modal data inputs. When the SMILES and PV are passed through their corresponding uni-
Figure 1: Overview of the model architecture and pre-training objectives of SPMM. The contrastive loss aligns the output feature of two unimodal encoders into the same embedding space. The fusion encoder learns the relations between two modalities, trained with Next Word Prediction (NWP), Next Property Prediction (NPP), and SMILES-Property Matching loss (SPM). The momentum teacher provides the knowledge distillation for the model.

Then, the encoded SMILES and PV features are passed through the cross-modal encoders, which perform cross-attention between two modalities. This fusion encoder can perform cross-attention with an alternation of its query and key/value input because the contrastive learning aligns the output of the SMILES encoder and the PV encoder into the same feature space. The cross-modal encoder is pre-trained with Next Word Prediction (NWP) for SMILES, Next Prop-
Property Prediction (NPP), and SMILES-PV Matching loss (SPM). Prediction of the next component from the given transformer input is a commonly used self-supervised learning objective, and our NWP and NPP tasks make the model learn the contextual relationship between SMILES tokens and properties with the aid of the other modality’s semantic feature. Additionally, SPM predicts whether a given pair of SMILES and PV represents the same molecule or not.

We have experimentally demonstrated that our pre-trained model performs well on a task that handles both SMILES and properties, such as SMILES-to-properties and properties-to-SMILES generation. Furthermore, this pre-trained model can be applied to single-modality downstream tasks and shows comparable performances to state-of-the-art models, which suggests the model’s generalization ability as a foundation model.

**Related works** A concept of pre-training a neural network in an unsupervised manner for a better feature representation has been adapted for various chemical fields. N-Gram Graph and GROVER used a graph neural network (GNN) and a graph transformer network, respectively, to obtain a pre-trained model from the molecular graph. ChemBERTa-2 trained roBERTa with 77 million SMILES to build a molecular foundation model.  

Several recent works tried to obtain a better feature of a molecule by sharing knowledge from different data representations. Winter et al. trained a translation model between SMILES and InChI key to get a feature vector with meaningful information that both molecular representations have in common. Zhu et al. used a self-supervised training method of BYOL between different molecule representations of SMILES and molecular graphs to build a dual-view model. Yet, these works introduced multimodality only for the enhancement of a molecule feature for unimodal tasks, not the interplay between those different modalities.
Results

The model learns cross-modal comprehension between SMILES and properties. After the pre-training process, we made our pre-trained SPMM generate a PV with SMILES input only. This task is equivalent to performing 53 property predictions of a given SMILES at once. No additional training is required since the pre-training with NPP already trained the model to perform property generation. Properties are predicted in an autoregressive manner: the model predicts the first property value using only the property [CLS] token, then takes all previous outputs again to get the next prediction value, and so on.

We fed 1,000 SMILES that are not contained in the pre-training dataset to SPMM and then generated their corresponding PV. Fig. 2 is the scatter plot of the real property value against the generated output for 12 selected properties out of 53 that we used for pre-training. It is clear that SPMM’s predicted property is very close to the actual value, and most of the data point lies on the $y = x$ line. Although the model virtually has never seen a full-filled PV in the pre-training due to the 50% of random property masking, the model could autoregressively predict all 53 properties as a whole. The mean $r^2$ score of the 53 properties was 0.9737. Since the values of the properties span multiple orders of magnitude, we measured the Root Mean Square Error (RMSE) on normalized values, and the mean RMSE of the 53 normalized properties was 0.1317. The full scatter plot for all 53 properties with each $r^2$ score and raw RMSE is in the Supplementary Figure S1.

With the same approach as property generation, the pre-trained SPMM can also generate SMILES that agree with a given property, which is a crucial challenge for many chemical tasks such as de novo molecule design. As one of the major approaches for drug discovery, various methods have been suggested for generating molecules with desired properties$^{7-10}$. In these approaches presented so far, the maximum number of simultaneously controllable properties was not
so large. Also, the length of the input property vector cannot be changed. Whenever the number or type of properties we want to control changes, the model requires separate training. In contrast, the pre-trained SPMM can take 53 properties used in pre-training as desired conditions and generate molecules that satisfy all of those conditions. Moreover, for the properties that we don’t want to control, we can ignore those conditions by replacing them with the [UNK] token that we used in pre-training. This is very useful because controlling all 53 input properties is not a usual situation in practice, and is also not easy since the properties are correlated and entangled (for example, ‘5 atoms & 30 bonds’ or ‘2 rings & 5 aromatic rings’ is unlikely to be a valid PV input).
To demonstrate the molecule generation capability of SPMM, we obtained 1,000 PVs from SMILES that are not contained in the pre-training dataset and fed them to the pre-trained model to generate SMILES with the input property. The sampling process was done in a deterministic manner (greedy sampling): starting from the SMILES [CLS] token, the model predicts the probability distribution of the next token and chooses the option with the highest probability. As a result, among the output of deterministic PV-to-SMILES generation for 1,000 PVs, 97.1% of the generated output were valid SMILES. The properties of the generated samples agree with the property input, with the mean normalized RMSE of 0.2216. It is remarkable that every generated SMILES is different from the source molecule of their input PV and doesn’t exist in the pre-train dataset.

Beyond using PVs from already existing molecules, we can also use arbitrary PV inputs to design SMILES with desired characteristics. Figure 3 contains some examples of SPMM’s molecule design by changing specific values in the input PV. In this demonstration, we only passed 12 properties as a condition when the other 41 properties are masked with the [UNK] token. The figure shows that the generated molecules follow the input modification while maintaining unmodified properties similarly. Our model is even able to generate molecules with the out-of-domain conditions such as \( \log P = 7 \) (note that \( \sim 3.0\% \) of the pre-training dataset has \( \log P > 7 \)).

Application fields like drug discovery often require generating many molecules from a single wanted target property. This can be done by sampling the next token stochastically from the modeled probability distribution instead of using a token with the highest probability. To verify our model’s ability to generate multiple molecules from a single PV input, we generated 1,000 SMILES with stochastic sampling on a fixed PV.

Figure 4 shows the property distributions of 1,000 molecules generated from the same PV input. The mode of each property distribution locates near the input property value [(a)]. In the
Figure 3: The example of modified molecules by changing specific values from the original PV: (1) The output of the same PV of the source molecule. (2) The output when \#aromatic\_ring is changed to 0. (3) The output when \#ring and \#aromatic\_ring are changed to 1. (4) The output when log P is changed to 7. (5) The output when \#rotatable\_bond is changed to 12. While generating these results, the other 41 property conditions are masked by [UNK] token.

situation when only some of the properties are given, the model only regards the known properties while the other masked properties are not restricted [(b), (c)]. When we replace all input properties with [UNK] token [(d)], the model performs an unconditional molecule generation, and the output follows the distribution of the pre-training dataset. The validity of the generated molecule fluctuates depending on how feasible or difficult the property input is, but it lands in the range of 0.7~0.9 in most cases. The uniqueness, the ratio between the number of unique molecules against the number of validly generated molecules, was almost 100% in every condition we have experimented with.

About the overall molecule generation performance of SPMM, we want to emphasize that SPMM can generate suitable SMILES for many property conditions that the model has not seen in its pre-training. When we trained SPMM without 50% of random property masking with [UNK] token, the model only functions when all 53 properties are given since the model has not seen the
Figure 4: Property distribution of the generated molecules with different PV inputs and [UNK] token masking. The red vertical dotted lines are the input property values, and the grey vertical lines are the mean of that property in the pre-training dataset. The controlled properties are colored in red, and uncontrolled properties (=masked with [UNK] token) are colored in blue. Only 12 properties are shown for each case since there’s not enough space. (a) All 53 properties are controlled, without using the [UNK] token. (b) Molecular Weight to 150, and the other property inputs are masked. (c) #ring, #rotatable bond, TPSA, and logP are controlled to 2, 10, 50, and 0. The other property inputs are masked. (d) Every property is replaced with [UNK] token.
partially given properties. However, even with the technique of [UNK] token masking, the model cannot face most of the $2^{53}$ possible property combination during the pre-training process. The SPMM’s ability to handle arbitrary property conditions for SMILES generation comes from the fact that it treats PV as a ‘language with 53 words’ and could focus on each property separately, not simply considering the entire property input as a single condition.

**Generalization ability as a molecular foundation model.** So far, we have demonstrated that our pre-trained SPMM can be applied to tasks that require an understanding of the relationship between SMILES and properties. Yet, there are a lot of challenges that only use SMILES data, like molecular property prediction. The multimodal pre-training process of SPMM includes adjusting the output of the unimodal encoder to contain contextual information from the other modality by aligning it with the other unimodal encoder’s output. We tried to observe if our model had learned a good representation that can be readily used for other tasks, even for a single modality. So we only utilized the SMILES encoder of pre-trained SPMM and made a benchmark study on 8 MoleculeNet downstream tasks. Each MoleculeNet task is a regression or classification task for pharmaceutical/biochemical applications like solubility, toxicity, and brain penetrability.

Table 1 contains the performance of SPMM and other models for MoleculeNet. Using only 6 BERT encoder layers with 9 million parameters, SPMM showed comparable performances with state-of-the-art models for all tasks. It achieved the best performance for BBBP, Clearance, and BACE regression tasks, showing its potential as a foundation model. We’ve also observed that the score of our model dramatically decreased without pre-training.

We also trained SPMM for the reaction prediction task on USPTO-480k dataset, which requires the model to predict the product SMILES from the reactant SMILES. This can be approached with a ‘separated task’ by indicating which molecule in the reactant is the major compo-
| Dataset       | regression [RMSE] | classification [AUROC in %] |
|---------------|-------------------|----------------------------|
|               | Delaney ESOL     | LIPO | Freesolv | BACE | Clearance | BBBP | BACE | Clintox |
| Delaney       | 1128             | 4200 | 642      | 1513 | 1         | 2039 | 1513 | 1478    |
| Freesolv      | 1.050            | 0.683| 2.082    | 2.253| 49.754    | 71.0 | 80.9 | 90.6    |
| BACE          | 1.074            | 0.812| 2.688    | 1.318| 52.077    | 69.7 | 77.9 | 77.5    |
| Clearance     | 1.083            | 2.072| 5.061    | -    | -         | 69.1 | 79.1 | 87.5    |
| BBBP          | 1.100            | 0.739| 2.764    | -    | -         | 68.7 | 84.5 | 72.6    |
| BACE          | 0.895            | 0.823| 2.272    | -    | -         | 69.5 | 82.6 | 81.2    |
| Clintox       | 0.798            | 0.660| 1.877    | -    | -         | 72.4 | 85.6 | 90.1    |
| ChemRL-GEM    | 0.889            | 0.798| 1.363    | 48.515| 69.5 | 82.6 | 56.3    |
| ChemRL-GEM(w/o pre-train) | 1.082 | 0.996| 3.019 | 1.382| 51.949| 66.6 | 79.6 | 67.9    |
| SPMM          | 0.837            | 0.725| 1.986    | 1.129| 46.173   | 73.0 | 82.4 | 90.1    |

Table 1: Benchmark result on MoleculeNet downstream tasks. The best performance for each task was written in bold. For each task, we fine-tuned our model in three random seeds and recorded the mean of those results. The benchmark model results are from ChemRL-GEM and ChemBERTa-2.

Discussion

In this work, we proposed a transformer-based foundation model called SPMM trained with a multimodal pre-training. In the process, we presented a method of treating property collections like a language so that the model could learn the relationship between SMILES and each property independently. We demonstrated that pre-trained SPMM showed remarkable performances in problems for interactions between SMILES and PV domains. And not only for multimodal challenges, but even its unimodal feature for SMILES, SPMM also provides a good representation that can be readily fine-tuned for various molecular downstream tasks. All of these results...
were obtained with a pre-training of 1,000,000 molecules, which is relatively small compared to other large pre-training approaches and still has room for better performance with more data and parameters in a better computational environment.

Despite the noticeable performances of SPMM, it has several chances for improvement. One of those comes from using the SMILES notation. Although SMILES can contain full details about the 2D structure of the molecule, the information on how atoms and bonds are connected only exists implicitly. Also, a slight modification in molecular structure can be a drastic change in SMILES. Graph format is another widely used modality for molecule representation that contains the explicit information of the adjacency matrix, which can be an alternative for SMILES. Another limitation in our current SPMM is that the 53 property we used happens to be invariant with the changes in the stereochemistry of the given molecule. It is known that considering stereochemistry plays a crucial part in various biochemical tasks. However, the 53 property we used cannot provide any knowledge about properties that depends on the stereochemistry since their values are unchanged in different stereoisomers. This makes the SMILES encoder output of the different stereoisomers converge since the contrastive loss aligns them to the same PV feature. We believe this is the prominent factor that lowered the performance of SPMM in MoleculeNet tasks. Overcoming these drawbacks of the current study and making the model more applicable to other chemical tasks could be the works for the future.

Nevertheless, we believe that our work and the idea of multimodal pre-training and handling properties as a sequence of words have a vast potential to be adapted to other challenges in the chemical field.
Methods

Handling SMILES and property values as a language. SMILES is a sequence of characters that represents the 2D structure of the molecule. Many researchers treat SMILES as a kind of language data and utilize a concept of language models for chemical tasks on SMILES data\(^9\,^{17}\,^{35}\). Like a standard BERT input, the raw SMILES string is tokenized by the tokenizer and embedded by the SMILES encoder with [CLS] token and [SEP] token. Our tokenizer uses a subword dictionary of 300 characters and character combinations obtained from the pre-training data SMILES corpus by the BPE algorithm.

Meanwhile, a set of chemical properties does not change its carrying information by changing the internal order, but they certainly have correlations between the properties. And it is known that a transformer architecture also performs well for different modalities like images, by giving arbitrary order to its component patches and treating them as a sequence. For this work, we used a PV that contains 53 molecular properties for each SMILES and considered this as a sentence with a length of 53. One benefit of viewing PV as a language is that we do not have to collect all elements to build a PV. In contrast to a simple vector input, some property elements can be removed or masked in our approach, as described in Figure 5.
Figure 5 illustrates our embedding procedure for the input data. Every element in the PV is a numerical value and normalized with the mean and standard deviation of that property. Each value in the PV is encoded to a feature vector using a linear layer as a value encoding ([a]).

Although every property we used can be easily and thoroughly calculated by the computer, this might not be the case for other properties and situations in practice. Actually, there is no problem in describing a molecule using only part of these properties. Therefore, we randomly replaced 50% of the property features into the [UNK] token to simulate that the property is unknown ([b]). This prevents the model from overly dependent on the specific property and also has the effect of data augmentation.

On top of the value encoding, we fixed the order of every 53 properties and added a positional encoding ([c]). We note that this position embedding is equivalent to giving an index for each property and adding an embedding of that corresponding index. Then we pass the final result to the PV encoder with the property [CLS] token.

**Pre-training objectives.** Contrastive learning aims to learn better unimodal representation by aligning the features from different modalities into the same feature space. When the encoded features of [CLS] tokens of SMILES $S$ and PV $P$ are given as $S_{cls}$ and $P_{cls}$, we calculate the similarity function $sim(S, P)$ and $sim(P, S)$ as:

$$sim(S, P) = (h_S(S_{cls}))^\top h_P(P_{cls}), \quad sim(P, S) = (h_P(P_{cls}))^\top h_S(S_{cls})$$

(1)

where $h_S$ and $h_P$ are the linear projection + normalization layer for SMILES and property vector, respectively. Now, for a given pair of $S$ and $P$, we calculate the SMILES-to-PV and PV-to-
SMILES intermodal similarities as follows:

\[ s_{2p} = \frac{\exp \left( \frac{\text{sim}(S, P_n)}{\tau} \right)}{\sum_{n=1}^{N} \exp \text{sim}(S, P_n)/\tau), \quad s_{p2s} = \frac{\exp \left( \frac{\text{sim}(P, S_m)}{\tau} \right)}{\sum_{m=1}^{M} \exp \text{sim}(P, S_m)/\tau)} \tag{2} \]

where \( \tau \) is a learnable temperature parameter, which has a sharpening effect by exaggerating the similarity difference. The intramodal similarities can be calculated in the same way.

\[ s_{2s} = \frac{\exp \left( \frac{\text{sim}(S, S_m)}{\tau} \right)}{\sum_{m=1}^{M} \exp \text{sim}(S, S_m)/\tau), \quad s_{p2p} = \frac{\exp \left( \frac{\text{sim}(P, P_n)}{\tau} \right)}{\sum_{n=1}^{N} \exp \text{sim}(P, P_n)/\tau)} \tag{3} \]

The overall contrastive loss is defined using the cross-entropy loss \( H \) and one-hot similarity \( y \):

\[ L_{\text{contrastive}} = \frac{1}{2}(H(y_{s2p}, s_{2p}) + H(y_{p2s}, s_{p2s}) + H(y_{s2s}, s_{2s}) + H(y_{p2p}, s_{p2p})) \tag{4} \]

Following the recent contrastive loss application in VLP, we build the SMILES and PV queues that stores the \( k \) most recent SMILES and PV instances and use them for contrastive loss. We set our queue size \( k \) to 4,096.

Next Word Prediction (NWP) trains the model to predict the \((n + 1)\)-th SMILES token when \( 0 \sim n \)-th tokens and the corresponding PV are given. Predicting the next token is a common objective for training language models, known for being utilized in the pre-training of GPT-3\(^{16}\). This can be done with a single-flow for each SMILES by applying causal mask in the self-attention of the SMILES encoder and fusion encoder. Let \( S \) and \( P \) denote the input SMILES and the corresponding PV, and \( p^{NWP}(S, P) \) denote the model’s predicted next word probability distribution. The loss for NWP is defined as

\[ L_{\text{NWP}} = H(S, p^{NWP}(S, P)) \tag{5} \]
since $S$ itself becomes a label for next word prediction task.

We applied a similar concept of NWP for the property vector as Next Property Prediction (NPP). NPP makes the model to predict the next property value using its corresponding SMILES and the previous properties. Since each property element is a numerical value, we replaced the cross-entropy loss in NWP to MSELoss. When $S$ and $P$ denotes the input SMILES and the corresponding PV, and $\hat{P}(P, S)$ denotes the model’s predicted next property values with causal mask in the PV and the fusion encoder, the loss for NPP is given by

$$L_{NPP} = MSE(P, \hat{P}(P, S))$$

(6)

In NPP, the model does not predict the property value if it is replaced with [UNK] token.

SMILES-Property Matching (SPM) learns if a given SMILES-PV pair is matched or not. We concatenate the feature of two [CLS] tokens from the fusion encoder output of the SMILES and PV, and performs a binary classification with a linear layer SPM head. The SPM loss can be defined as

$$L_{SPM} = H(y^{SPM}, p^{SPM}(S, P))$$

(7)

where $p^{NWP}$ is a output of SPM head and $y^{SPM}$ is a one-hot label for SMILES-PV matching.

For each SMILES and PV instance, we randomly select a “negative” pair from the other modality and match them as negative pairs. Here, we employed hard-negative mining, which gives a higher chance of being selected as a negative pair for instances that has a higher similarity according to Eq. (2) but is not a positive match. This makes the training more difficult and forces the model to learn how to distinguish similar instances.
The overall pre-training objective is the combined loss of Contrastive, NWP, NPP and SPM loss:

\[ L = L_{\text{Contrastive}} + L_{\text{NWP}} + L_{\text{NPP}} + L_{\text{SPM}} \] (8)

In contrastive learning, using a one-hot label could be too strict since it regards all instances that came from other pairs as equally-negative instances. However, some PVs might agree with many SMILES, not only one SMILES that they’re paired with. Even SMILES can be matched with different PVs since there’s a 50% of masking in a PV (for example, “MW=[UNK], logP=2.1, #atom=12” and “MW=78, logP=2.1, #atom=[UNK]” both explain Benzene, even if they came from the different molecules). A similar problem also occurs for NWP. Sometimes there could be multiple good options for being the next token, but using a one-hot label for ground truth might ignore this fact.

To resolve this issue, we built the momentum teacher model and utilized its output for contrastive learning and NWP. The momentum teacher performs a knowledge distillation by providing a pseudo-label that reflects how the teacher model comprehends. The model is trained to minimize the loss \( L \) to predict both one-hot labels \( L_{\text{onehot}} \) and the distilled pseudo-label \( L_{\text{momentum}} \), with an adjusting hyperparameter \( \alpha \). The parameters of the momentum teacher model \( w_{\text{momentum}} \) are updated by the exponential moving average (EMA) using the student model’s parameters \( w_{\text{model}} \) and a momentum hyperparameter \( \lambda \):

\[ L = (1 - \alpha)L_{\text{onehot}} + \alpha L_{\text{momentum}} \] (9)
and

\[ w_{\text{momentum}} = (1 - \lambda)w_{\text{model}} + \lambda w_{\text{momentum}} \]  

(10)

**Training for downstream tasks.** Supplementary Figure S2 describes how we employed our pre-trained model for downstream tasks. For property generation and SMILES generation (Supplementary Figure S2-(a), (b)), we don’t need an additional fine-tuning since their training objectives are already in the pre-training (NWP, NPP). For inferences, the model generates PV or SMILES with autoregressive sampling. Specifically, starting from [CLS] token, the model predicts the first component, and repeats taking the previous outputs to predict the next component until it’s done or meets a sign to stop. A causal mask has to be used for the self-attention of the fusion encoder and the corresponding unimodal encoder.

For MoleculeNet downstream tasks that only provide SMILES data, we utilized only the SMILES encoder part of the model (Supplementary Figure S2-(c)). After the input molecule is encoded with the SMILES encoder, we pass the feature of the [CLS] token through a classification/regression head to get an output. The classification/regression head consists of MLP with one hidden layer. We fine-tuned our model with the given training set and get a checkpoint with the lowest validation loss, and recorded that checkpoint’s performance in the test set.

The chemical reaction prediction task provides a reactant SMILES (that contains multiple reagent molecules) and a product SMILES. We encode these two inputs with the SMILES encoder, then feed them into the fusion encoder + token prediction head and train them to predict the original product SMILES (Supplementary Figure S2-(d)). In the inference stage, starting from the [CLS] token, the model predicts the next token until it generates the [SEP] token.
Data preparation. We obtained 1,000,000 SMILES of general molecules from PubChem for pre-training, then filtered out SMILES that are too long. All 53 properties we used can be calculated with SMILES using the RDKit python module. The dataset for the MoleculeNet downstream tasks is provided by the DeepChem python library, split to train/valid/test set in a ratio of 8:1:1 using a scaffold splitter. For the reaction prediction task, we used the USPTO-480k dataset which contains 479,035 pairs of reactants and the major product of their reaction.

Implementation details. We employed the architecture of 4 BERT layers for our PV encoder and SMILES encoder, and 4 BERT layers with cross-attention for our fusion encoder. Based on the BERT\textsubscript{base}, we reduced the model size by changing the hidden size to 384, the number of attention heads to 8, and the feedforward layer’s intermediate feature size to 2,048.

We pre-trained the model until it converges using a batch size of 32 and the AdamW optimizer with a weight decay of 0.02. The learning rate is warmed up to $1e^{-4}$ and decreased to $1e^{-5}$ with a cosine scheduler. We used the adjusting hyperparameter $\alpha$ of 0.4. Since the pseudo-label from the momentum teacher is not useful in the early stages of the training, we linearly increased $\alpha$ from 0 to 0.4 during the first epoch. The momentum hyperparameter $\lambda$ was fixed to 0.995, and the size of the PV and SMILES queue was 4,096. The full description of training for downstream tasks is in Supplementary Table 3.

Correspondence Correspondence and requests for materials should be addressed to Jong Chul Ye. (email: jong.ye@kaist.ac.kr).

Acknowledgements This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2019-0-00075, Artificial Intelligence Graduate School Program (KAIST)), National Research Foundation(NRF) of Korea grant NRF-2020R1A2B5B03001980, and by the KAIST Key Research Institute (Interdisciplinary Research Group)
Author Contributions  J.C. prepared the code, performed all experiments and analysis, collected data, and wrote the manuscript.  J.C.Y. supervised the project in conception and discussion, and prepared the manuscript.

Competing Interests  The authors declare that they have no competing financial interests.

Data Availability  Scaffold-split MoleculeNet datasets are available via DeepChem python module https://deepchem.io/, and raw databases can be found in the MoleculeNet website https://moleculenet.org/. The USPTO dataset that we’ve used in this work is available at https://github.com/wengong-jin/nips17-rexgen.

Code Availability  The code for SPMM will be available upon request.

References

1. Ryu, S. & Lee, S.  Accurate, reliable and interpretable solubility prediction of druglike molecules with attention pooling and bayesian learning (2022).

2. Kuenneth, C. & Ramprasad, R.  polybert: A chemical language model to enable fully machine-driven ultrafast polymer informatics (2022).

3. Moon, S., Zhung, W., Yang, S., Lim, J. & Kim, W. Y.  Pignet: A physics-informed deep learning model toward generalized drug–target interaction predictions (2022).

4. Chen, S. & Jung, Y.  A generalized-template-based graph neural network for accurate organic reactivity prediction (2022).
5. Paul, D. et al. Artificial intelligence in drug discovery and development. *Drug Discovery Today* (2020).

6. Xu, C., Wang, Y. & Farimani, A. B. Transpolymer: A transformer-based language model for polymer property predictions. *arXiv.org* (2022).

7. Lim, J., Ryu, S., Kim, J. W. & Kim, W. Y. Molecular generative model based on conditional variational autoencoder for de novo molecular design (2018).

8. Lim, J., Hwang, S.-Y., Moon, S., Kim, S. & Kim, W. Y. Scaffold-based molecular design with a graph generative model. *Chem. Sci.* **11**, 1153–1164 (2020).

9. Wang, W., Wang, Y., Zhao, H. & Sciabola, S. A transformer-based generative model for de novo molecular design (2022).

10. Zhou, Z., Kearnes, S., Li, L., Zare, R. N. & Riley, P. Optimization of molecules via deep reinforcement learning (2019).

11. Chithrananda, S., Grand, G. & Ramsundar, B. Chemberta: Large-scale self-supervised pre-training for molecular property prediction. *arXiv preprint arXiv:2010.09885* (2020).

12. Zhang, Z. et al. Can pre-trained models really learn better molecular representations for ai-aided drug discovery? *arXiv:2209.07423* (2022).

13. Melnyk, I. *et al.* Reprogramming large pretrained language models for antibody sequence infilling (2022).

14. Bommasani, R. *et al.* On the opportunities and risks of foundation models (2021).

15. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding (2018).
16. Brown, T. B. et al. Language models are few-shot learners (2020).

17. Ahmad, W., Simon, E., Chithrananda, S., Grand, G. & Ramsundar, B. Chemberta-2: Towards chemical foundation models (2022).

18. Horawalavithana, S. et al. Foundation models of scientific knowledge for chemistry: Opportunities, challenges and lessons learned. In Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models, 160–172 (Association for Computational Linguistics, virtual+Dublin, 2022).

19. Chen, F. et al. Vlp: A survey on vision-language pre-training (2022).

20. Radford, A. et al. Learning transferable visual models from natural language supervision (2021).

21. Chen, Y.-C. et al. Uniter: Universal image-text representation learning (2019).

22. Li, X. et al. Oscar: Object-semantics aligned pre-training for vision-language tasks (2020).

23. Li, J. et al. Align before fuse: Vision and language representation learning with momentum distillation (2021).

24. Yu, J. et al. Coca: Contrastive captioners are image-text foundation models (2022).

25. Park, S., Lee, E. S., Shin, K. S., Lee, J. E. & Ye, J. C. Self-supervised co-learning of uncurated images and reports enables oversight ai in radiology (2022).

26. Liu, S., Demirel, M. F. & Liang, Y. N-gram graph: Simple unsupervised representation for graphs, with applications to molecules (2018).

27. Rong, Y. et al. Self-supervised graph transformer on large-scale molecular data (2020).
28. Winter, R., Montanari, F., Noé, F. & Clevert, D.-A. Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations. *Chem. Sci.* **10**, 1692–1701 (2019).

29. Zhu, J. *et al.* Dual-view molecule pre-training (2021).

30. Wu, Z. *et al.* Moleculenet: a benchmark for molecular machine learning. *Chem. Sci.* **9**, 513–530 (2018).

31. Yang, K. *et al.* Analyzing learned molecular representations for property prediction. *Journal of Chemical Information and Modeling* **59**, 3370–3388 (2019). PMID: 31361484.

32. Hu, W. *et al.* Strategies for pre-training graph neural networks (2019).

33. Fang, X. *et al.* Geometry-enhanced molecular representation learning for property prediction. *Nature Machine Intelligence* **4**, 127–134 (2022).

34. Jin, W., Coley, C., Barzilay, R. & Jaakkola, T. Predicting organic reaction outcomes with weisfeiler-lehman network. In Guyon, I. *et al.* (eds.) *Advances in Neural Information Processing Systems*, vol. 30 (Curran Associates, Inc., 2017).

35. Lee, K., Jang, J., Seo, S., Lim, J. & Kim, W. Y. Drug-likeness scoring based on unsupervised learning (2021).

36. Kim, S. *et al.* Pubchem in 2021: new data content and improved web interfaces. *Nucleic Acids Research* **49**, D1388–D1395 (2021).

37. Ramsundar, B. *et al.* *Deep Learning for the Life Sciences* (O’Reilly Media, 2019).
Supplementary Materials

Figure S1: The scatter plots of the model's generated property against the real property value for all 53 properties. The $r^2$ score and RMSE for each property are described at the top of each plot.
Figure S2: Overview of the inference and fine-tuning of SPMM for various downstream tasks: (a) The inference process of pre-trained SPMM for molecule generation. (b) The inference process of pre-trained SPMM for PV generation. (c) The model architecture for MoleculeNet downstream tasks. The SMILES encoder of pre-trained SPMM is used as a backbone. (d) The model architecture for the reaction prediction task. We adapted the SMILES encoder and the fusion encoder of pre-trained SPMM, and build a translation model that takes the reactant SMILES as an input to predict the product SMILES.
Table 3: Detailed training hyperparameters for fine-tuning SPMM in MoleculeNet downstream tasks.